02 UPORABNA INFORMATIKA

2024 ‹ ŠTEVILKA 2 ‹ LETNIK XXXII ‹ ISSN 1318-1882

U P O R A B N A
I N F O R M A T I K A
2024 ŠTEVILKA 2 APR/MAJ/JUN LETNIK XXXII ISSN 1318-1882

Znanstveni prispevki

Strokovni prispevki

Informacije

2024 ŠTEVILKA 2 APR/MAJ/JUN LETNIK XXXII ISSN 1318-1882

Ustanovitelj in izdajatelj
Slovensko društvo INFORMATIKA
Litostrojska cesta 54, 1000 Ljubljana

Predstavnik
Slavko Žitnik

Odgovorni urednik
Mirjana Kljajić Borštnar

Tehnični urednik
Timotej Knez

Lektoriranje angleških izvlečkov
Marvelingua (angl.)

Oblikovanje
KOFEIN DIZAJN, d. o. o.

Prelom in tisk
Boex DTP, d. o. o., Ljubljana

Naklada
110 izvodov

Naslov uredništva
Slovensko društvo INFORMATIKA
Uredništvo revije Uporabna informatika
Litostrojska cesta 54, 1000 Ljubljana
www.uporabna-informatika.si

Revija izhaja četrtletno. Cena posamezne številke je 20,00 EUR. Letna naročnina za podjetja 85,00 EUR, za vsak nadaljnji izvod 60,00 EUR, za posameznike 35,00 EUR, za študente in seniorje 15,00 EUR. V ceno je vključen DDV.

Revija Uporabna informatika je od številke 4/VII vključena v mednarodno bazo INSPEC.

Revija Uporabna informatika je pod zaporedno številko 666 vpisana v razvid medijev, ki ga vodi Ministrstvo za kulturo RS.

Revija Uporabna informatika je vključena v Digitalno knjižnico Slovenije (dLib.si).

Izid publikacije je finančno podprla Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije.

Vabilo avtorjem

V reviji Uporabna informatika objavljamo kakovostne izvirne prispevke domačih in tujih avtorjev z najširšega področja informatike, ki se nanašajo tako na poslovanje podjetij, javno upravo, družbo in posameznika. Prispevki so lahko znanstvene, strokovne ali informativne narave, še posebno spodbujamo objavo interdisciplinarnih prispevkov. Zato vabimo avtorje, da prispevke, ki ustrezajo omenjenim usmeritvam, pošljejo uredništvu revije po elektronski pošti na naslov ui@drustvo-informatika.si.

Avtorje prosimo, da pri pripravi prispevka upoštevajo navodila, ki so objavljena na naslovu http://www.uporabna-informatika.si.

Za kakovost prispevkov skrbi mednarodni uredniški odbor. Prispevki so anonimno recenzirani, o objavi pa na podlagi recenzij samostojno odloča uredniški odbor. Recenzenti lahko zahtevajo, da avtorji besedilo spremenijo v skladu s priporočili in da popravljeni prispevek ponovno prejmejo v pregled. Sprejeti prispevki so pred izidom revije objavljeni na spletni strani revije (predobjava), še prej pa končno verzijo prispevka avtorji dobijo v pregled in potrditev. Uredništvo lahko še pred recenzijo zavrne objavo prispevka, če njegova vsebina ne ustreza vsebinski usmeritvi revije ali če prispevek ne ustreza kriterijem za objavo v reviji.

Pred objavo prispevka mora avtor podpisati izjavo o avtorstvu, s katero potrjuje originalnost prispevka in dovoljuje prenos materialnih avtorskih pravic. Avtorji prejmejo enoletno naročnino na revijo Uporabna informatika, ki vključuje avtorski izvod revije in še nadaljnje tri zaporedne številke. S svojim prispevkom v reviji Uporabna informatika boste pomagali k širjenju znanja na področju informatike. Želimo si čim več prispevkov z raznoliko in zanimivo tematiko in se jih že vnaprej veselimo

Uredništvo revije

Navodila avtorjem člankov

Članke objavljamo praviloma v slovenščini, članke tujih avtorjev pa v angleščini. Besedilo naj bo jezikovno skrbno pripravljeno. Priporočamo zmernost pri uporabi tujk in, kjer je mogoče, njihovo zamenjavo s slovenskimi izrazi. V pomoč pri iskanju slovenskih ustreznic priporočamo uporabo spletnega terminološkega slovarja Slovenskega društva Informatika, Islovar (www.islovar.org).

Znanstveni prispevek naj obsega največ 40.000 znakov, kratki znanstveni prispevek do 10.000 znakov, strokovni članki do 30.000 znakov, obvestila in poročila pa do 8.000 znakov.

Prispevek naj bo predložen v urejevalniku besedil Word (*.doc ali *.docx) v enojnem razmaku, brez posebnih znakov ali poudarjenih črk. Za ločilom na koncu stavka napravite samo en presledek, pri odstavkih ne uporabljajte zamika.

Naslovu prispevka naj sledi polno ime vsakega avtorja, ustanova, v kateri je zaposlen, naslov in elektronski naslov. Sledi naj povzetek v slovenščini v obsegu 8 do 10 vrstic in seznam od 5 do 8 ključnih besed, ki najbolje opredeljujejo vsebinski okvir prispevka. Sledi naj prevod naslova povzetka in ključnih besed v angleškem jeziku. V primeru, da oddajate prispevek v angleškem jeziku, velja obratno. Razdelki naj bodo naslovljeni in oštevilčeni z arabskimi številkami.

Slike in tabele vključite v besedilo. Opremite jih z naslovom in oštevilčite z arabskimi številkami. Na vsako sliko in tabelo se morate v besedilu prispevka sklicevati in jo pojasniti. Če v prispevku uporabljate slike ali tabele drugih avtorjev, navedite vir pod sliko oz. tabelo. Revijo tiskamo v črno-beli tehniki, zato barvne slike ali fotografije kot original niso primerne. Slikam zaslonov se v prispevku izogibajte, razen če so nujno potrebne za razumevanje besedila. Slike, grafikoni, organizacijske sheme ipd. naj imajo belo podlago. Enačbe oštevilčite v oklepajih desno od enačbe.

V besedilu se sklicujte na navedeno literaturo skladno s pravili sistema IEEE navajanja bibliografskih referenc, v besedilu to pomeni zaporedna številka navajenega vira v oglatem oklepaju (npr. [1]). Na koncu prispevka navedite samo v prispevku uporabljeno literaturo in vire v enotnem seznamu, urejeno po zaporedni številki vira, prav tako v skladu s pravili IEEE. Več o sistemu IEEE, katerega uporabo omogoča tudi urejevalnik besedil Word 2007, najdete na strani https://owl.purdue.edu/owl/research_and_citation/ieee_style/ieee_general_format.html.

Prispevku dodajte kratek življenjepis vsakega avtorja v obsegu do 8 vrstic, v katerem poudarite predvsem strokovne dosežke.

Dejavniki vpliva na prevzemanje aplikacij za napredno planiranje in terminiranje v proizvodnji

Nika Kalan[1], Marina Trkman[2]
[1] RESULT, d.o.o., Celovška cesta 182, 1000 Ljubljana
[2] Fakulteta za upravo, Univerza v Ljubljani, Gosarjeva 5, 1000 Ljubljana
nikaa.kalan@gmail.com, marina.trkman@fu.uni-lj.si

Izvleček

Danes so kupci zahtevni, saj želijo pravi izdelek ob pravem času na pravem mestu za pravo ceno. Za ohranjanje konkurenčnosti na globalnem trgu, se danes proizvodna podjetja odločajo za informatizicijo poslovanja s specializiranimi aplikacijami za napredno načrtovanje in terminiranje (APS). Odločitev podjetja za privzemanje teh aplikacij v obstoječ informacijski sistem ni enostavna. Značilen je odpor ljudi. Ker bi radi bolje razumeli ciljne uporabnike raziskujemo, kaj vpliva na namero uporabe APS. V raziskavi smo preverili vpliv univerzalnih konstruktov enotne teorije sprejemanja in uporabe tehnologije (UTAUT) ter dveh dodatnih konstruktov, ki sta negotovost zaposlitve in zaskrbljenost. Rezultati kažejo, da na namero uporabe statistično značilno vplivata zaskrbljenost in zaznana uspešnost. Rezultati raziskave so koristni vsem proizvodnim podjetjem, ki se odločajo za vpeljavo APS v svoj informacijski sistem.

Ključne besede: aplikacija za planiranje proizvodnje, UTAUT, zaskrbljenost, negotovost zaposlitve, informacijski sistemi.

Factors influencing the uptake of advanced planning and scheduling applications in manufacturing

Abstract

Nowadays customers are demanding. They want the right product at the right time at the right place for the right price. To remain competitive on the global market, today's manufacturing companies tend to digitalize their operations with specialized applications for advanced planning and scheduling (APS). The company's decision to adopt these applications into the existing information system is not an easy one. People's resistance is typical. As we would like to better understand the target users, we are investigating what influences the intention to use APS. In the research, we checked the influence of the universal constructs of the unified theory of acceptance and use of technology (UTAUT) and two additional constructs, which are job insecurity and anxiety. The results show that intention to use APS is statistically significantly influenced by the anxiety and perceived performance. The results of the research are useful to all manufacturing companies that think about implementing PAS into their information system.

Keywords: production planning application, UTAUT, anxiety, job insecurity, information systems.

1 UVOD

Proizvajanje dobrin se razvija in izpopolnjuje od industrijske revolucije do danes. Skozi zgodovino je razvoj proizvajanja šel čez štiri industrijske revolucije, ki so vsaka na svoj način pripomogle k hitrejši, optimalnejši in cenejši proizvodnji. Od industrijske revolucije 3.0 dalje, se ne da proizvajati ničesar v veliki količini brez uporabe računalnika. Del industrije 4.0 so algoritmi umetne inteligence, ki omogočijo boljšo pripravljenost na spremembe in preverjanje vplivov določenih scenarijev na proizvodne linije in podjetje kot celoto (Gander, 2023).

Danes so kupci zahtevni. Želijo pravi izdelek ob pravem času na pravem mestu za pravo ceno. Za

ohranjanje konkurenčnosti na globalnem trgu, se proizvodna podjetja odločajo za informatizacijo poslovanja s specializiranimi aplikacijami, ki sledijo konceptu naprednega načrtovanja in terminiranja proizvodnje (angl. advanced planning and scheduling; APS) (Hvolby & Steger-Jensen, 2010; Steger-Jensen in drugi, 2011). Koncept APS označuje način planiranja, ki upošteva tako proizvodni del podjetja kot zaledne službe, da doseže najboljše rezultate v dani situaciji. Aplikacije APS temeljijo na algoritmu optimizacije in načrtovanja na podlagi danih omejitev (Hvolby & Steger-Jensen; 2010), ki se nanašajo na nabor surovin, strojev in polizdelkov. Pogosta uporaba je v obliki časovnega in tabelaričnega pregleda, kar omogoča preglednost. Podjetjem omogoča optimizacijo načrtov za doseganje finančnih in drugih strateških ciljev.

Odločitev podjetja za privzemanje aplikacij APS v obstoječ informacijski sistem ni enostavno. V članku se fokusiramo na raziskovanje dejavnikov, ki vplivajo na privzemanje aplikacije APS med njenimi ciljnimi uporabniki. Enotna teorija sprejemanja in uporabe tehnologije (angl. unified theory of acceptance and use of technology; UTAUT) ponuja temelje tovrstnim raziskavam, saj opredeljuje univerzalne napovednike namena uporabe neke tehnologije (Venkatesh in drugi, 2003, 2016). Tako se naše prvo raziskovalno vprašanje glasi: Katere univerzalni UTAUT dejavniki vplivajo na namero uporabe aplikacij APS?

Ker je kontekst privzemanja APS edinstven, predvidevamo, da obstajajo tudi drugi napovedniki, ki jih omenjena teorija ne omenja. Osredotočili smo se na preiskovanje vzrokov za odpor ljudi do uporabe APS, saj vpeljava tehnologije vpliva na spremembe v načinu dela. Zaskrbljenost pred uporabo nove tehnologije lahko igra pomembno vlogo pri privzemanju (Venkatesh in drugi, 2003), podobno kot tudi negotovost zaposlitve zaradi nakupa nove tehnologije (Vander Elst in drugi, 2014). Vpliv dveh novih dejavnikov v članku statistično preverimo. Torej, naše drugo raziskovalno vprašanje se glasi: Ali zaskrbljenost in negotovost zaposlitve pri prevzemanju APS relevantno vplivata na namero uporabe APS?

V nadaljevanju članek sledi sledeči strukturi. V drugem in tretjem poglavju so predstavljena teoretična izhodišča in hipoteze. V četrtem poglavju je opisan metodološki pristop k naši raziskavi. Peto poglavje predstavi rezultate hipotez. V šestem poglavju je diskusija in v sedmem zaključne misli.

2 UPRAVLJANJE PROIZVODNJE

2.1 Optimizacija poslovanja proizvodnega podjetja

Proizvodnja omogoča pridelavo končnih izdelkov iz surovin z uporabo različnih metod, človeškega dela in opreme na stroškovno učinkovit način (Kenton, 2022). Učinkovite tehnike proizvajanja omogočijo podjetjem, da izkoristijo ekonomijo obsega in še povečajo zaslužek (Kenton, 2022). Planiranje proizvodnje je tako eden od ključnih procesov v proizvodnih podjetjih, ki omogoča usklajevanje različnih virov, kot so ljudje, stroji, surovine in čas, za doseganje ciljev proizvodnje. Gre za sistematičen pristop k določanju, organiziranju in usklajevanju proizvodnih aktivnosti, da bi dosegli optimalno uporabo virov, izpolnili zahteve trga in dosegli zadovoljstvo strank s pomočjo splošnega časovnega načrta (Jenkins, 2022). Vpliv planiranja proizvodnje je večplasten. Ker napovedovanje povpraševanja pomaga pri določanju optimalne zaloge, ima pomembno vlogo pri doseganju učinkovitosti, produktivnosti in konkurenčne prednosti proizvodnega podjetja (Jenkins, 2022).

V kontekstu planiranja proizvodnje ločimo med planiranjem in terminiranjem (angl. scheduling). Glavni cilj planiranja proizvodnje je priprava plana. Določa splošno usmeritev proizvodnje, postavi strateške cilje in dinamično določa ciljne ravni zalog za izpolnitev prihodnjega povpraševanja. Ob tem upošteva predvidena in dolgoročna naročila, sezonska nihanja, prodajne akcije in podobne vplive. Plan proizvodnje dobimo tako, da pregledamo vse možne alternative glede na omejitve in med njimi izberemo najboljšo. S planom postavimo okvir, v katerem kasneje natančneje razporedimo naloge. Pri terminiranju pa gre za detajlen pogled na izdelavo točno določenega artikla: kje, kdaj, kdo in iz česa se bo proizvajal. Gre za kompleksen urnik.

2.2 Digitalizacija proizvodnje

Cilj aplikacij APS je, da pomagajo obvladovat sledeče izzive, ki se pojavijo pri planiranju proizvodnje (Fleischmann in drugi, 2008, str. 82):

- Obstoj nasprotujočih si ciljev in dvoumnih preferenc

  Primer takšnih ciljev sta čim hitrejše zadovoljevanje želj strank in cilj po zmanjšanju zalog. Če želimo imeti čim manj zalog, moramo surovine naročevati sproti, kar pa lahko pripelje do dalj-

šega dobavnega roka našega izdelka, kar lahko zmanjša zadovoljstvo naše stranke.
- Negotova prihodnosti.
  Pri vsaki izvedbi plana lahko pride do manjših ali večjih napak, ki vse vplivajo na potek dogodkov. Na vse to se mora plan čim prej in čim boljše prilagoditi. Proti nepredvidljivosti se lahko borimo s pomočjo različnih taktik, kot so na primer varnostne zaloge, vendar to ni nujno zadostno ali pa varnostne zaloge zavzemajo prostor, ki ga sicer potrebujemo za druge izdelke.
- Število alternativ, ki jih je treba pregledati in oceniti, je preveliko.
  Aplikacija APS omogoča podjetjem boljši pregled nad proizvodnimi procesi ter napredno načrtovanje in optimizacijo virov, kar vodi k večji učinkovitosti, nižjim stroškom in posledično boljšemu konkurenčnemu položaju proizvodnega podjetja. Uporaba APS dokazano podjetjem nudi številne prednosti:
- Skrajšani časi nastavitve proizvodnje: Sistemi APS optimizirajo proizvodne načrte z upoštevanjem dejavnikov, kot so proizvodne zmogljivosti, razpoložljivost materiala in prioritete naročil. To vodi do skrajšanega časa nedejavnosti in minimiziranih časov nastavitve/menjave, kar ima za posledico hitrejše proizvodne cikle in krajše pretočne čase (RVJ, 2023).
- Upravljanje virov: Sistemi APS pomagajo pri dodeljevanju in uporabi virov z upoštevanjem razpoložljive proizvodne zmogljivosti in razpoložljivosti materiala. To zagotavlja učinkovito uporabo virov, zmanjšanje ozkih grl in izboljšanje splošnega upravljanja virov (RVJ, 2023).
- Natančni roki: Sistemi APS strankam zagotavljajo točne datume z upoštevanjem različnih dejavnikov, vključno s proizvodnimi zmogljivostmi, razpoložljivostjo materiala in prioritetami naročil. To pomaga pri upravljanju pričakovanj strank in zagotavljanju pravočasne dostave (Siva, 2022).
- Poenostavljeni proizvodni procesi: Sistemi APS poenostavijo proizvodne procese z optimizacijo proizvodnih urnikov. To vodi do povečane učinkovitosti pri načrtovanju in razporejanju, časov ciklov ter izboljšanega dodeljevanja in uporabe virov (RVJ, 2023).
- Izboljšano odločanje: Sistemi APS zagotavljajo m e - nedžerjem sprejemati odločitve na podlagi informacij. Z upoštevanjem dejavnikov, kot so proizvodne zmogljivosti, razpoložljivost materiala in napovedi povpraševanja, sistemi APS omogočajo boljše odločanje pri načrtovanju proizvodnje (RVJ, 2023).

3 RAZISKOVALNI MODEL

3.1 UTAUT

V dobi digitalne transformacije sta sprejemanje in uporaba tehnologije postala ključnega pomena za uspeh tako posameznikov kot organizacij. Razumevanje, zakaj nekatere tehnologije sprejmejo in uporabljajo, medtem ko druge ostajajo neizkoriščene, je ključnega pomena za oblikovanje učinkovitih strategij bodočega uvajanja teh tehnologij. Razumevanje lahko izboljšamo s pomočjo sistematične analize modela enotne teorije sprejemanja in uporabe tehnologije (UTAUT).

Model UTAUT se uporablja za napovedovanje in interpretacijo vedenjskih namenov in vedenja uporabnikov ne glede na tehnologijo (Venkatesh in drugi, 2003; Wu in drugi, 2022). Zanimiv je zato, ker njegovi dejavniki napovedujejo pripravljenost oziroma namen uporabe tehnologije (Wu in drugi, 2022). UTAUT omogoča vključitev dodatnih kontekstualnih dejavnikov, ki lahko pomembno napovedujejo sprejemanje in uporabo tehnologij (Venkatesh in drugi, 2012). Model je bil tako v preteklosti večkrat razširjen s strani več avtorjev, ki so ga aplicirali na specifične tehnologije. Mi smo se osredotočili na univerzalne napovednike namere uporabe iz originalnega članka, pri čemer smo opustili moderatorske spremenljivke in dejavnik olajševalni pogoji po zgledu Trkman (2023).

V raziskavi smo tako privzeli sledeče univerzalne UTAUT napovednike namere uporabe: zaznana uspešnost, zaznan napor, družben vpliv – hipoteze H1-H3. Tem napovednikom smo dodali dva kontekstualna napovednika, ki bi lahko pomembno prispevala k razumevanju namere uporabe APS. To sta zaskrbljenost in negotovost zaposlitve, ki ju obrazložimo v nadaljevanju v H4 in H5.

3.2 Razvoj hipotez

Študije so pokazale, da zaznana uspešnost neposredno vpliva na vedenjsko namero uporabe tehnologije, vključno z mobilnimi zdravstvenimi storitvami, mobilno trgovino in aplikacijami za sledenje stikom (Commer in drugi, 2018; Saprikis in drugi, 2021; Shanmugam in drugi, 2022). Zaznana uspešnost je povezana tudi z enostavnostjo uporabe sistema. Štu-

dije so pokazale, da je pričakovana uspešnost pozitivno povezana z enostavno uporabo mobilnih zdravstvenih storitev in odprtih podatkovnih tehnologij (Chua in drugi, 2018; Liu in drugi, 2022). Zaznana uspešnost lahko pozitivno vpliva na odnos do sprejemanja tehnologije. Študije so pokazale, da zaznana uspešnost in odnos do sprejemanja pozitivno vplivata na namero potrošnikov, da sprejmejo inovacijo (Saprikis in drugi, 2021). Iz tega sledi hipoteza:

H1: Zaznana uspešnost pozitivno vpliva na namero uporabe aplikacije APS.

Zaznan napor pomembno vpliva na namero po sprejetju nove tehnologije, vključno s tehnologijami e-uprave in mobilnega poslovanja (Commer in drugi, 2018; Naser Alraja in drugi, 2016). Zaznan napor se nanaša tudi na stopnjo enostavnosti, povezano z uporabo sistema. Študije so pokazale, da je pričakovani napor pozitivno povezan s pričakovano učinkovitostjo mobilnih zdravstvenih storitev in odprtih podatkovnih tehnologij (Shanmugam in drugi, 2022; Zuiderwijk in drugi, 2015). Iz tega sledi hipoteza:

H2: Zaznan napor negativno vpliva na namero uporabe aplikacije APS.

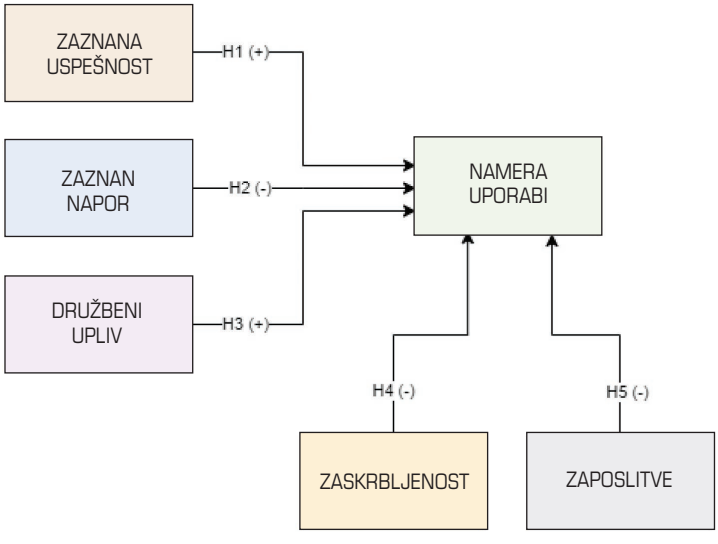
Družbeni vpliv lahko pozitivno vpliva na odnos do sprejemanja tehnologije. Študije so pokazale, da družbeni vpliv in odnos do sprejemanja pozitivno vplivata na namero potrošnikov, da sprejmejo inovacijo (Kulviwat in drugi, 2009) in tudi, da družbeni vpliv pozitivno vpliva na namero o uporabi različnih tehnologij, vključno s sistemi e-kampusa (Mohamad in drugi, 2021). Iz tega sledi hipoteza:

H3: Družbeni vpliv pozitivno vpliva na namero uporabe aplikacije APS.

Zaskrbljenost uporabnika se nanaša na njegovo tesnobo, bojazen, strah in nelagodje o morebitni uporabi nove tehnologije. Ta tesnoba lahko izvira iz pomanjkanja poznavanja tehnologij in negotovosti reševanja napak. Zaskrbljenost je pomemben napovedovalec sprejemanja tehnologije, saj lahko negativno vpliva na integracijo in uporabo tehnologije (Adikoeswanto in drugi., 2022). Študija Dönmez-Turan & Kir (2019) analizira 51 predhodnih študij, ki so proučevale razmerje med tesnobo uporabnikov in sprejemanjem tehnologije. Rezultati kažejo, da ima tesnoba uporabnikov pomemben negativen učinek na sprejemanje tehnologije, kar pomeni, da so višje stopnje tesnobe povezane z nižjimi stopnjami sprejemanja tehnologije. Uporabniki, ki doživljajo tesnobo, se morda ne bodo mogli v celoti vključiti v usposabljanje, kar ima negativne učinke na namero uporabe. Rezultati analize (Dönmez-Turan & Kir, 2019) kažejo na to, da ima zaskrbljenost uporabnikov pomemben negativen učinek na sprejemanje tehnologije, kar pomeni, da so višje stopnje zaskrbljenosti povezane z nižjimi stopnjami sprejemanja tehnologije. Iz tega sledi hipoteza:

H4: Zaskrbljenost negativno vpliva na namero uporabe aplikacije APS.

Ljudje v proizvodnji se v svojem profesionalnem življenju srečujejo s pritiski nadrejenih po večji produktivnosti. Ob tem se lahko poveča občutek posameznika, da je njegova služba ogrožena. Te občutke v psihologiji karakterizirajo kot negotovost zaposlitve (angl. job insecurity) (Vander Elst in drugi, 2014).



Slika 1: **Raziskovalni model**

Negotovost zaposlitve (angl. job insecurity scale) je tako stopnja zaznane negotovosti posameznika o izgubi trenutne službe (Vander Elst in drugih, 2014). Gre za subjektivno doživljanje, ki se razvije iz posameznikovega dojemanja in interpretacije dejanskega delovnega okolja (enako okolje lahko drugače vpliva na različne zaposlene). Delovno okolje v proizvodnem podjetju se z robotizacijo in digitalizacijo spreminja, kar predstavlja stres zaposlenim. Ta stres lahko povzroči strah pred izgubo zaposlitve, ki se kaže kot odpor pred uporabo nove APS aplikacije (Wang in drugi, 2023). Iz tega sledi hipoteza:

H5: Negotovost zaposlitve negativno vpliva na namero uporabe aplikacije APS.

Raziskovalni model je predstavljen na Sliki 1.

4 EMPIRIČNA RAZISKAVA

4.1 Opis izvedbe ankete in pridobljenega vzorca

Za preverjanje raziskovalnih vprašanj je bila izvedena spletna anketa pripravljena v orodju 1ka. Izvedbo je vodilo podjetje Result, slovenski ponudnik aplikacije APS. Namen ankete je bilo bolje razumevanje vedenja ciljnih uporabnikov aplikacije APS.

Zbiranje podatkov je bilo opravljeno s pomočjo agencije Red Bumerang, ki je podjetju ponudila svoj spletni panel. Povezavo do ankete v orodju 1ka je agencija poslala po elektronski pošti več kot 6000 ljudem v Sloveniji, ki so zaposleni v različnih proizvodnih podjetjih. Anketa je bila aktivna od 12. 5. do 24. 6. 2023. V tem času je na vsaj 60 % vprašanj odgovorilo 270 anketirancev, ki jih v nadaljevanju analiziramo.

Med anketiranci so bili tako uporabniki digitalnih planskih tabel (15%) kot tisti, ki se z njimi še niso srečali (85%). V raziskavi je sodelovalo 74,1 % moških in 25,9 % žensk. Povprečna starost je bila 43,6 leta, vendar je največ respondentov v času ankete imelo 46 let. Najnižja starost respondenta je bila 24, najvišja pa 66 let. Kar 61,1 % respondentov ima prvo, 22,9 % drugo in 5,7 % tretjo stopnjo visokošolske izobrazbe. Preostalih 10,3 % jih je imelo srednjo šolo. Največ respondentov je zasedalo delovno mesto tehnologa oziroma planerja in sicer 27,9 % respondentov - to so torej tisti, ki so najbolj vpleteni v uporabo aplikacij APS. Prodajalcev izdelkov je bilo 24,8 %, vodij proizvodnje 18,5 %, nabavnikov 10,4 %, tehničnih direktorjev 9,1 %, glavnih direktorjev 5,7 % in vodij informatike 3,6 %.

4.2 Spremenljivke

Spremenljivke smo prevzeli iz različnih virov, kot kaže Tabela 1. Uporabljena je Likertova lestvica od 1 do 5.

Tabela 1: **Spremenljivke**

Ref.	Latentna spremenljivka	Koda sprem.	Spremenljivka
(Venkatesh in drugi, 2003)	ZAZNANA USPEŠNOST	ZU1	Uporaba digitalne planske table bi mi olajšala delo.
		ZU2	Zaradi uporabe digitalne planske table bi delo opravil hitreje.
		ZU3	Uporaba digitalne planske table bi povečala mojo produktivnost.
		ZU4	Uporaba digitalne planske table bi povečala verjetnost za povišico.
(Venkatesh in drugi, 2003)	ZAZNAN NAPOR	ZN1	Moje interaktivno delo z digitalno plansko tablo bi bilo jasno in razumljivo.
		ZN2	Zame bi bilo enostavno obvladati uporabo digitalne planske table.
		ZN3	Digitalna planska tabla bi se mi zdela preprosta za uporabo.
		ZN4	Naučiti se, kako uporabljati digitalno plansko tablo, bi bilo zame enostavno.
(Venkatesh in drugi, 2003)	DRUŽBENI VPLIV	DV1	Ljudje, ki vplivajo na moje vedenje, mislijo, da bi moral uporabljati digitalno plansko tablo.
		DV2	Meni pomembni ljudje menijo, da bi moral uporabljati digitalno plansko tablo.
(Venkatesh in drugi, 2003)	NAMERA UPORABE	NU1	Ob predpostavki, da imam dostop do digitalne planske table, jo nameravam uporabiti v naslednjih 6 mesecih
		NU2	Glede na to, da imam dostop do digitalne planske table, predvidevam da jo bom uporabil v naslednjih 6 mesecih
		NU3	Predvidevam, da bi redno uporabljal digitalno plansko tablo, če bi imel dostop do nje
		NU4	V prihodnosti nameravam uporabljati digitalno plansko tablo

Ref.	Latentna spremenljivka	Koda sprem.	Spremenljivka
(Venkatesh in drugi, 2003)	ZASKRBLJENOST	Z1	Strah me je uporabe digitalne planske table.
		Z2	Strah me je ob misli, da bi lahko z uporabo digitalne planske table izgubil veliko informacij, če bi pritisnil napačno tipko.
		Z3	Oklevam z uporabo digitalne planske table, ker me je strah napak, ki jih ne bom mogel popraviti.
		Z4	Digitalna planska tabla me straši.
(Vander Elst in drugi, 2014)	NEGOTOVOST ZAPOSLITVE	NEG1	Obstaja verjetnost, da bom kmalu izgubil službo.
		NEG2	Prepričan sem, da lahko obdržim službo.
		NEG3	Negotovo se počutim o prihodnosti moje službe.
		NEG4	Mislim, da bom v bližnji prihodnosti izgubil službo.

5 REZULTATI

Zbrani podatki so bili analizirani s pomočjo strukturnega modela PLS-SEM in sicer v programu SmartPLS 3. PLS-SEM je za našo raziskavo primeren, ker lahko oceni razmerja med latentnimi spremenljivkami (dejavniki), ki nas zanimajo (Hair Jr in drugi, 2021).

5.1 PLS-SEM: ocena merskega modela

Kakovost modela ocenjujemo na podlagi koeficienta indikatorjev (angl. loadings) (Hair Jr in drugi, 2021, str. 186). Priporočena vrednost faktorja je nad 0,708 (Hair Jr in drugi, 2021). Iz Tabele 2 je razvidno, da imajo koeficienti indikatorja v naši raziskavi vsi ustrezno visoke vrednosti in so tako primerni za nadaljnjo analizo.

Tabela 2: **Faktorji in intervali zaupanja 2,5 % in 97,5 %**

Koda konstrukta.	Koda spremenljiv.	Koeficient indikatorja	T-test	P-vrednosti	Spodnja meja – 2,5 %	Zgornja meja – 97,5 %
ZASKRBLJENOST	Z1	0,876	30,889	0,000	0,808	0,919
	Z2	0,870	33,519	0,000	0,814	0,936
	Z3	0,894	30,101	0,000	0,819	0,936
	Z4	0,911	63,612	0,000	0,880	0,937
NEGOTOVOST ZAPOSLITVE	NZ1	0,811	12,701	0,000	0,752	0,941
	NZ2	0,904	14,852	0,000	0,827	0,951
	NZ3	0,912	13,412	0,000	0,784	0,962
ZAZNANA USPEŠNOST	ZU1	0,925	72,264	0,000	0,897	0,947
	ZU2	0,949	105,123	0,000	0,930	0,965
	ZU3	0,913	49,338	0,000	0,871	0,943
ZAZNAN NAPOR	ZN1	0,786	17,006	0,000	0,681	0,862
	ZN2	0,897	45,043	0,000	0,852	0,930
	ZN3	0,920	76,115	0,000	0,894	0,942
	ZN4	0,856	32,530	0,000	0,795	0,899
DRUŽBENI VPLIV	DU1	0,938	47,115	0,000	0,887	0,964
	DU2	0,954	96,611	0,000	0,933	0,972
NAMERA UPORABE	NU1	0,860	31,761	0,000	0,799	0,906
	NU2	0,803	20,565	0,000	0,715	0,872
	NU3	0,871	43,111	0,000	0,827	0,906
	NU4	0,831	26,034	0,000	0,758	0,881

Tabela 3: **Diskriminantna veljavnost – HTMT**

	DRUŽ. VPLIV	NEGOTOVOST	NAME. UPORA	ZASKRBLJENO.	ZAZNAN NAPOR	ZAZNA.USPEŠ.
DRUŽBENI VPLIV	0,075					
NAMERA UPORABE	0,349	0,187				
ZASKRBLJENOST	0,114	0,396	0,364			
ZAZNAN NAPOR	0,320	0,073	0,503	0,288		
ZAZNANA USPEŠNOST	0,484	0,036	0,587	0,246	0,645	

Tabela 3 kaže rezultate HTMT, ki ocenijo veljavnost diskriminante. Rezultati naše raziskave kažejo, da so vse spremenljivke sprejemljive, saj se nobene izmed vrednosti ni večja od 0,85 (Hair Jr in drugi, 2021).

Zanesljivost modela merimo s pomočjo Cronbachove alfe, ki to oceni s primerjavo količine skupne variance (Collins, 2007). Vrednosti se gibajo med 0 in 1. Sprejemljiva je od 0,7 naprej, kar je pod 0,5, pa je nesprejemljivo (Saidi & Siew, 2019). V našem primeru se rezultati Cronbachove alfe (Tabela 4) gibajo med 0,883 in 0,921, kar je sprejemljivo. Pri evaluaciji modela je pomembna tudi zanesljivost notranje konsistence, ki pove, v kakšni meri določeni indikatorji merijo isti konstrukt (Hair Jr in drugi, 2021). Za to uporabimo CR (angl. composite reliability) in AVE (angl. average variance extracted). Vse vrednosti CR so nad 0,7 in vrednosti AVE nad 0,5, kar je sprejemljivo (Hair in drugi, 2012).

5.2 PLS-SEM: ocena strukturnega modela

Vrednosti koeficienta determinacije (angl. coefficient of determination; R2) višje od 0,25, 0,50 in 0,75, štejejo na šibko, zmerno in za znatno veliko pojasnjevalno moč (Hair in drugi, 2011). Naš raziskovalni model pojasni 37 % variance namere uporabe, kar kaže na šibko pojasnjevalno moč.

Tabela 4: **Merjenje zanesljivosti modela: Cronbachova alfa, rho_c, AVE**

	Cronbachova alfa	CR	AVE
DRUŽBENI VPLIV	0,883	0,896	0,895
NEGOTOVOST	0,882	0,885	0,808
NAMERA UPORABE	0,865	0,883	0,709
ZASKRBLJENOST	0,911	0,925	0,788
ZAZNAN NAPOR	0,888	0,892	0,750
ZAZNANA USPEŠNOST	0,921	0,930	o,863

Nadalje smo ugotavljali, kateri napovedni dejavnik je imel največji učinek na odvisno spremenljivko. Vrednosti f2 nad 0,02, 0,15 in 0,35 pomenijo majhen, srednji oziroma velik učinek (Hair in drugi, 2019). Tabela 5 kaže, da imata zaskrbljenost in zaznano uspešnost majhen učinek.

Tabela 4: **Vpliv posameznega konstrukta na namero uporabe – f2**

	NU
DRUŽBENI VPLIV	0,012
NEGOTOVOST	0,015
NAMEN UPORABE	
ZASKRBLJENOST	0,037
ZAZNAN NAPOR	0,026
ZAZNANA USPEŠNOST	0,124

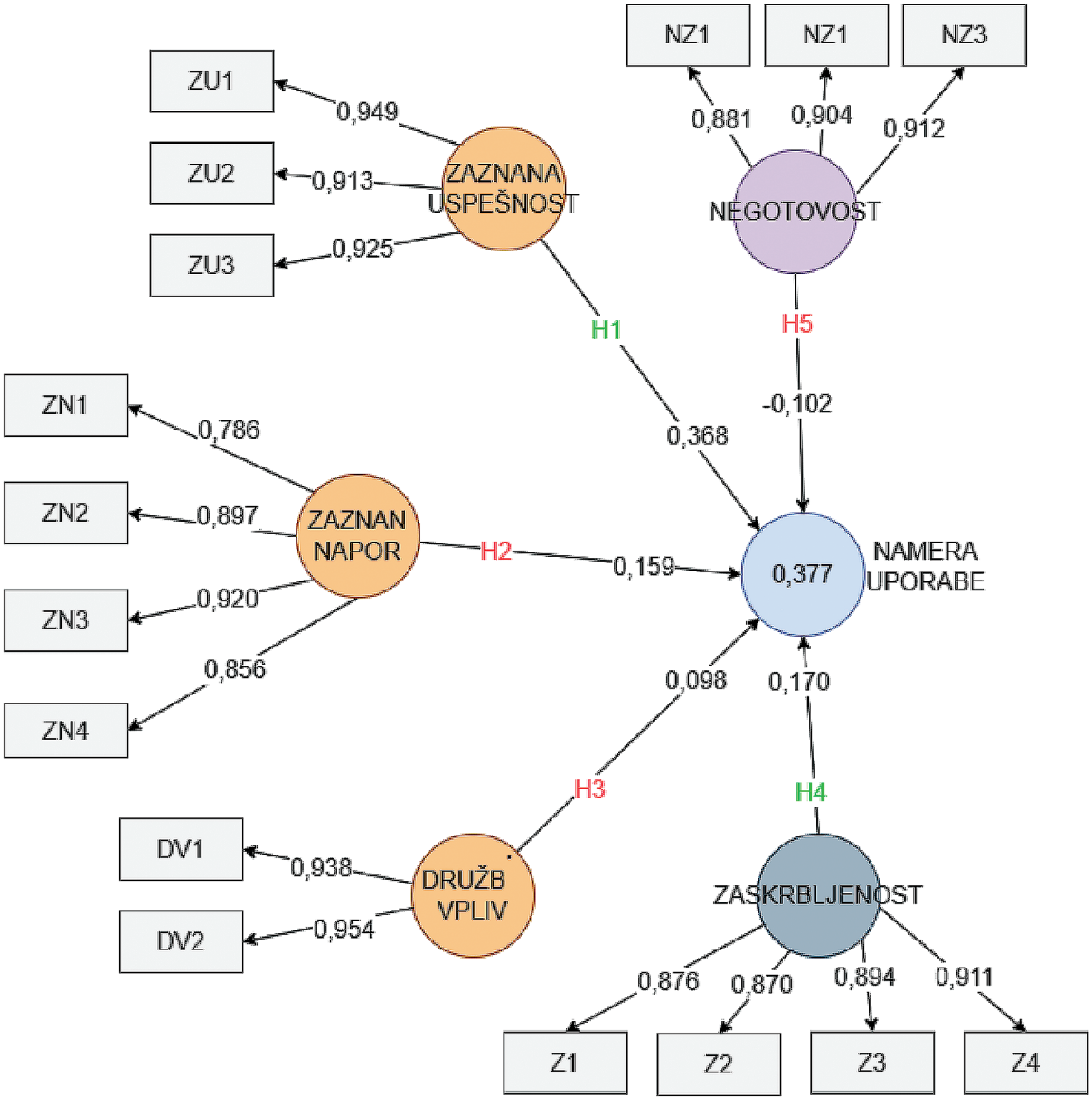
Nazadnje smo preverili hipoteze. Rezultati so predstavljeni v Tabeli 6. Slika 6.5 prikazuje rezultate statistične analize. Hipotezi, označeni z zeleno (H1 in H4), smo statistično potrdili, rdečih pa ne (H2, H3, H5). Rezultate smo grafično ponazorili v Sliki 2, medtem ko jih diskutiramo v naslednjem poglavju.

6 DISKUSIJA

Glede na rezultate analize lahko potrdimo H1 (p = 0,000), saj zaznana uspešnost pozitivno in statistično značilno vpliva na namero uporabe APS. To je v skladu z rezultati raziskave o nameri uporabe platforme spletnega učenja, ki je bila izvedena leta 2019 s strani Chen in drugih (2021). Analizirali so dokaj podobno tehnologijo – spletno aplikacijo, ki jo uporabnik uporablja z namenom olajšanja dokončanja neke naloge.

Tabela 6: **Vplivi izbranega konstrukta na namero uporabe**

KODA		Koeficient	T-test	P-vrednost	Sp. m. – 2,5 %.	Zg. m. – 97,5 %
H3	DRUŽBENI VPLIV - NAMERA UPORABE	0,098	1,733	0,083	–0,010	0,213
H5	NEGOTOVOST - NAMERA UPORABE	–0,102	1,892	0,059	–0,214	–0,004
H4	ZASKRBLJENOST - NAMERA UPORABE	–0,170	3,029	0,002	–0,282	–0,062
H2	ZAZNAN NAPOR - NAMERA UPORABE	0,159	1,850	0,064	0,003	0,340
H1	ZAZNANA USPEŠ. - NAMERA UPORABE	0,368	3,939	0,000	1,172	0,536



Slika 2: **Rezultat analize PLS-SEM**

Zaznana uspešnost je najpomembnejši dejavnik, ki vpliva na namero uporabe.

Hipoteze H2 o vplivu zaznanega napora na namero uporabe APS nismo potrdili. Podobno je bilo ugotovljeno v študiji Ramírez-Correa in drugih (2023), kjer so preučevali uporabo neke podobne tehnologije med starejšimi. Večina vprašanih v naši anketi je bila prav tako starejših, kar lahko pojasni naš statistično neznačilen rezultat.

Hipoteze H3 o vplivu družbenega vpliva na namero uporabe APS tudi nismo potrdili. Razlog za tak rezultat je lahko v vzorcu, ki je bil po večini sestavljen iz moških. Raziskave so namreč ugotovile, da je družbeni vpliv bolj izstopajoč pri ženskah (Tsourela & Roumeliotis, 2015). Vpliv na ta rezultat ima mogoče tudi izkušenost sodelujočih z ostalimi vrstami tehnologije. Namreč, v naši raziskavi ima največ sodelujočih fakultetno izobrazbo. Lahko sklepamo, da prihajajo iz družbe, ki se ne boji učenja in usvajanja novih veščin. Študija Vannoy & Palvia (2010) je pokazala, da na začetnike pri uporabi tehnologije družbeni vpliv bolj vpliva, medtem ko se izkušeni uporabniki morda bolj zanašajo na lastno mnenje.

Hipotezo H4 ($p = 0{,}002$) potrdimo. Zaskrbljenost negativno vpliva na namero uporabe APS in njen vpliv je statistično značilen. To je v skladu s pričakovanji glede na raziskavo Dönmez-Turan & Kir (2019), ki je zaskrbljenost dodala kot nov dejavnik v modelu TAM in katere rezultati so pokazali srednje močan vpliv zaznanega napora in zaskrbljenosti. Podobna študija je bila izvedena s strani Crespo-Martínez in drugih (2023), ki je pokazala, da stopnja računalniške tesnobe lahko vpliva na odnos in vedenjske namene uporabnikov v kontekstu uporabe sistema ERP.

Hipotezo H5 nismo potrdili. Negotovost zaposlitve nima statistično značilnega vpliva na namero uporabe APS. Razlog za to je lahko stopnja izobrazbe. Raziskava Muñoz De Bustillo & De Pedraza (2010) kaže, da se negotovost zaposlitve zmanjšuje s šolanjem, kar kaže, da so lahko višje stopnje izobrazbe povezane z manjšo negotovostjo zaposlitve. Drugi razlog bi lahko bil ta, da je bilo v času raziskave na trgu dela v Sloveniji veliko povpraševanja po delovni sili.

7 ZAKLJUČEK

Rezultati naše raziskave imajo praktičen doprinos, saj lahko pomagajo vsem podjetjem, ki se želijo bolje razumeti kaj pozitivno in kaj negativno vpliva na privzemanje aplikacije APS med ciljnimi uporabniki. Ugotovitve namreč kažejo, da je ključno poskrbeti za dvig percepcije zaposlenega o njegovi osebni delovni uspešnosti ob uporabi nove aplikacije in za zmanjšanje njegove morebitne tesnobe, bojazni, strahu in nelagodja zaradi vpeljave nove tehnologije.

Teoretičen doprinos raziskave je evalvacija vpliva univerzalnih UTAT dejavnikov tehnologije APS in razširitev njenega modela z novimi dejavniki. Raziskava je preverila vplive dejavnikov na namero uporabe aplikacije za napredno planiranje in terminiranje v Sloveniji. V ta namen je bila izvedena spletna anketa, ki je bila posredovana zaposlenim v proizvodnih podjetjih na različnih delovnih mestih. Odgovorili smo na raziskovalni vprašanji. Prvo se glasi: Katere univerzalni UTAUT dejavniki vplivajo na namero uporabe APS? V analizi smo ugotovili, da zaznana uspešnost vpliva statistično značilno na namero uporabe APS, medtem ko zaznan napor in družbeni vpliv pa ne.

Drugo raziskovalno vprašanje se glasi: Ali zaskrbljenost in negotovost zaposlitve pri prevzemanju tovrstnih aplikacij relevantno vplivata na namero uporabe APS? S pomočjo empirične raziskave smo pridobili odgovor, da zaskrbljenost statistično značilno vpliva na namero uporabe APS. Za negotovost zaposlitve pa tega ne moremo trditi.

Raziskava ima nekaj omejitev. Bilo je povabljenih 6000 ljudi zaposlenih na delovnih mestih v proizvodnih podjetjih lociranih izključno v Sloveniji. Izvedli smo spletno raziskavo in spletna izvedba ankete ima to slabost, da se osredotoča izključno na ljudi, ki so pogosto na internetu in so tako bolj naklonjeni uporabi novih tehnologij. Tako smo morebiti zgrešili segment potencialnih uporabnikov APS, ki interneta v službi ne uporabljajo in zato niso bili povabljeni k sodelovanju v anketi.

Prihodnost sistemov APS sistemov bodo oblikovali napredek v tehnologiji, analitika velikih podatkov in računalništvo v oblaku. Te tehnologije bodo sistemom APS omogočile zbiranje in analizo podatkov iz širšega nabora virov, kar bo zagotovilo celovitejši in natančnejši vpogled v proizvodne operacije (Khan, 2022). Uporaba teh tehnologij bo mogoče spodbudila drugačne strahove ali zadržke pri uporabi, kar je priložnost za nadaljnja raziskovalna dela.

ZAHVALA

Raziskovalno delo soavtorice Marine Trkman je financirala Agencija za raziskovalno dejavnost Repu-

blike Slovenije (ARIS) v okviru raziskovalnega projekta J7-50185. Omenjena soavtorica deluje v okviru programske skupine P5-0399.

LITERATURA

[1] Adikoeswanto, D., Eliyana, A., Syamsudin, N., Budiyanto, S., Arief, Z., & Anwar, A. (2022). The mediation role of adoption readiness on perceived anxiety and attitude toward using database management system at correctional institutions. *Heliyon*, 8(8). https://doi.org/10.1016/j.heliyon.2022.e10027

[2] Chen, M., Wang, X., Wang, J., Zuo, C., Tian, J., & Cui, Y. (2021). Factors Affecting College Students' Continuous Intention to Use Online Course Platform. *SN Computer Science*, 2(2). https://doi.org/10.1007/s42979-021-00498-8

[3] Chua, P. Y., Rezaei, S., Gu, M. L., Oh, Y. M., & Jambulingam, M. (2018). Elucidating social networking apps decisions: Performance expectancy, effort expectancy and social influence. *Nankai Business Review International*, 9(2), 118–142. https://doi.org/10.1108/NBRI-01-2017-0003

[3] Collins, L. M. (2007). Research Design and Methods. V J. E. Birren (Ur.), *Encyclopedia of Gerontology* (Second Edition) (str. 433–442). Elsevier. https://doi.org/https://doi.org/10.1016/B0-12-370870-2/00162-1

[4] Commer, P. J., Sci, S., Sair, S. A., & Danish, R. Q. (2018). Effect of Performance Expectancy and Effort Expectancy on the Mobile Commerce Adoption Intention through Personal Innovativeness among Pakistani Consumers. V *Pakistan Journal of Commerce and Social Sciences* (Let. 12, Številka 2).

[5] Crespo-Martínez, E., Astudillo-Rodríguez, C., Chica-Contreras, G., & Vásquez-Aguilera, A. (2023). Technology Acceptance Model of ERP software in Small Business: A Systematic Literature review. *Enfoque UTE*, 14(1), 46–61. https://doi.org/https://doi.org/10.29019/enfoqueute.884

[6] Dönmez-Turan, A., & Kir, M. (2019). User anxiety as an external variable of technology acceptance model: A meta-analytic study. *Procedia Computer Science*, 158, 715–724. https://doi.org/10.1016/j.procs.2019.09.107

[7] Fleischmann, B., Meyr, H., & Wagner, M. (2008). Advanced Planning. V H. Stadtler & C. Kilger (Ur.), *Supply chain management and advanced planning (Fourth edition): Concepts, models, software, and case studies* (4. izd., str. 81–106). Springer. https://doi.org/10.1007/978-3-540-74512-9

[8] Gander, P. (2023). *The Manufacturing Evolution from Industry 1.0 to industry 5.0.* https://www.assuredpartners.com/blogs/manufacturing/2023/the-manufacturing-evolution-from-industry-1-to-industry-5/

[9] Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). *Multivariate Data Analysis*. Cengage. https://books.google.si/books?id=0R9ZswEACAAJ

[10] Hair, J. F., Ringle, C. M., & Sarstedt, M. (2011). PLS-SEM: Indeed a silver bullet. *Journal of Marketing Theory and Practice*, 19(2), 139–152. https://doi.org/10.2753/MTP1069-6679190202

[11] Hair, J. F., Sarstedt, M., Ringle, C. M., & Mena, J. A. (2012). An assessment of the use of partial least squares structural equation modeling in marketing research. *Journal of the Academy of Marketing Science*, 40(3), 414–433. https://doi.org/10.1007/s11747-011-0261-6

[12] Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., Sarstedt, M., Danks, N. P., & Ray, S. (2021). *Classroom Companion: Business Partial Least Squares Structural Equation Modeling* (PLS-SEM) Using R A Workbook (3rd izd.). Springer.

[13] Hvolby, H. H., & Steger-Jensen, K. (2010). Technical and industrial issues of Advanced Planning and Scheduling (APS) systems. *Computers in Industry*, 61(9), 845–851. https://doi.org/10.1016/j.compind.2010.07.009

[14] Jenkins, A. (2022, avgust 23). What *Is Production Planning & Why Is It Important?* https://www.netsuite.com/portal/resource/articles/inventory-management/production-planning.shtml#:~:text=Production%20planning%20helps%20companies%20build,adjust%20operations%20when%20problems%20occur.

[15] Kenton, W. (2022). *Manufacturing: Definition, Types, Examples, and Use as Indicator.* https://www.investopedia.com/terms/m/manufacturing.asp

[16] Khan, J. (2022, september 13). *The evolution and future of Advanced Planning and Scheduling (APS) systems*. https://blogs.sap.com/2022/09/13/the-evolution-and-future-of-advanced-planning-and-scheduling-aps-systems/

[17] Kulviwat, S., Bruner, G. C., & Al-Shuridah, O. (2009). The role of social influence on adoption of high tech innovations: The moderating effect of public/private consumption. *Journal of Business Research*, 62(7), 706–712. https://doi.org/10.1016/j.jbusres.2007.04.014

[18] Liu, Y., Lu, X., Zhao, G., Li, C., & Shi, J. (2022). Adoption of mobile health services using the unified theory of acceptance and use of technology model: Self-efficacy and privacy concerns. *Frontiers in Psychology*, 13. https://doi.org/10.3389/fpsyg.2022.944976

[19] Mohamad, Z. B., Kamarozaman, Z. B., Kassim, M. F. R. B., & Razak, F. Z. A. (2021). Does social influence affect continuance intention to use e-campus? A Study in Malaysian private higher institution. *Journal of Physics: Conference Series*, 1793(1). https://doi.org/10.1088/1742-6596/1793/1/012008

[20] Muñoz De Bustillo, R., & De Pedraza, P. (2010). Determinants of Job Insecurity in 5 European Countries.

[21] Naser Alraja, M., Hammami, S., Chikhi, B., & Fekir, S. (2016). International Review of Management and Marketing The Influence of Effort and Performance Expectancy on Employees to Adopt E-government: Evidence from Oman. *International Review of Management and Marketing*, 6(4), 930–934. http:www.econjournals.com

[22] Ramírez-Correa, P., Grandón, E. E., Ramírez-Santana, M., Arenas-Gaitán, J., & Rondán-Cataluña, F. J. (2023). Explaining the Consumption Technology Acceptance in the Elderly Post-Pandemic: Effort Expectancy Does Not Matter. *Behavioral Sciences*, 13(2). https://doi.org/10.3390/bs13020087

[23] RVJ. (2023, maj 24). *The Benefits of Implementing Advanced Planning and Scheduling (APS) Systems*. https://www.deskera.com/blog/advanced-production-planning/

[24] Saidi, S. S., & Siew, N. M. (2019). Investigating the Validity and Reliability of Survey Attitude towards Statistics Instrument among Rural Secondary School Students. *International Journal of Educational Methodology*, 5(4), 651–661. https://doi.org/10.12973/ijem.5.4.651

[25] Saprikis, V., Avlogiaris, G., & Katarachia, A. (2021). Determinants of the intention to adopt mobile augmented reality apps in shopping malls among university students. *Journal of Theoretical and Applied Electronic Commerce Research*, 16(3), 491–512. https://doi.org/10.3390/jtaer16030030

[26] Shanmugam, B., Rabiah, A., Grima, S., Mensah, I. K., & Zeng, G. (2022). *The behavioral intention to adopt mobile health services: The moderating impact of mobile self-efficacy*. https://doi.org/10.3389/fpubh.2022.1020474

[27] Siva, A. (2022, november 15). *Why an APS Solution is Key to Meeting Manufacturing Demands*. https://www.g2.com/articles/aps

[28] Steger-Jensen, K., Hvolby, H. H., Nielsen, P., & Nielsen, I. (2011). Advanced planning and scheduling technology. *Pro-*

duction Planning and Control, 22(8), 800–808. https://doi.org/10.1080/09537287.2010.543563
[29] Trkman, M., Popovič, A., & Trkman, P. (2023). The roles of privacy concerns and trust in voluntary use of governmental proximity tracing applications. *Government Information Quarterly,* 40(1). https://doi.org/10.1016/j.giq.2022.101787
[30] Tsourela, M., & Roumeliotis, M. (2015). The moderating role of technology readiness, gender, and sex in consumer acceptance and actual use of Technology-based services. *The Journal of High Technology Management Research*, 26(2), 124–136. https://doi.org/https://doi.org/10.1016/j.hitech.2015.09.003
[31] ander Elst, T., De Witte, H., & De Cuyper, N. (2014). The Job Insecurity Scale: A psychometric evaluation across five European countries. *European Journal of Work and Organizational Psychology*, 23(3), 364–380. https://doi.org/10.1080/1359432X.2012.745989
[32] Vannoy, S. A., & Palvia, P. (2010). The Social Influence Model of Technology Adoption. Communications of the ACM, 53(6), 149–153. https://doi.org/10.1145/nnnnnn.nnnnnn
[33] Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. V *Quarterly* (Let. 27, Številka 3).
[34] Venkatesh, V., Thong, J. Y. L., & Xu, X. (2016). Unified Theory of Acceptance and Use of Technology: A Synthesis and the Road Ahead. J ournal. http://ais.site-ym.com/?SeniorScholarBasket
[35] Venkatesh, V., Thong, J. Y. L., Xu, X., & Walton, S. M. (2012). Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology Quarterly Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology1. V Source: *MIS Quarterly* (Let. 36, Številka 1).
[36] Wang, P. X., Kim, S., & Kim, M. (2023). Robot anthropomorphism and job insecurity: The role of social comparison. *Journal of Business Research*, 164. https://doi.org/10.1016/j.jbusres.2023.114003
[37] Wu, W., Zhang, B., Li, S., & Liu, H. (2022). Exploring Factors of the Willingness to Accept AI-Assisted Learning Environments: An Empirical Investigation Based on the UTAUT Model and Perceived Risk Theory. *Frontiers in Psychology*, 13. https://doi.org/10.3389/fpsyg.2022.870777
[38] Zuiderwijk, A., Janssen, M., & Dwivedi, Y. K. (2015). Acceptance and use predictors of open data technologies: Drawing upon the unified theory of acceptance and use of technology. *Government Information Quarterly*, 32(4), 429–440. https://doi.org/10.1016/j.giq.2015.09.005

▪

Nika Kalan je magistrica družboslovne informatike na Fakulteti za družbene vede Univerzi v Ljubljani. Magistrirala je z delom Dejavniki vpliva na prevzemanje aplikacij za napredno planiranje in terminiranje proizvodnje. Zaposlena je v Result-u, kjer deluje na področju digitalizacije in optimizacije proizvodnje ter podatkovni arhitekturi.

▪

Doc. dr. Marina Trkman poučuje informatiko na Fakulteti za upravo Univerzi v Ljubljani. Raziskovalno se ukvarja: 1) s preiskovanjem dejavnikov, ki vplivajo na privzemanje informacijsko komunikacijskih tehnologij in 2) z analizranjem poslovnih procesov za optimizacijo poslovanja. Kot prvi avtor je objavila članke v prestižnih revijah kot sta International journal of information management in pa Government information quarterly.

Ranljivosti v programih zaradi dvojnega sproščanja pomnilnika

Marin Gazvoda de Reggi, Matevž Pesek
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana
mg4234@student.uni-lj.si, matevz.pesek@fri.uni-lj.si

Izvleček

V računalništvu je učinkovito upravljanje pomnilnika ključno za delovanje programske opreme. Napake pri tem lahko vodijo do resnih varnostnih ranljivosti, ki omogočajo izvajanje poljubne kode ali krajo občutljivih podatkov. Prispevek obravnava podrobnosti napada na podlagi dvojnega sproščanja pomnilnika (*double free*) in prikazuje primer ranljivosti v programu za upravljanje podatkovne baze. Pojasni, kako lahko napadalec pridobi administratorske pravice brez gesla. Predlagane rešitve vključujejo uporabo pomnilniško varnih jezikov, orodij za statično analizo, omejevanje privilegijev in defenzivno programiranje. Poudarjen je pomen celovitega pristopa k varovanju programske opreme pred takimi napadi.

Ključne besede: dvojno sproščanje, napad, upravljanje pomnilnika, varnostne ranljivosti

BINARY VULNERABILITIES DUE TO DOUBLE FREE

Abstract

In computer science, effective memory management is crucial for software performance. Mistakes in this area can lead to serious security vulnerabilities, such as arbitrary code execution or theft of sensitive data. This paper discusses the details of a double-free memory attack and illustrates an example of a vulnerability in a database management program. It explains how an attacker can gain administrative rights without a password. Proposed solutions include using memory-safe languages, static analysis tools, privilege restriction, and defensive programming. The importance of a comprehensive approach to protecting software from such attacks is emphasized.

Key words: double free, attack, memory management, security vulnerabilities

1 UVOD

V sodobni digitalni dobi so računalniški sistemi postali nepogrešljiv del naše družbe, prisotni v vseh vidikih življenja in delovanja. Programska oprema, ki poganja te sisteme, igra ključno vlogo pri upravljanju procesov od osebnih naprav do kompleksne infrastrukture. Ta vseprisotnost pa prinaša nove izzive, predvsem na področju varnosti in zanesljivosti.

Z naraščajočo odvisnostjo od digitalnih sistemov postajajo posledice varnostnih ranljivosti vse resnejše. Vdori lahko vodijo do kraje podatkov, finančnih izgub, ogrožanja zasebnosti in motenj v delovanju kritične infrastrukture. Te grožnje niso več omejene le na specializirane skupine, temveč postajajo dostopne širšemu krogu potencialnih napadalcev zaradi razširjenosti orodij in znanja o izkoriščanju ranljivosti.

Eno ključnih področij, ki zahteva posebno pozornost, je upravljanje pomnilnika. To je še posebej pomembno v jezikih z ročnim upravljanjem, kot sta C in C++, ki se pogosto uporabljajo za razvoj sistemske programske opreme zaradi hitrosti in učinkovitosti. Vendar pa lahko ravno ta fleksibilnost vodi do subtilnih napak z resnimi varnostnimi posledicami.

Med najpogostejšimi napakami pri upravljanju pomnilnika so dvojno sproščanje (*double free*), uporaba blokov po sprostitvi (*use-after-free*) in uhajanje pomnilnika (*memory leak*). Napadalci lahko te napake izkoristijo za izvajanje poljubne kode, pridobivanje občutljivih podatkov ali destabilizacijo sistema. Kljub dolgoletnemu zavedanju o teh težavah ostajajo te ranljivosti razširjene in predstavljajo izziv za varnost programske opreme.

V tem članku se osredotočamo na analizo napada, ki izkorišča ranljivost dvojnega sproščanja pomnilnika. Za ilustracijo predstavljamo primer v programu, ki simulira preprostega upravitelja podatkovne baze, in pokažemo, kako lahko napadalec izkoristi to navidez nedolžno napako za pridobitev administratorskih pravic brez poznavanja gesla.

Razumevanje mehanizma teh napadov je ključno za razvoj varne programske opreme. S poglobljenim vpogledom lahko razvijalci bolje razumejo potencialne grožnje in razvijejo učinkovitejše strategije za njihovo preprečevanje. Takšna analiza pomaga tudi pri oblikovanju boljših orodij za odkrivanje in preprečevanje tovrstnih ranljivosti.

Cilj tega članka je dvojen: prispevati k boljšemu razumevanju mehanizma napadov, ki izkoriščajo ranljivosti pri upravljanju pomnilnika, in spodbuditi razvoj robustnejših pristopov k programiranju in varnostnim praksam. S tem želimo prispevati k razvoju varnejših in zanesljivejših računalniških sistemov, ki bodo kos varnostnim izzivom v vedno bolj digitaliziranem svetu.

2 SORODNA DELA

V zadnjih letih je področje varnosti računalniških sistemov doživelo porast raziskav, osredotočenih na ranljivosti pri upravljanju pomnilnika. Te študije se v glavnem ukvarjajo z vzroki, klasifikacijo in zaznavanjem ranljivosti, pri čemer posebno pozornost namenjajo trem ključnim problemom: uhajanju pomnilnika, dvojnemu sproščanju in uporabi pomnilniških blokov po njihovi sprostitvi. V nadaljevanju predstavljamo pregled ključnih prispevkov na tem področju, ki skupaj tvorijo celovito sliko trenutnega stanja raziskav.

Temeljna študija Caballera in sodelavcev je poudarila povezavo med ustvarjanjem in uporabo visečih kazalcev ter ranljivostmi zaradi uhajanja in dvojnega sproščanja pomnilnika. Njihovo orodje "Undangle" omogoča zgodnje odkrivanje teh ranljivosti in je bilo uspešno ovrednoteno na osmih praktičnih primerih, vključno z dvema novima ranljivostma v spletnem brskalniku Firefox [1]. Ta raziskava je spodbudila razvoj sorodnih orodij, kot so FreeSentry [2], DangSan [3], HeapExpo [4] in pSweeper [5], ki so dodatno izboljšala analizo visečih kazalcev.

Na področju odkrivanja varnostnih ranljivosti pri upravljanju pomnilnika so raziskovalci razvili različne pristope, ki združujejo statično analizo in dinamično testiranje. Hu in sodelavci so v tem kontekstu formalno opredelili tovrstne ranljivosti in razvili ogrodje "MRVDAVF" za analizo izvorne kode. Njihov pristop se je izkazal za učinkovitejšega v primerjavi z obstoječimi rešitvami [6]. Komplementarno temu so Rebel in sodelavci predlagali modularni pristop za samodejno izkoriščanje in iskanje ranljivosti pri upravljanju kopice [7], s čimer so razširili nabor orodij za celovito analizo varnosti pomnilnika.

Pomemben napredek na področju dinamičnega testiranja predstavlja orodje "UAFL", ki so ga razvili Wang in sodelavci. To orodje za *fuzz* testiranje temelji na analizi stanja tipov (angl. *typestate analysis*) in se je izkazalo za posebej učinkovito pri odkrivanju ranljivosti zaradi uporabe pomnilniških blokov po sprostitvi [8]. Na tem področju so Yan in sodelavci naredili dodaten korak z integracijo strojnega učenja v statično analizo, kar je privedlo do izboljšane natančnosti pri odkrivanju ranljivosti [9].

V sklopu dinamične analize so Gui in sodelavci razvili odprtokodno orodje "UAFSan", ki vsakemu pomnilniškemu bloku dodeli edinstveno oznako. S preverjanjem konsistence teh oznak so avtorji uspešno identificirali napake zaradi uporabe blokov po sprostitvi [10]. Sorodno temu pristopu so Erdő s in sodelavci predstavili orodje "MineSweeper", ki deluje na principu karantene sproščenih pomnilniških blokov. To orodje preprečuje ponovno alokacijo bloka, dokler se ne potrdi odsotnost kazalcev nanj, s čimer učinkovito preprečuje ranljivosti pri uporabi pomnilniških blokov po sprostitvi [11].

Raziskave so se usmerile tudi v preučevanje tehnik enkratnega dodeljevanja (OTA) istega bloka po-

mnilnika kot preventivnega ukrepa proti omenjenim ranljivostim. V tem kontekstu so Wickman in sodelavci razvili pomnilniški dodeljevalec "FFmalloc", ki optimizira določene pomanjkljivosti enkratnih dodeljevalcev pomnilnika in s tem poveča njihovo praktično uporabnost [12].

Za celovit pregled področja so Gui in sodelavci opravili sistematično analizo, primerjavo in ovrednotenje trenutnih tehnik za odkrivanje in preprečevanje ranljivosti zaradi uporabe pomnilniških blokov po sprostitvi. Njihova študija vključuje primerjavo učinkovitosti izvajanja in porabe pomnilnika različnih tehnik [13], s čimer ponuja dragocen vpogled v kompromise med varnostjo in zmogljivostjo. Nadaljnje raziskave so se usmerile tudi v specifične domene, kot so industrijski nadzorni sistemi (ICS), kjer so Liu in sodelavci opravili pregled ranljivosti v izvršljivih datotekah [14], s čimer so poudarili pomen varnosti pomnilnika v kritičnih infrastrukturnih sistemih.

Ta pregled sorodnih del kaže na kompleksnost in večplastnost problematike varnosti pomnilnika ter poudarja potrebo po celovitem pristopu k odkrivanju in preprečevanju ranljivosti. Raziskave kažejo trend k razvoju vse bolj sofisticiranih orodij in tehnik, ki združujejo statično in dinamično analizo, strojno učenje ter inovativne pristope k upravljanju pomnilnika. Kljub pomembnemu napredku pa ostaja področje varnosti pomnilnika aktualen izziv, ki zahteva nadaljnje raziskave in razvoj.

3 METODOLOGIJA

3.1 Raziskovalni pristop

V tej raziskavi smo uporabili kvalitativni raziskovalni pristop, ki temelji na kombinaciji študije primera in analize tveganja. Naš metodološki okvir je zasnovan tako, da omogoča celovito razumevanje problematike dvojnega sproščanja pomnilnika, od teoretičnih osnov do praktičnih implikacij in možnih rešitev. Raziskava je potekala v štirih ključnih fazah: priprava, analiza, validacija in sinteza.

V fazi priprave smo razvili preprost program, ki simulira upravitelja podatkovne baze, z namerno vgrajeno ranljivostjo dvojnega sproščanja pomnilnika. Ta namenski primer nam je služil kot osnova za podrobno preučevanje mehanizma ranljivosti v kontroliranem okolju.

Sledila je faza analize, v kateri smo korak za korakom preučili, kako lahko napadalec izkoristi ranljivost za pridobitev nepooblaščenega dostopa. Ta del je vključeval podroben pregled dogajanja v pomnilniku med izvajanjem programa, kar nam je omogočilo globlje razumevanje tehničnih vidikov ranljivosti. Analitična stopnja raziskave je bila ključna za razkritje subtilnih mehanizmov, ki omogočajo izkoriščanje te vrste ranljivosti.

V fazi validacije smo našo teoretično analizo nadgradili s študijo primera iz prakse. Preučili smo konkreten primer ranljivosti v široko uporabljeni mobilni aplikaciji, kar je služilo kot most med našo teoretično analizo in praktičnimi implikacijami. Ta korak je bil ključen za potrditev relevantnosti naše raziskave v kontekstu kompleksnih produkcijskih sistemov.

Zaključna faza sinteze je bila namenjena oblikovanju nabora strokovnih priporočil za preprečevanje in ublažitev tovrstnih ranljivosti. Na podlagi ugotovitev iz predhodnih delov raziskave smo razvili praktične smernice, ki razvijalcem in varnostnim strokovnjakom ponujajo konkretne napotke za izboljšanje varnosti programske opreme.

Naš večstopenjski pristop, ki združuje nadzorovan eksperiment in analizo realnega primera, zagotavlja ravnovesje med natančnostjo laboratorijske analize in relevantnostjo za resnične scenarije v razvoju programske opreme. S tem smo dosegli celovito obravnavo problematike, ki presega zgolj teoretično razumevanje in ponuja praktične vpoglede v varnostne izzive sodobnega razvoja programske opreme.

3.2 Ranljivosti v izvršljivih datotekah

Izvršljive datoteke so datoteke, ki jih lahko računalnik izvede (izvrši). V operacijskem sistemu Linux so običajno zapisane v formatu ELF (*Executable and Linkable Format*). Sestavljene so iz več različnih sekcij, ki vsebujejo strojno kodo, podatke, simbole in druge metapodatke [15]. Ko uporabnik zažene izvršljivo datoteko, operacijski sistem prebere metapodatke in naloži kodo v pomnilnik. Koda se po tem začne izvajati (proces).

Proces je osnovna enota izvajanja v operacijskem sistemu. Vsak izmed njih ima svoj ločen prostor v pomnilniku, kjer shranjuje podatke, potrebne za izvajanje. Podatki, ustvarjeni v času izvajanja, so običajno ločeni na sklad (angl. *Stack*), kjer se nahajajo lokalne spremenljivke in naslovi za vrnitve iz funkcij, in kopico (angl. *Heap*), ki se uporablja za dinamično dodeljevanje pomnilnika.

Procesi, ki so ranljivi zaradi napak v programski kodi, so pogosto tarča napadov. Med najpogostejši-

mi ranljivostmi v izvršljivih datotekah so prelivanje medpomnilnika (angl. *Buffer overflow*), ranljivosti v knjižnicah (angl. *Library vulnerabilities*) in ranljivosti pri upravljanju pomnilnika (angl. *Memory management vulnerabilities*).

Prav tako lahko v grobem ločimo napade na sklad in kopico. V nadaljevanju bomo podrobneje obravnavali napade na kopico, ki izkoriščajo ranljivosti pri upravljanju pomnilnika.

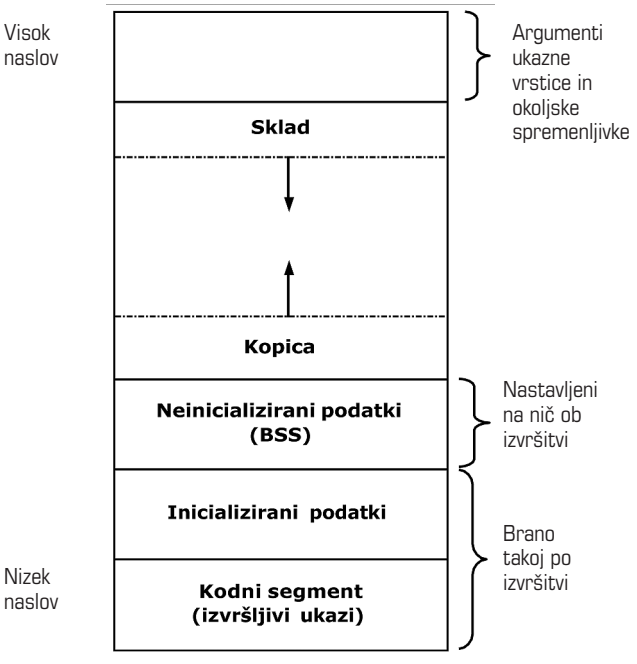
3.3 Osnovni opis kopice

Kopica (angl. *Heap*) je prilagodljivo območje pomnilnika za hranjenje večjih podatkovnih struktur in podatkov z dinamično življenjsko dobo. Za razliko od lokalnega pomnilnika, ki se samodejno dodeli in sprosti, je treba s kopico upravljati eksplicitno. V jezikih, kot sta Java ali C++, se pri ustvarjanju struktur ali objektov to običajno izvede z uporabo operatorja new. V programskem jeziku C pa se za dinamično dodelitev pomnilnika uporablja funkcija malloc().

Dodeljen blok pomnilnika (ali objekt) ostane v uporabi, dokler ni eksplicitno sproščen. V nižjenivojskih programskih jezikih to nalogo prevzema programer. Takšen pristop programerju omogoča večji nadzor nad upravljanjem pomnilnika, a hkrati nalaga večjo odgovornost za aktivno skrb zanj. Ena pogostejših napak pri tem je ohranitev reference na pomnilniško lokacijo brez ustreznega sproščanja. Temu rečemo puščanje pomnilnika (angl. *memory leak*).

V mnogih komercialnih programih, napisanih v C ali C++, se pojavlja puščanje pomnilnika, ki povzroči pomanjkanje prostega pomnilnika in sesutje programa. Java in drugi nekoliko višjenivojski jeziki to napako odpravljajo s pomočjo avtomatskega upravljanja oz. čiščenja pomnilnika (angl. *garbage collection*). Slabost tega pristopa pa je, da čiščenje pomnilnika nekoliko upočasni delovanje programa ter se zgodi v nepredvidljivih časih [17].

V operacijskem sistemu Linux je za upravljanje kopice v programskem jeziku C zadolžena knjižnica GNU libc.



Slika 1: **Shematičen prikaz pomnilnika procesa [16].**

3.4 Delovanje funkcij malloc() in free()

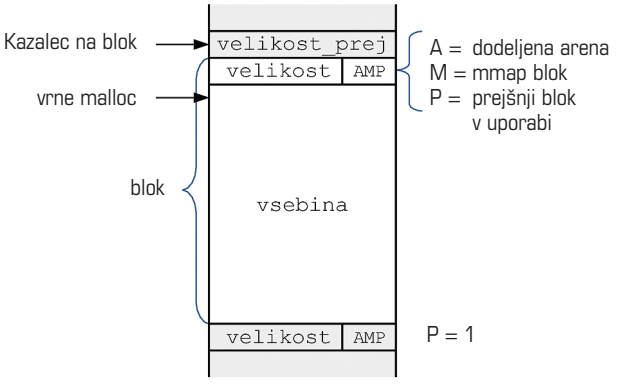
3.4.1 Bloki

Knjižnica GNU libc deli kopico na bloke različnih velikosti. Vsak blok vsebuje metapodatke o velikosti in o sosednjih blokih. Ko je blok v uporabi, se v pomnilniku hranita le njegova velikost in zastavice, ko pa je sproščen, pa se poleg tega v pomnilnik zapišeta še kazalca na sosednja bloka [18]. Strukturo dodeljenega in sproščenega bloka prikazujeta sliki 2 in 3.

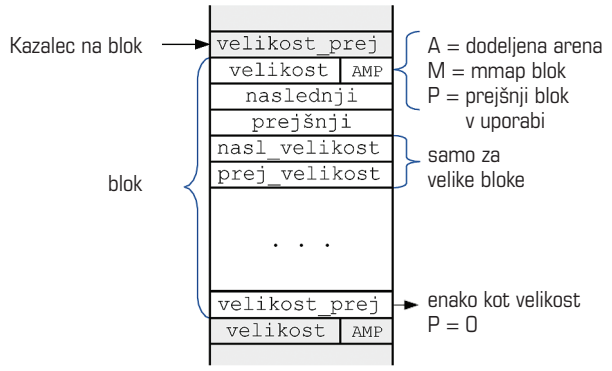
3.4.2 Koši

Sproščeni bloki so shranjeni v različnih seznamih – koših (angl. *bins*) glede na velikost in zgodovino, da jih lahko knjižnica učinkovito ponovno dodeli ob novi zahtevi. Koši so štirih vrst: hitri, nerazvrščeni, majhni in veliki.

Pri večnitnih procesih ima vsaka nit svoj lokalni predpomnilnik blokov, dostopnih brez zaklepanja (angl. *tcache*) [18].



Slika 2: **Blok v uporabi [18].**



Slika 3: **Sproščen blok [18].**

3.4.3 Zaznavanje okvar kopice

Sistem za alokacijo in dealokacijo pomnilnika sproti preverja za morebitne okvare, vendar je večina pregledov hevrističnih (npr. preverjanje kazalcev) in se jih da pretentati z lažnimi bloki, ki izgledajo resnično. V tem primeru lahko okvara preživi kar nekaj časa, ne da bi bila zaznana [18].

3.5 Dvojno sproščanje pomnilnika

Ena izmed pogostih napak, ki se pojavijo pri programiranju v jezikih z ročnim upravljanjem pomnilnika, je dvojno sproščanje pomnilnika (angl. *double free*). Pojavi se, ko program (po pomoti) dvakrat kliče funkcijo free() z istim argumentom (kazalcem na dodeljeni blok pomnilnika). To privede do korupcije podatkovnih struktur za upravljanje pomnilnika, kar lahko povzroči, da program preneha delovati ali pa v nekaterih okoliščinah spremeni tok izvajanja [19]. Napadalec lahko s prepisovanjem pomnilniških prostorov pripravi program, da izvede skorajda poljuben kos kode, kar mu omogoča pridobitev dostopa do lupine.

Poglejmo enostaven primer ranljivosti, ki se pojavi zaradi dvojnega sproščanja pomnilnika:

```
void *ptr = malloc(SIZE);
...
if (some_error) {
    free(ptr);
}
...
free(ptr);
```

Pogosta vzroka ranljivosti sta obravnava napak in drugih izjemnih okoliščin ter nejasnost, kateri del programa je odgovoren za sproščanje pomnilnika. Čeprav nekatere ranljivosti niso veliko bolj zapletene od zgornjega primera, so večinoma razpršene po stotinah vrstic kode ali celo po različnih datotekah, kar močno zmanjšuje preglednost in otežuje sledenje toku izvajanja programa. Ta problem je še posebej izrazit pri uporabi globalnih spremenljivk [19].

4 PRIMER IN RAZLAGA NAPADA

V tem delu bomo podrobneje opisali potek in specifike napada na primeru enostavnega programa, napisanega v programskem jeziku C [20].

Program deluje interaktivno v ukazni vrstici po principu REPL (*Read–Eval–Print Loop*) in simulira preprostega upravitelja podatkovne baze. Razlikuje med dvema uporabniškima vlogama: uporabnikom in administratorjem. Uporabnik lahko izvaja poizvedbe na bazi, medtem ko ima administrator poleg te možnosti tudi pravico vnašati nove vnose in brisati vsebino baze.

Za dostop do administratorskih funkcij je potrebna prijava z geslom, ki ga pozna le administrator, zato navadni uporabnik kot tudi napadalec brez gesla ne moreta dostopati do teh funkcij.

Poglejmo si poziv, ki ga vidi uporabnik:

```
Logged in as: user
1) Quit
2) Change <user>
3) Query <something|*>
Enter your choice:
```

Če uporabnik želi z vnosom 2 admin spremeniti uporabniško vlogo in ob tem vpiše napačno geslo, se mu kot pričakovano izpiše sporočilo:

```
Incorrect password!
```

Rezultat veljavne poizvedbe npr. 3 Jack je sledeč:

```
 id | name
----+----------------
  9 | Jack
Found 1 entry.
```

Če pa je poizvedba neveljavna, npr. za ukazom 3 ne podamo argumenta, se izpiše sporočilo:

```
Invalid query.
```

Poglejmo si poenostavljen del kode, ki je odgovoren za obravnavanje poizvedb:

```
void select_from_db(char *line) {
  // line <- "3 Jack"
  DBEntry *e = malloc(sizeof(DBEntry));

  char query[SIZE];
  int args;
  args = sscanf(line, "%*s %s", query);
  if (args != 1) {
    puts("Invalid query.");
    free(line);
    free(e);
    return;
  }
  // query <- "Jack"

  int count = 0;

  ... // print matching entries

  printf("Found %d entries.\n", count);

  free(e);
}
```

Funkcija select_from_db() kot parameter prejme dinamično dodeljen niz znakov line, ki predstavlja vrstico z ukazom. Za dodelitev in sprostitev tega niza sicer skrbi klicatelj funkcije, vendar pa se v primeru neveljavne poizvedbe niz sprosti znotraj funkcije. Zato se ob neveljavni poizvedbi ta blok pomnilnika sprosti dvakrat, kar predstavlja varnostno ranljivost.

V našem primeru so alokacije prikladno enakih velikosti. Poglejmo dogajanje v hitrem košu (*fastbin*) pri sproščanju pomnilnika ob neveljavni poizvedbi:

1. Sprosti se line.

   GLAVA -> line -> REP
2. Sprosti se e.

   GLAVA -> e -> line -> REP
3. Izven funkcije se znova sprosti line.

   GLAVA -> line -> e -> line -> REP

Če bi dvakrat zapored sprostili isti blok, bi tudi starejše verzije knjižnice GNU libc zaradi omenjenih hevristik zaznale napako in prekinile izvajanje programa. V tem primeru pa napaka zaenkrat ostaja nezaznavna.

Analizirajmo še funkcijo za prijavo:

```
void change_user(char *line) {
  // line <- "2 admin"
  char *username = malloc(SIZE);
  ... // set: username <- "admin"
  ... // match username

  char *password = malloc(SIZE);
  int fd = open("passwd.txt", O_RDONLY);
  read(fd, password, SIZE);
  close(fd);

  printf("Enter password: ");
  fgets(line, SIZE, stdin);

  if (!strncmp(line, password, SIZE)) {
    uid = ADMIN_UID;
    puts("Switched to admin.");
  } else
    puts("Incorrect password!");

  free(password);
  free(username);
}
```

Če takoj za neveljavno poizvedbo uporabnik poskusi zamenjati uporabniško vlogo, se pri dodeljevanju pomnilnika zgodi sledeče:

1. Še izven funkcije se dodeli line.

`GLAVA -> e -> line -> REP`

Blok line je zdaj obenem dodeljen kot tudi sproščen.

2. Dodeli se username.

`GLAVA -> line -> REP`

V prejšnji funkciji sproščen blok e se zdaj ponovno dodeli kot username.

3. Dodeli se password.

`GLAVA -> REP`

Hitri koš je zdaj prazen. Blok line je zdaj dodeljen kot password. Vendar je to isti blok, ki je že v uporabi. Trenutno oba kazalca, line in password, kažeta na isti kos pomnilnika.

V funkciji change_user() najprej preberemo geslo iz datoteke passwd.txt in ga shranimo v blok password. Za primere demonstracije se geslo hrani kot čistopis v datoteki, v pravih sistemih pa naj bi bilo geslo seveda ustrezno shranjeno in šifrirano.

Nato uporabnika prosimo za vnos gesla. Če se vnos ujema s prebranim geslom, uporabniku dodelimo administratorske pravice. Prebrano geslo se

shrani v blok line. Ker pa gre za isti blok kot pri kazalcu username, se vsebina bloka enostavno prepiše z uporabniškim vnosom. Posledično se primerja isti niz znakov s samim seboj, kar seveda vedno vrne true, zato se uporabniška vloga uspešno zamenja na administratorsko. Pri tem se izpiše sporočilo:

```
Switched to admin.

Logged in as: admin
1) Quit
2) Change <user>
3) Query <something|*>
4) Insert <entry> into database
5) Wipe database
Enter your choice:
```

5 REZULTATI

Analiza primera ranljivosti dvojnega sproščanja pomnilnika v preprostem programu za upravljanje podatkovne baze je razkrila več ključnih ugotovitev:

1. **Uspešno izkoriščanje ranljivosti:**
   Napadalcu je uspelo pridobiti administratorske pravice brez poznavanja gesla. To je bilo doseženo z zaporedjem specifičnih korakov: izvedba neveljavne poizvedbe, ki sproži dvojno sproščanje, in nato poskus zamenjave uporabniške vloge.
2. **Mehanizem napada:**
   Dvojno sproščanje je povzročilo, da sta dva kazalca (line in password) kazala na isti blok pomnilnika. To je omogočilo prepis gesla z uporabniškim vnosom, kar je privedlo do uspešne avtentikacije.
3. **Pogoji za uspešen napad in omejitve zaznavanja:**

- Hevristični varnostni pregledi v knjižnici GNU libc niso zaznali te specifične oblike dvojnega sproščanja, kar kaže na pomanjkljivosti varnostnih mehanizmov pri odkrivanju kompleksnejših vzorcev napačnega upravljanja s pomnilnikom.
- Demonstriran napad je učinkovit na sistemih z GNU libc pred različico 2.26. To vključuje Ubuntu 17.10 in 16.04 LTS, pri čemer je slednja še vedno podprta in prejema razširjene varnostne posodobitve do aprila 2026 [21]. Posledično je prikazan primer napada še vedno aktualen za določene sisteme v uporabi.
- Izdaja GNU libc 2.26 je uvedla podporo za lokalni predpomnilnik blokov za niti (*tcache*), ki onemogoča to specifično obliko napada [22].
- Nadaljnje iteracije knjižnice so izboljšale učinkovitost upravljanja kopice, vendar so hkrati opustile nekatere varnostne mehanizme, kar je odprlo vrata za novejše načine napadov [23].

4. **Širše posledice:**
   Čeprav je bil primer demonstriran na preprostem programu, rezultati kažejo, da lahko podobne ranljivosti v kompleksnejših sistemih vodijo do resnih varnostnih tveganj. Analiza razkriva potrebo po bolj robustnih metodah za preverjanje pravilnosti upravljanja s pomnilnikom v programih, napisanih v jezikih z ročnim upravljanjem pomnilnika.

Za namen demonstracije je bil pripravljen tudi vsebnik Docker, ki simulira okolje z ranljivostjo [20]. Bralec je vabljen, da ga prenese in preizkusi sam.

Ti rezultati poudarjajo pomen natančnega upravljanja s pomnilnikom in potrebo po večslojnih varnostnih pristopih pri razvoju programske opreme. Obenem kažejo na stalno evolucijo varnostnih izzivov in potrebo po nenehnem prilagajanju varnostnih strategij.

6 DISKUSIJA

6.1 Izraba ranljivosti v praksi

V grobem so napadi na kopico zahtevnejši od napadov na sklad in zahtevajo natančno poznavanje notranje strukture kopice. Dodelitve in sprostitve pomnilnika so v sistemih v praksi običajno manj predvidljive, kot je bilo prikazano v našem primeru, princip napada pa ostaja enak. Običajno je cilj napadalca izvesti poljubno kodo ali pridobiti dostop do lupine ali občutljivih podatkov.

Ker do delitve pravic med izvajanjem programa po navadi ne pride slučajno, kot v našem primeru, se napadalci obenem poslužujejo tudi drugih tehnik, kot so prelivanje medpomnilnika, ranljivosti v knjižnicah in napadi na sklad. Sproščeni in hkrati alocirani blok pa se lahko uporabi za prepisovanje kazalca na naslednji sproščeni blok v košu. S tem se lahko doseže, da se programu in s tem napadalcu ob enem izmed prihodnjih klicev funkcije malloc() (v kolikor uspe blok pretentati hevristične varnostne preglede knjižnice) dodeli dostop do skorajda poljubnega segmenta pomnilnika, tudi do kode, ki se izvaja, ali pa do občutljivih podatkov.

Ko napadalec ugotovi naslov standardne knjižnice v pomnilniku, lahko vrnitveni naslov prepiše z naslovom funkcije, ki jo želi izvesti, npr. system("/bin/sh") in s tem pridobi dostop do lupine. Takemu napadu pravimo "ret2libc".

6.2 Primeri ranljivosti v aplikaciji WhatsApp za Android

Leta 2019 so raziskovalci odkrili ranljivost v knjižnici android-gif-drawable [24], ki jo uporablja tudi priljubljena aplikacija WhatsApp za Android. Ranljivost je omogočala izvajanje poljubne kode na daljavo in dostop do lupine na napravi uporabnika, če je ta odprl posebej oblikovan GIF. Napadalec bi lahko izkoristil to ranljivost za krajo občutljivih podatkov, kot so fotografije in sporočila [25].

Ranljivost poteka sledeče:

1. Napadalec pošlje GIF datoteko uporabniku preko kateregakoli kanala. Ena od možnosti je, da pošlje datoteko kot dokument preko WhatsApp-a.
2. Če je napadalec v stiku z uporabnikom (npr. prijatelj), se okvarjena GIF datoteka glede na privzete nastavitve samodejno prenese brez uporabnikovega posredovanja.
3. Uporabnik želi poslati medijsko datoteko kateremu od svojih prijateljev preko WhatsApp-a. Zato pritisne na gumb za pripenjanje datotek in odpre galerijo WhatsApp-a, da izbere medijsko datoteko, ki jo želi poslati prijatelju. Pri tem ni potrebno, da uporabnik dejansko pošlje datoteko, saj že samo odpiranje galerije sproži napako.
4. Ker WhatsApp prikaže predogled vsake medijske datoteke (vključno z GIF datoteko, ki jo je prejel), sproži napako dvojnega sproščanja pomnilnika in omogoči napad.

Ranljivost je znana pod oznako CVE-2019-11932 in je bila odpravljena z izdajo posodobitve aplikacije 2.19.244 [25]. Ta primer iz prakse dodatno poudarja resnost ranljivosti dvojnega sproščanja pomnilnika in potrebo po stalnem posodabljanju programske opreme ter implementaciji robustnih varnostnih mehanizmov. Obenem kaže, da so tovrstne ranljivosti lahko prisotne tudi v zelo razširjenih aplikacijah, kar še povečuje njihov potencialni vpliv.

7 PREDLAGANE REŠITVE

Demonstracija je pokazala, da lahko navidezno nedolžna ranljivost dvojnega sproščanja pomnilnika privede do resnih posledic. Za zmanjšanje verjetnosti pojava takšnih napak predlagamo več pristopov:

- **Implementacija načela enkratnega lastništva** pri upravljanju pomnilnika. To načelo določa, da je za vsak blok pomnilnika odgovoren le en del kode. Če je blok sproščen, ga ni več dovoljeno uporabljati. To načelo je še posebej pomembno pri delu z globalnimi spremenljivkami. V primerih, ko striktno upoštevanje tega načela ni mogoče, pa je ključnega pomena, da so posamezne funkcije in deli kode jasni in pregledni ter da ima vsaka funkcija konceptualno en sam namen.
- **Uporaba pomnilniško varnih** (angl. *memory-safe*) **programskih jezikov**, ki vključujejo vgrajene zaščite pred omenjenimi napadi. Med te spadajo jeziki z avtomatskim upravljanjem pomnilnika kot so Python, Swift, C#, Java in Go. Za aplikacije, kjer je kritična učinkovitost izvajanja, pa sta primerni alternativi Rust ali uporaba pametnih kazalcev v C++.
- **Vključitev orodij za statično analizo kode** in odkrivanje napak pri upravljanju pomnilnika v razvojni proces. Primeri takih orodij so Valgrind in AddressSanitizer. Dodatno je koristno izvajanje *fuzz* testiranja, tj. avtomatiziranega testiranja z obsežnim naborom naključnih, nepredvidenih ali neveljavnih vhodov.
- **Implementacija najnovejših varnostnih smernic** in pravil za programiranje. Ključno je tudi redno posodabljanje uporabljenih knjižnic za odpravo znanih ranljivosti. V operacijskih sistemih je priporočljiva aktivacija varnostnih mehanizmov, kot sta ASLR in DEP, ki otežujeta napadalcem napovedovanje naslovov pomnilnika in izvajanje kode v podatkovnih segmentih.
- **Omejitev privilegijev programa** in izvajanje v peskovniku (angl. *sandboxing*) za zmanjšanje potencialnih posledic napadov. Tehnike kot so Seccomp, Landlock, AppArmor in SELinux omogočajo omejevanje dostopa programa do sistemskih virov (npr. branja in pisanja datotek izven predvidenih direktorijev ali dostopa do lupine). Virtualizacija in uporaba vsebnikov dodatno prispevata k izolaciji programa in preprečevanju dostopa do občutljivih podatkov drugih aplikacij.
- **Uveljavitev načel defenzivnega programiranja.** Ta pristop zahteva sistematično predvidevanje potencialnih napak in napadov ter implementacijo ustreznih zaščitnih mehanizmov. To vključuje temeljito preverjanje vhodnih podatkov, validacijo rezultatov funkcijskih klicev in verifikacijo veljavnosti kazalcev pred njihovo uporabo.

8 ZAKLJUČEK

V tem članku smo podrobno analizirali ranljivost zaradi dvojnega sproščanja pomnilnika, ki se skupaj s sorodnimi ranljivostmi, kot sta uporaba pomnilniških blokov po sprostitvi in puščanje pomnilnika, uvršča med najpogostejša varnostna tveganja pri upravljanju s kopico. Te ranljivosti se pojavljajo predvsem zaradi napak v programih, napisanih v programskih jezikih z ročnim upravljanjem pomnilnika.

Na praktičnem primeru smo prikazali, kako lahko napadalci izkoristijo omenjeno ranljivost za pridobitev administratorskih pravic v preprostem programu, ki simulira upravitelja podatkovne baze. Prav tako smo opisali, kako se lahko ta ranljivost v praksi izkoristi za izvajanje poljubne kode, kar smo ponazorili s primerom ranljivosti v priljubljeni mobilni aplikaciji.

Kljub naraščajoči priljubljenosti pomnilniško varnih programskih jezikov ostaja uporaba jezikov z ročnim upravljanjem pomnilnika, kot je C, pogosta zaradi njihove hitrosti in praktičnosti v specifičnih razvojnih okoljih. To pomeni, da ostaja nevarnost izkoriščanja opisanih ranljivosti še vedno relevantna.

V priporočilih za zmanjšanje tveganj smo podali več strategij, kako lahko razvijalci in skrbniki sistemov omejijo tovrstne varnostne ranljivosti ter zaščitijo svoje aplikacije pred napadi. To vključuje uporabo sodobnih orodij za statično in dinamično analizo kode, uvedbo strožjih pravil in smernic za upravljanje pomnilnika ter omejitev privilegijev programa. Prav tako smo poudarili pomen defenzivnega programiranja, ki pomaga preprečevati napake že v najzgodnejših fazah razvoja programske opreme.

S tem smo izpostavili kompleksnost varnosti pomnilnika in nujnost celovitega pristopa k varovanju programske opreme pred napadi. Razumevanje teh ranljivosti in uvedba ustreznih zaščitnih ukrepov sta ključna koraka za zagotavljanje varnosti in zanesljivosti aplikacij v vedno bolj povezanem digitalnem okolju.

LITERATURA

[1] Juan Caballero in sod. “Undangle: early detection of dangling pointers in use-after-free and double-free vulnerabilities”. V: *Proceedings of the 2012 International Symposium on Software Testing and Analysis.* ISSTA 2012. Minneapolis, MN, USA: Association for Computing Machinery, 2012, str. 133–143. ISBN: 9781450314541. DOI: 10.1145/2338965.2336769.

[2] Yves Younan. “FreeSentry: Protecting Against Use-After--Free Vulnerabilities Due to Dangling Pointers”. V: jan. 2015. DOI: 10.14722/ndss.2015.23190.

[3] Erik van der Kouwe, Vinod Nigade in Cristiano Giuffrida. “DangSan: Scalable Use-after-free Detection”. V: *Proceedings of the Twelfth European Conference on Computer Systems.* EuroSys ’17. Belgrade, Serbia: Association for Computing Machinery, 2017, str. 405–419. ISBN: 9781450349383. DOI: 10.1145/3064176.3064211.

[4] Zekun Shen in Brendan Dolan-Gavitt. “HeapExpo: Pinpointing Promoted Pointers to Prevent Use-After-Free Vulnerabilities”. V: *Proceedings of the 36th Annual Computer Security Applications Conference.* ACSAC ’20. Austin, USA: Association for Computing Machinery, 2020, str. 454–465. ISBN: 9781450388580. DOI: 10.1145/3427228.3427645.

[5] Daiping Liu, Mingwei Zhang in Haining Wang. “A Robust and Efficient Defense against Use-after-Free Exploits via Concurrent Pointer Sweeping”. V: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’18. Toronto, Canada: Association for Computing Machinery, 2018, str. 1635–1648. ISBN: 9781450356930. DOI: 10.1145/3243734.3243826.

[6] Jinchang Hu in sod. “A memory-related vulnerability detection approach based on vulnerability features”. V: *Tsinghua Science and Technology 25.5 (2020)*, str. 604–613. DOI: 10.26599/TST.2019.9010068.

[7] Dusan Repel, Johannes Kinder in Lorenzo Cavallaro. “Modular Synthesis of Heap Exploits”. V: *Proceedings of the 2017 Workshop on Programming Languages and Analysis for Security.* PLAS ’17. Dallas, Texas, USA: Association for Computing Machinery, 2017, str. 25–35. ISBN: 9781450350990. DOI: 10.1145/3139337.3139346.

[8] Haijun Wang in sod. “Typestate-guided fuzzer for discovering use-after-free vulnerabilities”. V: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. ICSE ’20. Seoul, South Korea: Association for Computing Machinery, 2020, str. 999–1010. ISBN: 9781450371216. DOI: 10.1145/3377811.3380386.

[9] Hua Yan in sod. “Machine-Learning-Guided Typestate Analysis for Static Use-After-Free Detection”. V: *Proceedings of the 33rd Annual Computer Security Applications Conference.* ACSAC ’17. Orlando, FL, USA: Association for Computing Machinery, 2017, str. 42–54. ISBN: 9781450353458. DOI: 10.1145/ 3134600.3134620.

[10] Binfa Gui, Wei Song in Jeff Huang. “UAFSan: an object-identifier-based dynamic approach for detecting use-after-free vulnerabilities”. V: *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis.* ISSTA 2021. Virtual, Denmark: Association for Computing Machinery, 2021, str. 309–321. ISBN: 9781450384599. DOI: 10.1145/3460319.3464835.

[11] Márton Erdő s, Sam Ainsworth in Timothy M. Jones. “MineSweeper: a “clean sweep” for drop-in use-after-free prevention”. V: *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems.* ASPLOS ’22. Lausanne, Switzerland: Association for Computing Machinery, 2022, str. 212–225. ISBN: 9781450392051. DOI: 10.1145/ 3503222.3507712.

[12] Brian Wickman in sod. “Preventing Use-After-Free Attacks with Fast Forward Allocation”. V: *30th USENIX Security Symposium* (USENIX Security 21). USENIX Association, avg. 2021, str. 2453– 2470. ISBN: 978-1-939133-24-3. URL: https://www.usenix.org/conference/usenixsecurity21/ presentation/wickman.

[13] Binfa Gui in sod. “Automated Use-After-Free Detection and Exploit Mitigation: How Far Have We Gone?” V: *IEEE Transactions on Software Engineering 48.11* (2022), str. 4569–4589. DOI: 10.1109/TSE.2021.3121994.

[14] Qi Liu, Kaibin Bao in Veit Hagenmeyer. “Binary Exploitation in Industrial Control Systems: Past, Present and Future”. V: *IEEE Access* 10 (2022), str. 48242–48273. DOI: 10.1109/ACCESS.2022.3171922.
[15] TIS Committee. *Tool Interface Standard (TIS) Executable and Linking Format (ELF) Specification.* Ver. 1.2. 1995. URL: https://refspecs.linuxfoundation .org /elf/elf.pdf (pridobljeno 12. 5. 2024).
[16] Yanpas – Wikimedia Commons. *C memory layout of program. bss, stack, heap.* 2015. URL: https: //en.wikipedia.org/wiki/File:C-memlayout.svg (pridobljeno 12. 5. 2024).
[17] OpenDSA. *Heap Memory*. URL: https://opendsa-server.cs.vt.edu/ODSA/Books/CS2/html/ HeapMem.html (pridobljeno 12. 5. 2024).
[18] Malloc Internals. URL: https : / / sourceware . org / glibc / wiki / MallocInternals (pridobljeno 12. 5. 2024).
[19] CWE-415: Double Free. URL: https://cwe.mitre.org/data/definitions/415.html (pridobljeno 12. 5. 2024).
[20] GitHub repozitorij. URL: https://github.com/marindereggi/double-free.
[21] Ubuntu 16.04 LTS transitions to Extended Security Maintenance (ESM). URL: https://canonical. com/blog/ubuntu-16--04-lts-transitions-to-extended-security-maintenance-esm (pridobljeno 12. 5. 2024).
[22] The GNU C Library Repository. Ta potrditev uvede podporo za lokalni predpomnilnik blokov za niti v izvorni kodi. URL: https : / / sourceware . org / git / ?p = glibc . git ; a = commitdiff ; h = d5c3fafc4307c9b7a4c7d5cb381fcdbfad340bcc (pridobljeno 12. 5. 2024).
[23] tukan. thread local caching in glibc malloc. URL: http://tukan.farm/2017/07/08/tcache/ (pridobljeno 12. 5. 2024).
[24] Android GIF Drawable Source Code. URL: https://github.com/koral--/android-gif-drawable/ tree/dev/android-gif-drawable/src/main/c (pridobljeno 12. 5. 2024).
[25] Awakened. How a double-free bug in WhatsApp turns to RCE. URL: https://awakened1712.github. io/hacking/hacking-whatsapp-gif-rce/ (pridobljeno 12. 5. 2024).

▪

Marin Gazvoda de Reggi je študent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Zanimajo ga področja razvoja programske opreme, kibernetske varnosti in umetne inteligence. Njegovi raziskovalni interesi zajemajo teorijo programskih jezikov in njihovo varnost.

▪

Matevž Pesek je docent in raziskovalec na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je diplomiral (2012) in doktoriral (2018). Od leta 2009 je član Laboratorija za računalniško grafiko in multimedije. Od leta 2024 izvaja predmet Varnost programov.

Uporaba lokalnih podatkov za boljše spremljanje turističnih tokov: kritična perspektiva

Urška Starc Peceny, Tomi Ilijaš
Arctur d.o.o., Industrijska cesta 1a, 5000 Nova Gorica
urska@arctur.si, tomi.ilijas@arctur.si

Izvleček

Na področju upravljanja turizma je učinkovito spremljanje turističnih tokov ključnega pomena za spodbujanje trajnostnega razvoja in za preudarno razporejanje virov. Medtem, ko se običajne metode pogosto zanašajo na združene ali posplošene podatke, ta članek zagovarja ključno vlogo lokalnih podatkov, pri čemer poudarja njihovo zmožnost, da v realnem času ali z višjo frekvenco zajema nudijo različne vpoglede in uporabne informacije. To je pomembno tudi ob razvoju podatkovnih prostorov v Evropi, v katerih je ključnega pomena izkoriščanje lokalnih virov podatkov, zlasti tistih, ki se zbirajo, vendar se ne delijo za nadaljnjo ponovno uporabo. Predstavljeni so rezultati projekta Planinstvo 4.0 in nujnost zakonodajnih reform, ki bi olajšale ponovno uporabo določenih podatkov, saj nekatere obstoječe pravne vrzeli v ekosistemu trenutno ogrožajo javno varnost, namesto da bi jo varovale.

Ključne besede: Turistični tokovi, lokalni podatki, trajnostni turizem, Turizem 4.0

Leveraging Local Data for Enhanced Monitoring of Tourist Flows: A Critical Perspective

Abstract

In the field of tourism management, the effective monitoring of tourist flows is crucial for the promotion of sustainable development and for the prudent allocation of resources. While conventional methods often rely on aggregated or generalised data, this paper argues for the key role of local data, highlighting its ability to provide diverse insights and actionable information in real-time or at higher frequency captures. This is also important in the development of data spaces in Europe where it is crucial to exploit local data sources, especially those that are collected but not shared for further reuse. The results of the Mountaineering 4.0 project and the necessity of legislative reforms that would facilitate the re-use of certain data are presented, as several existing legal gaps in the ecosystem currently threaten public safety instead of protecting it.

Keywords: tourist flows, local data, sustainable tourism, Tourism 4.0

1 UVOD

Upravljanje je v turizmu eno od kritičnih področij, kjer je učinkovito spremljanje turističnih tokov ključno za spodbujanje trajnostnega razvoja in preudarno razporejanje virov. Navadno se metode spremljanja opirajo na agregirane ali posplošene podatke, vendar pa ta članek zagovarja ključno vlogo lokalnih podatkov pri ponujanju poglobljenih vpogledov za ukrepanje v realnem času ali z visoko frekvenco, kar je ključno pri podatkovno vodenem odločanju.

Poleg tega, v času hitrega razvoja podatkovnih prostorov v Evropi, postaja koriščenje lokalnih virov podatkov nujno, še zlasti tistih, ki so zbrani, vendar niso takoj na voljo za nadaljnjo uporabo. Ta potreba je predstavljena skozi prizmo projekta Planinstvo 4.0, ki je pionirski v alpskem okolju. Sodelujoči deležniki, Planinska zveza Slovenije, CIPRA Slovenija in Planinsko društvo Tržič, so skupaj s tehnološkim partnerjem na petih priljubljenih planinskih destinacijah Slovenije namestili pametne senzorje za štetje pohodnikov. S po-

vezovanjem različnih podatkovnih virov je tako znotraj platforme FLOWS podjetja Arctur v okviru projekta bilo omogočeno celovito razumevanje in napovedovanje vzorcev obiskovalcev na izbranih lokacijah.

Kljub učinkovitemu sistemu zbiranja podatkov o turističnih prihodih in nočitvah za potrebe turistične takse v Sloveniji turistične destinacije nimajo takojšnjega dostopa do njih, kar ovira njihovo sposobnost hitrega odzivanja.

V prizadevanju za dvig ozaveščenosti in spodbujanje dialoga med odločevalci ta članek predstavlja temeljni koncept pobude Turizem 4.0, ki se osredotoča na kakovost življenja lokalne skupnosti. Pojasnjuje, kako inovativna orodja omogočajo zbiranje podatkov iz raznolikih virov (npr. podatkov v realnem času, zgodovinskih in statističnih podatkov o turističnih tokovih ipd.), kar omogoča učinkovito načrtovanje in podatkovno vodeno odločanje. Poleg tega identificira obstoječe pravne vrzeli v podatkovnem ekosistemu in osvetljuje potrebo po reformah za izboljšanje dostopnosti in izkoriščanja podatkov

2 RAZUMEVANJE PREKOMERNEGA TURIZMA

Upravljanje turističnih destinacij, tako v Sloveniji kot drugod, pogosto sledi paradigmi rasti, ki daje prednost povečanju števila obiskovalcev. Vendar pa ta neusmiljeni lov za rastjo predstavlja vse večji pritisk na destinacije, kar vodi v pojav, imenovan prekomerni turizem. Prekomerni turizem se kaže na različne načine, vključno s prenatrpanostjo turistov, napetimi odnosi z lokalnimi prebivalci, poslabšanjem izkušenj obiskovalcev, obremenjenostjo infrastrukture, okoljsko degradacijo in ogroženostjo kulturne dediščine. Ta izziv je še dodatno okrepljen zaradi močne odvisnosti turizma od naravnih in kulturnih virov, kar povzroča skrbi glede trajnosti same industrije [5].
Težava prekomernega turizma se je v zadnjih letih v Sloveniji pokazala na primer na Bledu, kjer je anketa med prebivalci pokazala visoko raven nezadovoljstva [1]. Na mednarodni ravni se težave kažejo skozi proteste in demonstracije na priljubljenih turističnih destinacijah, kot so Kanarski otoki v Španiji, kjer so protestniki zahtevali zamrznitev turizma, sklicujoč se na dejstvo, da je trenutni model delovanja destinacije nepopravljivo podražil življenje in je okoljsko nevzdržen za prebivalce [11]. Obravnava prekomernega turizma predstavlja kompleksen izziv, ki izvira iz njegove od konteksta odvisne narave. V nasprotju z drugimi vprašanji, povezanimi s turizmom, prekomerni turizem ni enostavno merljiv in se razlikuje od destinacije do destinacije. Njegova definicija zajema številne dejavnike, kot so število turistov, njihovo vedenje in zmogljivost destinacije, da jih učinkovito sprejme. Čeprav je prekomerni turizem vedno bolj prepoznan, se strategije za zmanjšanje njegovih negativnih vplivov pogosto osredotočajo na simptome, namesto da bi se ukvarjale s temeljnimi vzroki - nekontrolirano rastjo obsega turizma. Preprečevanje prekomernega turizma je pogosto lažje izvedljivo in bolj učinkovito kot poskus odpravljanja njegovih posledic, zlasti na podeželju in v primestnih območjih, kjer je ravnovesje med turističnim razvojem in lokalno trajnostjo izjemno občutljivo.

Razumevanje lokalnih raznolikosti zahteva dostop do lokalnih podatkov, ki so hkrati nepogrešljivi za izkoriščanje priložnosti, ki jih prinaša digitalizacija. Digitalna preobrazba, ki se širi skozi gospodarstvo in družbo, ponuja ogromen potencial za napredek turističnih storitev. Digitalni razvoj širi obseg generiranja, zbiranja in uporabe podatkov, s čimer omogoča bolj trajnostne in inovativne turistične izkušnje. S povečanjem deljenja podatkov med javnim in zasebnim sektorjem se lahko turistične storitve razvijajo v smeri zagotovljanja personalizirane izkušnje, obogatene s tehnologijo, kar spodbuja trajne učinke.

Premik glede praks deljenja podatkov lahko spodbudi razvoj inovativnih turističnih storitev, ki spodbujajo trajnost in izboljšujejo mobilnost ter omogočajo upravljanje s turističnimi tokovi v realnem času, kar pripomore k reševanju mnogih izzivov, tudi izziva prekomernega turizma. Z izkoriščanjem vpogledov v podatke lahko podjetja napovedujejo povpraševanje, analizirajo profile strank in izboljšujejo njihove izkušnje. Javno-zasebna partnerstva so ključna za oblikovanje novih dogovorov in platform za deljenje podatkov, s čimer se izboljšujejo procesi odločanja. Povečano deljenje podatkov med zasebnimi subjekti in javnimi organi lahko znatno okrepi konkurenčnost malih in srednje velikih turističnih podjetij ter odpornost destinacij. Vendar je nujno, da takšne pobude upoštevajo zakonodajo o zasebnosti in spoštujejo komercialne interese vseh zainteresiranih strani.

Poleg navedenega obstaja možnost ustvarjanja platform, kjer si lastništvo podatkov deli skupnost, namesto da bi bili le-ti skoncentrirani v rokah nekaj subjektov. Pobude, kot so platformne zadruge, kažejo na možnost podatkovnih pobud v lasti skupnosti, ki spodbujajo vključenost in inovativnost v turistič-

nem sektorju. Če povzamemo, z izkoriščanjem priložnosti, ki jih ponujata digitalizacija in izboljšana izmenjava podatkov, lahko zainteresirani deležniki spodbudijo transformativen napredek v turističnih storitvah ter spodbujajo bolj trajnosten, odporen in vključujoč turistični ekosistem [7].

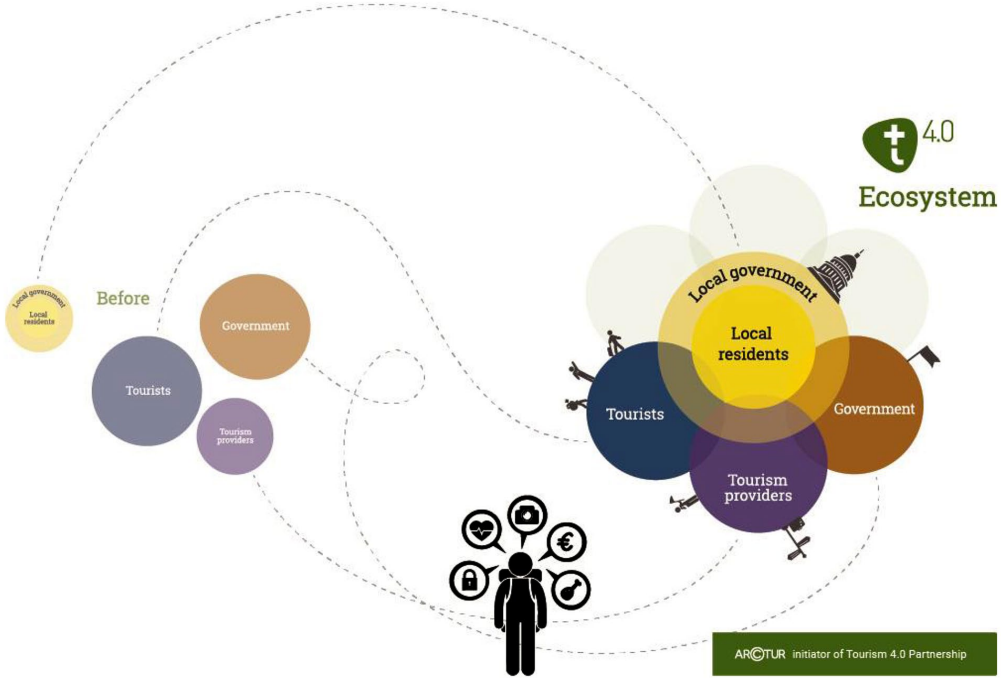
2.1 Pobuda Turizem 4.0 in izkoriščanje moči lokalnih podatkov

Ena od uspešnih pobud, ki si prizadeva graditi opisani ekosistem, je pobuda Turizem 4.0. Koncept izhaja iz sodobne industrijske paradigme, znane kot Industrija 4.0, s ciljem povečanja vrednosti ponudbe turizma preko inovacij, znanja, tehnologije in ustvarjalnosti. Turizem 4.0 si prizadeva spodbuditi model sodelovanja, ki blaži negativne vplive turizma in hkrati izboljšuje celotno izkušnjo obiskovalcev. Doseganje tega cilja vključuje izkoriščanje smernic in orodij, ki jih ponuja Pametni turizem, pojav, za katerega je značilna integracija informacijske in komunikacijske tehnologije (IKT) v izkušnjo turizma [4]. Ta integracija se uresničuje preko platforme, ki vključuje vse deležnike, aktivne v turističnem sektorju, vključujoč lokalno skupnost, vladne organe, ponudnike turističnih storitev in same turiste, kot je prikazano v grafiki 1.

Pobudo Turizem 4.0 je inciralo podjetje Arctur, slovensko visoko tehnološko podjetje, potem ko je prepoznalo pomanjkljivost pripravljenosti malih in srednjih podjetij v turističnem sektorju za sprejemanje ključnih omogočitvenih tehnologij. Posledično je bilo vzpostavljeno partnerstvo Turizem 4.0, ki spodbuja sodelovanje vseh deležnikov pri raziskavah in razvoju v turističnem sektorju po celotnem svetu. Partnerstvo sestavljajo različni deležniki turističnega ekosistema, ki bi radi digitalizacijo izkoristili v svoj prid. Do danes se je partnerstvu pridružilo že več kot 230 članov iz celega sveta.

Pomena lokalnih podatkov pri upravljanju turizma ni mogoče prezreti, zlasti ker lokalni podatki ponujajo bolj poglobljeno razumevanje vedenja in preferenc obiskovalcev, ter omogočajo prilagojen vpogled v posamezno destinacijo. Za razliko od zanašanja zgolj na podatke velikih platform, ki lahko ponudijo splošne vpoglede, lokalni podatki zagotavljajo podroben pregled turističnih tokov, vzorcev potrošnje in interakcij znotraj skupnosti. Z izkoriščanjem takšnih informacij lahko zainteresirani deležniki oblikujejo bolj ciljno usmerjene strategije za izboljšanje izkušnje obiskovalcev, hkrati pa ublažijo negativne vplive turizma na lokalno okolje in kulturo. Upoštevanje lokalnih podatkov zagotavlja, da odločitve temeljijo na edinstvenih značilnostih in potrebah vsake destinacije, kar spodbuja bolj trajnostno in pristno turistično izkušnjo.

Poleg tega zmanjševanje odvisnosti od podatkov velikih platform omogoča lokalnim skupnostim, da pre-



Grafika 1: **Turizem 4.0 ekosistem (arhiv Arctur)**

vzamejo nadzor nad svojimi strategijami za upravljanje s turizmom. Z zbiranjem in analizo lastnih podatkov destinacije lahko presežejo omejitve zunanjih platform, in prilagodijo svoje pobude zadovoljevanju interesa tako obiskovalcev kot svojih prebivalcev. Ta lokalni pristop poleg spodbujanja močnejšega vključevanja lokalnih skupnosti omogoča tudi hitrejši in bolj agilen odziv na spreminjajočo se dinamiko turizma. Nazadnje, dajanje prednosti uporabi lokalnih podatkov destinacijam omogoča, da bolj avtentično oblikujejo svoje turistične pripovedi ter spodbujajo trajnostni razvoj in odpornost v luči razvijajočih se trendov v industriji.

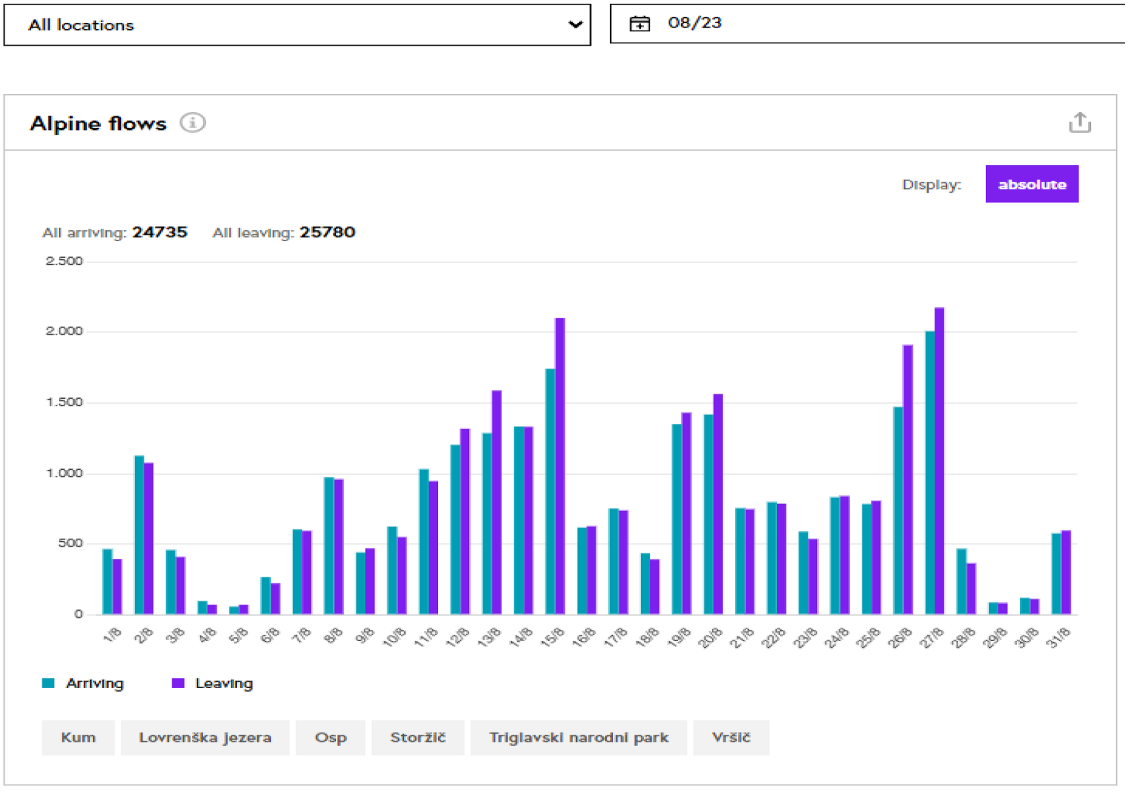
2.2 Planinstvo 4.0 in in analiza turističnih tokov na mikro lokacijah

Projekt Planinstvo 4.0 predstavlja enega od pionirskih podvigov v uvajanju digitalizacije v turističnem sektorju. Vključeni deležniki, Planinska zveza Slovenije, CIPRA Slovenija in Planinsko društvo Tržič, so se skupaj s tehnološkim partnerjem lotili te inovativne pobude v alpskem okolju kot odziv na velik porast pohodnikov v času pandemije Covida-19. V središču projekta Planinstvo 4.0 je bila integracija pametnih senzorjev za pohodnike, nameščenih na petih priljubljenih pohodniških destinacijah v Sloveniji. Podatki iz teh senzorjev, združeni s številnimi drugimi viri podatkov, kot so mobilni podatki, prometne informacije, vremenske razmere in nesreče v gorah, so bili povezani v aplikaciji FLOWS, ki jo je razvilo podjetje Arctur. Ta celovita integracija je omogočala spremljanje in napovedovanje vzorcev obiskovalcev v realnem času na vsaki lokaciji, kar je prispevalo k bolj informiranim odločitvam in trajnostnemu upravljanju.

Med pandemijo Covida-19 je narasla priljubljenost pohodništva. Želja po svežem zraku in neomejenem gibanju je v naravo v hribe pripeljala številne obiskovalce. Če je morda nosilnost urbanih središč lahko bolj raztegjiva, se povečan obisk v manjših destinacijah zelo hitro prevesi v negativno izkušnjo za vse, obiskovalce in lokalno skupnost.

V naslednjih grafikah so predstavljeni primeri analize, ki je konzorciju projekta omogočala vpogled v realne podatke.

Grafika 2 prikazuje obisk v avgustu 2023 na vseh lokacijah (Kum, Lovrenška jezera, Osp, Stožič, Triglavski narodni park in Vršič), kjer so bili šteti obiskovalci. Odstopanja med vsemi prihodi in odhodi



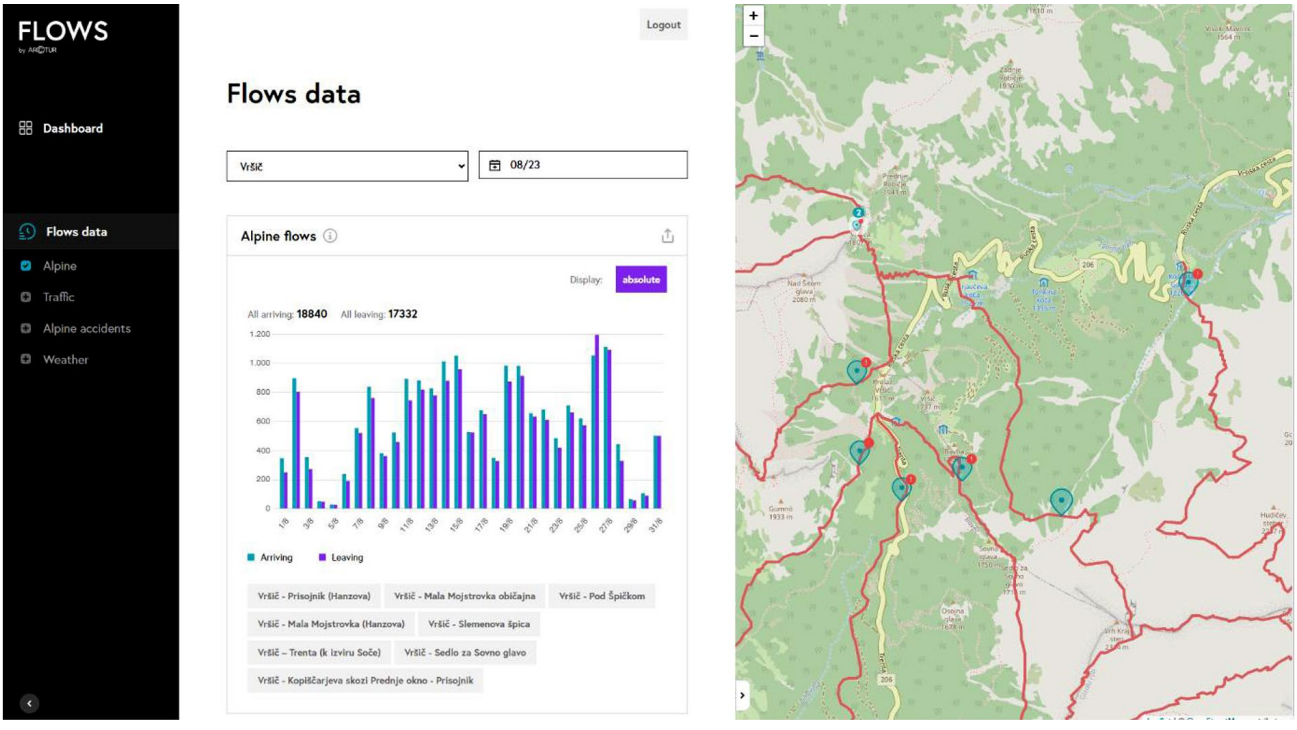
Grafika 2: **Obisk v avgustu 2023 na vseh lokacijah (arhiv Arctur)**

nastajajo, ker gredo pohodniki po različnih poteh, tudi tistih, ki niso opremljene s senzorji. Zelo netipično za ta čas je opaziti znatno zmanjšanje obiska na začetku avgusta 2023, ko so bile poplave. Pričakovano pa izstopata dva vrhunca: 15. avgust in zadnji počitniški vikend v avgustu. Vsaka izredna situacija je priložnost za učenje. Tako kot je Covid-19 pomagal razumeti, kaj pomeni, ko je turizem in gibanje ljudi popolnoma ustavljeno, se lahko iz podatkov med in po poplavah naučimo tudi, kako kratek je čas, ko so se kljub izrednim razmeram takrat v državi obiskovalci spet pričeli gibati. Realni podatki, ki lahko pomagajo pripraviti odlične načrte upravljanja turističnih destinacij v kriznih situacijah. Nedavno so take načrte v okviru projekta Evropske komisije pripravljale tudi 3 slovenske destinacije [2].

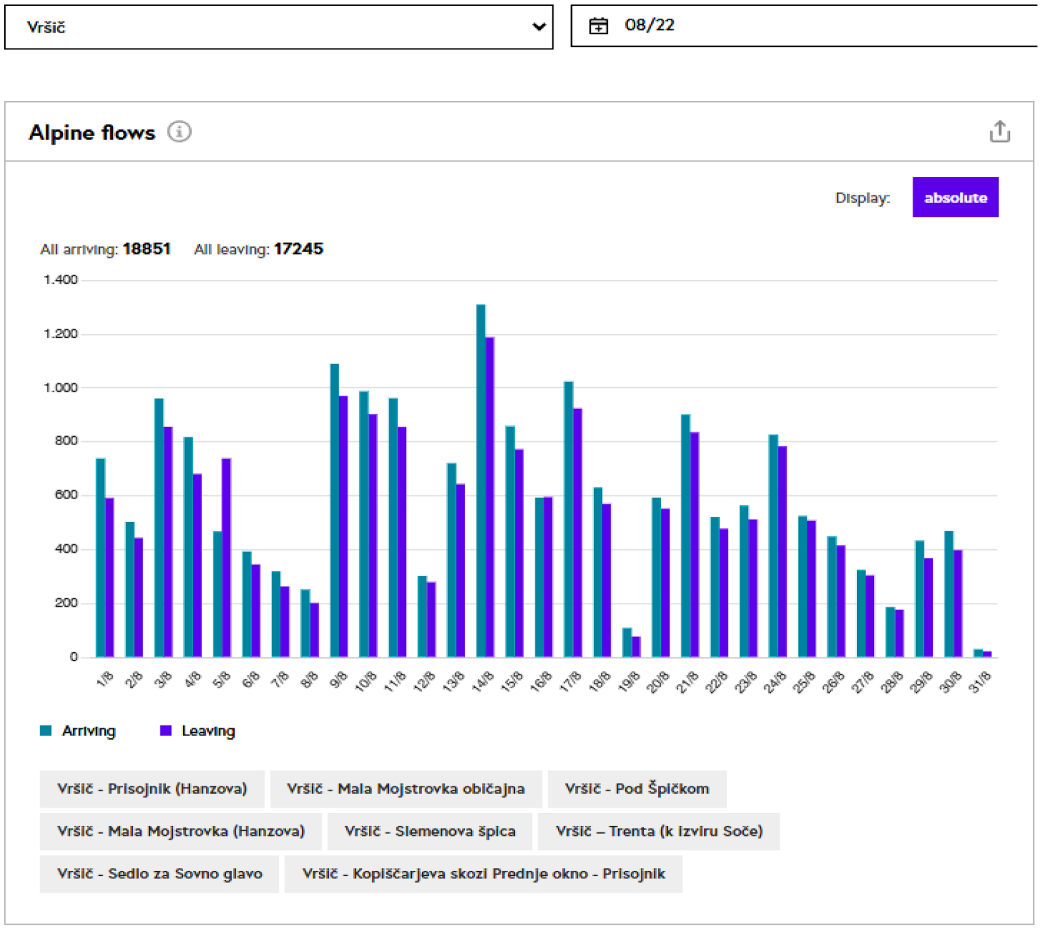
Grafika 3 prikazuje obisk v avgustu 2023 na Vršiču, ki je tudi daleč najbolj obiskana lokacija od vseh lokacij, kjer so se izvajale meritve. Na zemljevidu so prikazane planinske poti, ki jih upravlja PZS [8] in lokacije posameznega števca.

Za primerjavo grafik 3 in 4 vidimo vzorec, in sicer najvišji obisk okoli 15. avgusta 2023 (prazniki, podaljšan vikend). Zanimivo je, da je skupen obisk v mesecu avgustu v obeh letih skoraj enak:

- 2023: iz 18840 / proti 17332
- 2022: iz 18851 / proti 17245.



Grafika 3: **Obisk na Vršiču v avgustu 2023 (arhiv Arctur)**



Grafika 4: **Obisk na Vršiču v avgustu 2022 (arhiv Arctur)**

Grafika 5 prikazuje povezave med številom obiskovalcev planinskih poti in številom vozil, ki so peljale mimo državnih števcev prometa pri Bovcu in Kranjski gori v avgustu 2023. Podobne povezave lahko ugotovimo tudi s primerjanjem vremena (visok odstotek vlage pomeni slabo vreme).

Grafika 5: **Združevanje različnih virov podatkov (arhiv Arctur)**

Zanimiv je tudi podatek za oktober 2022 v grafiki 6, ko je bilo 16. 10. preštetih več kot 1500 obiskovalcev, ki so se odpravili na Slemenovo špico v času rumenih macesnov.

2.3 Pomembnost dostopnosti podatkov

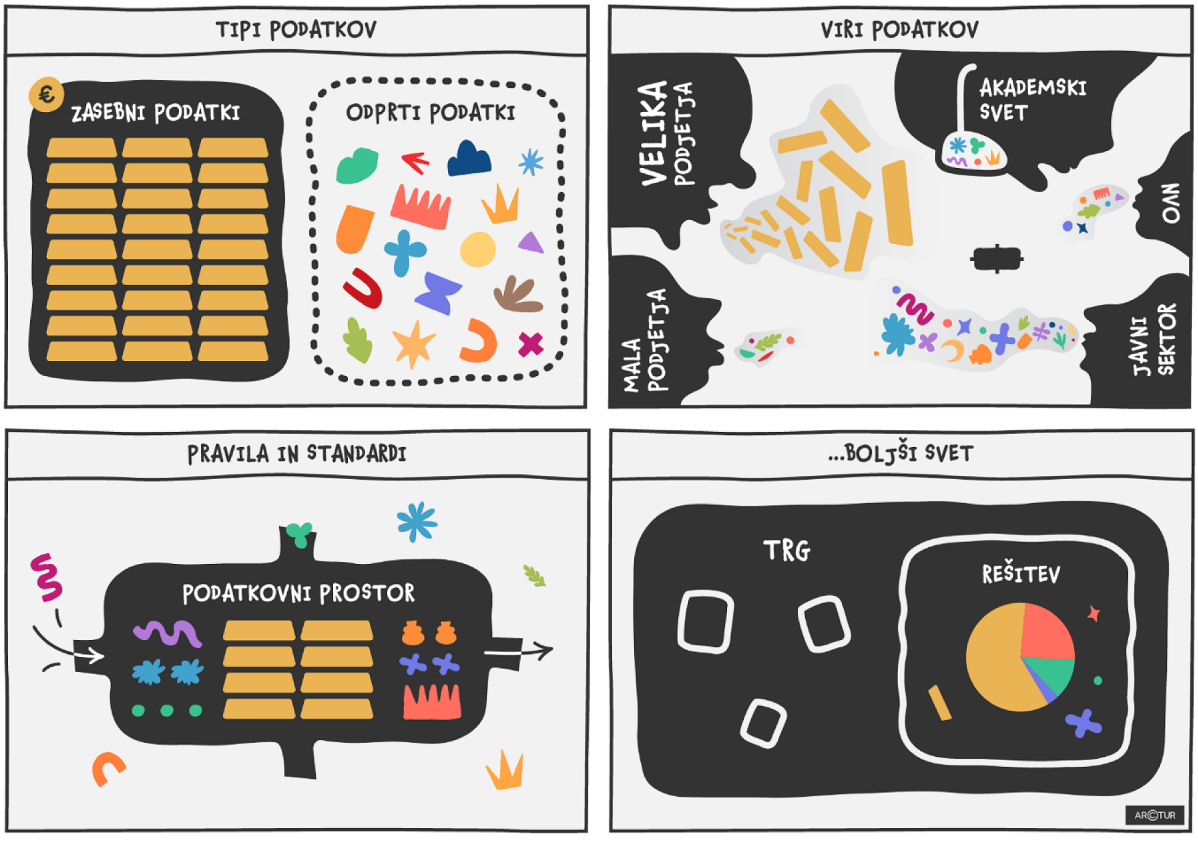
Predstavljeni primeri nam nazorno kažejo, kako nam kombiniranje različnih podatkov pomaga pri razumevanju celotne slike stanja. Zato je dostopnost podatkov, predvsem tistih, ki zelo pomagajo izboljšati kakovost slike stanja, poglavitnega pomena. Že vrsto let inciatorji Partnerstva za Turizem 4.0 opozarjajo na pomembnost dostopnosti neagregiranih vendar anonimiziranih podatkov o turističnih nočitvah v Sloveniji. Žal je tukaj zakonodaja velikokrat v napoto. V Sloveniji morajo vsi ponudniki turističnih nastanitev v roku 12 ur prijaviti gosta v spletno aplikacijo eTurist, ki jo je za namen vodenja evidence gostov, za namen obračuna in plačila turistične takse ter za statistične namene vzpostavila Agencija Republike Slovenije za javnopravne evidence in storitve (AJPES). Ažurni in neagregirani podatki o turističnih nočitvah torej obstajajo, žal pa razen za Statistični urad in policijo niso dostopni. Občine sicer tudi lahko dobijo

Grafika 6: **Obisk ob posebnih dogodkih (arhiv Arctur)**

anonimizirane podatke, vendar z mesečno zamudo, pogosto samo agregirane.

Na evropski ravni nastajajo t.i. skupni Evropski podatkovni prostori, ki bodo pripomogli k sprostitvi ogromnega potenciala inovacij, ki temeljijo na uporabi podatkov. Omogočili bodo dostopnost in izmenjavo podatkov iz cele evropske unije (EU) na zanesljiv in varen način. Podjetja, javne uprave in posamezniki iz EU bodo imeli nadzor nad podatki, ki jih ustvarijo. Ob tem je zelo pomembno širiti zavedanje in razumevanje vloge odprtih podatkov, ki so javne evidence ali informatizirane zbirke podatkov, ki jih na podlagi izvajanja javnih nalog, zbirajo zavezanci za dostop in ponovno uporabo (podatkov) v javnem sektorju. Zbirke podatkov se spletno objavijo v obliki elektronskih datotečnih formatov, v skladu z odprtimi standardi ter jih s tem dajo na voljo za kakršnokoli ponovno uporabo brez tehničnih ali licenčnih omejitev [6]. Grafika 7 v štirih delih prikazuje najprej tipe podatkov z razliko med zasebnimi oz. zaprtimi in odprtimi podatki. Prvi v lasti večinoma večjih korporacij, ki so zaprti v privatnih bazah in samo proti plačilu pridejo v obtok v obdelani obliki. Varovani in nedosegljivi za večino deležnikov so kot nekakšne zlate palice. Medtem, ko so odprti podatki pravo nasprotje. Različni, redko urejeni, marsikje še ujeti v strojno neberljive oblike, čeprav bi po zakonu morali biti dostopni. Podatkovni viri so različni deležniki, ki jih producirajo eni več – velike korporacije in državne inštitucije, drugi manj – mala podjetja in nevladnje organizacije. Akademski svet je pomemben opazovalec, ki ima večinoma dostop do neplačljivih virov. Pomembno je, da ustvarimo pravno urejena mesta in standarde - podatkovne prostore - kjer se lahko vsi srečajo in nastajajo rešitve, ki nam pomagajo sprejemati prave odločitve in izboljšujejo kvaliteto našega življenja.

Ministrstvo za digitalno preobrazbo sledi temu razvoju in je sklicalo srečanje različnih deležnikov kot prvi korak v Sloveniji do vzpostavitve podatkovnih prostorov in izmenjave podatkov na zaupanja vreden način [10]. V sklopu te inciative je podjetje Arctur, kot edini predstavnik Slovenije v evropskih projektih, ki postavljajo pravila za podatkovne prostore v turizmu, kulturni dediščini in na področju



Grafika 7: **Pomembnost odprtih podatkov ob nastajanju podatkovnih prostorov (arhiv Arctur)**

medijev, ter projekta D3HUB, ki postavlja evropski podatkovi kompetenčni center za turistične destinacije[1] za primer smiselne nadgradnje projekta Planinstvo 4.0 pripravil popis virov, ki so že uporabljeni:

- Števci obiska na planinskih poteh (5 lokacij):
  - Senzorji (IoT), v realnem času
  - TNP: uvoz iz Excela, na zahtevo
  - Odprti podatki, dostopni preko OPSI API-ja
- Promet:
  - Lokacije števcev, podatki o prometu
  - NAP Portal, v realnem času, API
- Vreme:
  - Lokacije merilnih postaj (opazovalne, samodejne)
  - Podatki o vremenu, v skoraj realnem času (na uro)
  - ARSO API
- Gorske nesreče:
  - Lokacija, čas nesreče
  - Uvoz podatkov iz Excela, na zahtevo, možno tudi preko RSS (pomanjkljivi podatki).
  - URSZ, SPIN3.sos112
- Planinske poti – API PZS:
  - Planinske poti

Ter želenih virov, ki bi pomagali skristalizirati realno sliko:

- Podatki o nočitvah v planinskih kočah
  - AJPES – obstajajo, so strukturirani, opremljeni z metapodatki, vendar niso dostopni (zakonska podlaga),
  - Planinske koče in drugi nastanitveni objekti na lokaciji
- Podatki o turističnem obisku – SURS
  - Obstajajo, vendar so premalo natančni, agregirani, so dostopni z zamikom.
- Podatki o obisku vrhov – vpisne knjige
  - Niso v digitalni obliki
- Podatki o uporabnikih mobilnih naprav
  - A1/Invenium
  - Dragi, obdelani (v njihovem dashboardu)
- Podatki o obisku turističnih znamenitosti
  - Niso dostopni, ročni vnosi/uvozi, lastniški podatki
- Podatki o obisku TIC
- Niso dostopni, ročni vnosi/uvozi, lastniški podatki

V marcu 2024 je Statistični urad RS (SURS) zagnal novo interaktivno orodje Dnevni turistični utrip Slovenije. V interaktivnem prikazu so s tridnevnim zamikom na voljo dnevno osveženi podatki o prihodih in prenočitvah turistov v Sloveniji. Prikazani so po nekaterih občinah, statističnih regijah, nastanitvenih obratih in državi prebivališča turista [9].
Vsekakor so to koraki na poti odpiranja pomembnega vira, vendar agregirani podatki na ravni statistične regije posamezni destinaciji, ki je samo del določene regije ne morejo pomagati prikazati jasne slike za svoje potrebe.

Zakaj taka potreba po podatkih v realnem času ali s čim krajšim zamikom? Menimo, da bi vsem nam moralo biti v interesu, da se potrebno modificira zakonodaja, da bodo v primeru naslednje izredne situacije turistične destinacije in reševalci v realnem času vedeli, koliko prebivalcev in turistov je potrebno evakuirati. Ta podatek v Sloveniji trenutno za tako potrebo ni na voljo kljub temu, da obstaja.

3 ZAKLJUČEK

Članek poudarja pomembnost uporabe lokalnih podatkov za učinkovitejše spremljanje turističnih tokov, kar prispeva k bolj trajnostnemu in odpornejšemu razvoju turizma. Projekt Planinstvo 4.0 in pobuda Turizem 4.0 ponujata praktične primere, kako lahko lokalni podatki omogočajo podrobnejše razumevanje vedenja obiskovalcev in izboljšajo upravljanje destinacij, zlasti v času izrednih situacij. Vendar pa obstoječe pravne omejitve in pomanjkanje dostopa do specifičnih podatkovnih virov ovirajo polni potencial teh pristopov. Članek zato poziva k zakonodajnim reformam, ki bi olajšale dostop do ključnih podatkov ter spodbudile širše deljenje podatkov med vsemi deležniki, s čimer bi se okrepila pripravljenost in odzivnost turističnih destinacij. S tem bi prispevali k bolj trajnostnemu, prilagodljivemu in vključujočemu turističnemu ekosistemu, ki bi bolje služil interesom lokalnih skupnosti in obiskovalcev.

VIRI IN LITERATURA

[1] Blejci nezadovoljni s turizmom: onesnaževanje in višanje stroškov življenja (2024, May 15). 24ur.com. https://www.24ur.com/novice/slovenija/blejci-nezadovoljni-s-turizmom.html
[2] Crisis Management and Governance in Tourism (2024, May 5). Making EU tourism resilient. https://eismea.ec.europa.eu/crisis-management-and-governance-tourism_en

[3] D3HUB Competence Centre (2024, May 15). https://www.d3hub-competencecentre.eu/
[4] Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. (2015). Smart tourism: Foundations and developments. Electronic Markets, 25(3), 179–188. https://doi.org/10.1007/s12525-015-0196-8
[5] Juvan, E., & Dolnicar, S. (2016). Measuring environmentally sustainable tourist behaviour. Annals of Tourism Research, 59, 30-44. https://doi.org/10.1016/j.annals.2016.03.006
[6] Kaj so odprti podatki? (2024, May 5) OPSI. https://podatki.gov.si/posredovanje-podatkov/kaj-so-odprti-podatki
[7] Kirsanova, E., Mokhirev, A., Sokolov, A., Suvorova, E., & Zikirova, S. (2021). Platform Cooperativism—A New Model in the Knowledge Economy (pp. 141–147). https://doi.org/10.1007/978-3-030-57831-2_15
[8] maPZS (2024, May 5). https://mapzs.pzs.si/home/trails
[9] Novo interaktivno orodje Dnevni turistični utrip Slovenije (2024, May 19). https://www.slovenia.info/sl/novinarsko-sredisce/sporocila-za-javnost/27353-novo-interaktivno-orodje-dnevni-turisticni-utrip-slovenije
[10] Prvi korak do vzpostavitve podatkovnih prostorov in izmenjave podatkov na zaupanja vreden način (2024, May 2). https://www.gov.si/novice/2024-01-18-prvi-korak-do-vzpostavitve-podatkovnih-prostorov-in-izmenjave-podatkov-na-zaupanja-vreden-nacin/
[11] Thousands protest in Spain's Canary Islands over mass tourism (2024, May 5). Euronews with EBU. https://www.euronews.com/2024/04/21/thousands-protest-in-spains-canary-islands-over-mass-tourism
[12] Zakon o prijavi prebivališča (2024, May 2). Pravno-informacijski sistem Republike Slovenije. https://pisrs.si/pregledPredpisa?id=ZAKO6046

▪

Dr. Urška Starc Peceny je strokovnjakinja za inovacije na področju poslovnega komuniciranja in novih tehnologij. Izobrazbo je pridobila v Sloveniji, Italiji in Avstriji, kjer je leta 2001 je doktorirala na Univerzi v Salzburgu z disertacijo z naslovom »Netlife«, ki je utirala pot raziskovanju sodelovalnih modelov z uporabo umetne inteligence. Ima več kot dvajset let mednarodnih podjetniških izkušenj s področja digitalnih inovacij, novih medijev in poslovnega komuniciranja. Kot vodja inovacij v podjetju Arctur vodi oddelek Turizem 4.0, namen katerega je preoblikovati današnjo turistično industrijo in so-ustvarjati nove storitve in izdelke za nenehno izboljševanje naše skupne prihodnosti. Ponosna je, da je del več strateških projektov, ki soustvarjajo evropski razvoj, vključno s podatkovnimi prostori za turizem, kulturno dediščino in medije. Zelo je navdušena nad digitalnim inoviranjem dediščine. Je docentka in gostujoča predavateljica za področje pametnih tehnologij na mednarodnih konferencah in univerzah. Nenehno v gibanju, živi med Slovenijo in Avstrijo, kjer je tudi vodja NASA Space Apps Challenge.

▪

Tomi Ilijaš je ustanovitelj in direktor podjetja Arctur in je diplomiral Fakulteti za elektrotehniko Univerze v Ljubljani. Gospod Ilijaš je podjetnik, ki se osredotoča na visokotehnološke inovacije in sodeluje pri številnih raziskovalnih projektih doma in v tujini. Uvajal je napredne poslovne modele in prebijal led pri uvajanju superračunalništva in umetne inteligence v mala in srednja podjetja širom Evrope. Bil je član IAC (Innovation Advisory Council) evropskega programa PRACE in predstavnik Slovenije v EURO HPC Research and Innovation Advisory Group (RIAG) in V zadnjem času se osredotoča na prenos ključnih tehnologij Industrije 4.0 na druga področja, predvsem v zdravstvo in turizem.

Zastrupljanje protokolov za razreševanje imen na lokalnih omrežjih

Urban Dopudja, Matevž Pesek
Univerza v Ljubljani, Fakulteta za Računalništvo in Informatiko, Večna pot 113, 1000 Ljubljana
ud74172@student.uni-lj.si, matevz.pesek@fri.uni-lj.si

Izvleček

V kontekstu povezovanja različnih informacijskih sistemov je razreševanje domenskih naslovov ključni proces identifikacije deležnikov v širšem okolju IT infrastrukture, ki ob pomanjkljivi konfiguraciji lahko predstavlja tveganje za zlorabo s strani napadalcev. Zaradi rastoče kompleksnosti infrastrukture se količina takšnih vektorjev napada na informacijske sisteme v zadnjem času povečuje. V pričujočem članku se poglobimo v delovanje protokolov za več vrstno oddajanje (angl. multicast) razreševanje imen v omrežjih ter njihovo potencialno zlorabo. Na tipičnih primerih pokažemo načine izrabe različnih orodij, s katerimi lahko relativno enostavno izvedemo takšne napade. Skladno z demonstracijo napadov nato prikažemo različne tehnike, s katerimi je mogoče prikazane napade zadostno omejiti.

Ključne besede: DNS zastrupljanje, LLMNR, Omrežna varnost, Šifrirni algoritmi

LOCAL NETWORK NAME RESOLUTION POISONING

Abstract

In the context of connecting different information systems, the resolution of domain addresses is a key process of identification of stakeholders in the wider environment of the IT infrastructure, which in the case of faulty configuration can pose a risk of abuse by attackers. Due to the growing complexity of the infrastructure, the amount of such attack vectors on information systems has been increasing recently. In this article, we delve deeper into the operation of protocols for the multicast name resolution in networks and their potential abuse. On typical examples, we show ways of using various tools that can be used to carry out such attacks relatively easily. According to the demonstration of the attacks, we then show various mitigations of the displayed attacks, with which the displayed attacks can be sufficiently limited.

Key words: DNS poisoning, LLMNR, Network security, Hash algorithms

1. UVOD

V zadnjih dveh letih smo bili priča napadom na večje ukrajinske organizacije pred začetkom ruske invazije leta 2022 [4, 13]. Ti napadi so razkrili načine razširjenega zasega zgoščenih poverilnic z namenom dešifriranja le-teh in njihove uporabe v obsežnih napadalskih kampanjah na ukrajinsko internetno in komunikacijsko infrastrukturo. Ti napadi so se izkazali za efektivne, hkrati pa je odziv nanje pokazal več enostavnih prijemov, ki so takšne napade v nadaljevanju vojne odbili ali vsaj efektivno omejili. V okoljih Windows je izkoriščanje ranljivosti, zlasti s tehnikami kot sta LLMNR in NBT-NS zastrupitev, pogosto [11]. Te tehnike se pogosto uporabljajo v omrežjih za preusmerjanje prometa in krajo poverilnic, kar ogroža varnost celotnega sistema. Manipulacija proce-

sov DNS razreševanja je ključni del teh napadov, saj omogoča napadalcem, da prestrežejo in preusmerijo omrežni promet [15]. Avtomatizacija takih napadov, ki jo omogočajo orodja, kot sta Metasploit ali Responder, povečuje tveganje za varnost predvsem sistemov v okolju Windows domen [2]. Ta orodja olajšajo napadalcem izvajanje kompleksnih napadov, ki bi sicer zahtevali več tehničnega znanja in izkušenj.

V kontekstu potencialnih ranljivosti v praksi se je nadzor nad avtentikacijskimi procesi izkazal kot ključen. Freimanis idr. so analizirali vpliv avtentikacijskih metod na splošno varnost računalniških sistemov večjih organizacij [6]. Njihove ugotovitve, ki temeljijo na več izvedenih penetracijskih testih, kažejo, da je strikten nadzor nad podprtimi avtentikacijskimi algoritmi ključnega pomena pri zagotavljanju zaupnosti in celovitosti računalniških sistemov [17]. Zlasti je to pomembno pri preprečevanju t.i. "pass-the-hash" in "pass-the-ticket" zlorab, ki so med najpogostejšimi napadi na Windows sisteme [14]. Te zlorabe omogočajo napadalcem pridobivanje dostopa do omrežnih virov brez dejanske pridobitve gesel, kar dodatno poudarja potrebo po strogih varnostnih ukrepih.

Naš cilj je poglobljeno raziskati zlorabo protokolov za razreševanje imen z večvrstnim oddajanjem, (angl. multicast) z namenom zajetja poverilnic kot delu kompleksnejšega napada na IT infrastrukturo. Demonstracija in analiza takšnih napadov nam omogoča vzpostavitev varnejše in bolj odporne infrastrukture proti tovrstnim napadom. V nadaljevanju članka najprej predstavimo tehnične potrebe za delovanje protokolov, ki so potencialno ranljivi — Link-Local Multicast Name Resolution (LLMNR), NetBIOS Name Service (NBT-NS) in Multicast DNS (mDNS). Nato obravnavamo načine zlorabe teh protokolov z različnimi orodji ter predstavimo lastno okolje za avtomatizacijo tovrstnih napadov. članek zaključimo s pregledom obrambnih mehanizmov za zaščito pred takšnimi napadi.

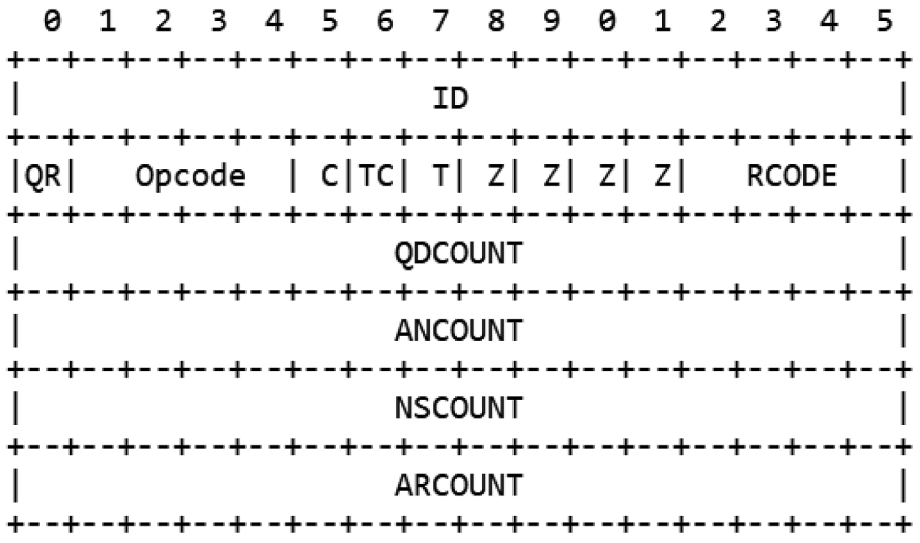
2 TEHNIČNE SPECIFIKACIJE RAZREŠEVALNIH PROTOKOLOV

2.1 Link-Local Multicast Name Resolution

Link-Local Multicast Name Resolution (LLMNR) [1] je protokol druge plasti ISO/OSI modela, ki ponuja alternativo (ali t.i. »fallback«) DNS-u za razreševanje imen v lokalnih omrežjih. LLMNR deluje decentralizirano po principu poizvedb večvrstnega oddajanja znotraj lokalnega omrežja, s katerim zagotavlja učinkovito razreševanje imen brez potrebe po centralizirani DNS infrastrukturi, vendar pa je zaradi njegove narave lahko zlorabljen v okviru kibernetskih napadov.

LLMNR deluje na vratih 5355, pri čemer so IPv4 poizvedbe poslane na naslov za večvrstno oddajanje 224.0.0.252, IPv6 poizvedbe pa na naslov FF02::1:3. V kontekstu LLMNR so gostitelji (angl. hosts) običajno konfigurirani tako kot pošiljatelji kot tudi odzivniki, lahko pa so tudi izključno pošiljatelji (vendar ne obratno), saj mora vsak gostitelj, konfiguriran kot odzivnik, delovati tudi kot pošiljatelj z namenom zagotavljanja edinstvenosti imen.

Postopek razreševanja se odvija v zaporedju, kjer pošiljatelj sproži poizvedbo, na katero nato odgovori odzivnik. Odgovor se pošlje nazaj pošiljatelju kot večvrstni ali enovrstni UDP paket, odvisno od narave poizvedbe. Format LLMNR paketa (poizvedba ter odgovor) temelji na formatu DNS-a, kateri je definiran v standardu RFC1035 - razdelek 4. Standard RFC predvideva pošiljanje UDP paketov znotraj dovoljenih velikosti z namenom izogibanja drobljenju (oz. fragmentaciji) - priporočljivo do 512 oktetov. Implementacija protokola pa lahko sprejme UDP pakete do velikosti največje enote prenosa (angl. maximum transmission unit - MTU) ali 9194 oktetov – velikost Ethernet jumbo 9 KB okvirja, z odštetimi 22 okteti za glavo ter oznaki navideznega omrežja (VLAN) in CRC kode.



Slika 1: **Format zaglavja paketa[1]**

- ID: 16-bitni identifikator, dodeljen poizvedbam, ki pošiljateljem omogoča ujemanje odgovorov. Zaradi varnosti je nastavljen na psevdonaključno vrednost.
- QR: 1-bitno polje, ki označuje, ali je sporočilo odgovor (set) ali poizvedba (clear).
- OPCODE: 4-bitno polje, ki določa vrsto poizvedbe.
- C: Označuje konflikt v poizvedbi ali edinstvenost imena v odgovoru.
- TC: Določa prirezovanje (truncation) zaradi omejitev dolžine. Če je nastavljeno v odgovoru, mora pošiljatelj ponovno poslati poizvedbo prek TCP.
- T: Označuje pogojni odgovor, če oseba, ki je odgovorila, ni preverila edinstvenosti imena.
- Z: Rezervirano za prihodnjo uporabo, trenutno nastavljeno na 0.
- RCODE: Koda odziva, nastavljena v odgovorih. V poizvedbah mora biti nič. RCODE, ki ni ničel, v odgovorih za večvrstno oddajanje vodi do poizvedbe TCP.
- QDCOUNT, ANCOUNT, NSCOUNT, ARCOUNT: 16-bitna nepredznačena (unsigned) števila, ki določajo število vnosov v različnih delih sporočila. Upoštevati mora določena pravila, da se prepreči tiho zavrženje s strani pošiljateljev ali prejemnikov.

2.2 Multicast DNS

Protokol mDNS[3] deluje na podoben princip in služi podoben namen kot LLMNR, le da je večinoma uporabljen v energetsko omejenih napravah/vgrajenih ter operacijskih sistemih kot so Linux ter MacOS, za razliko od LLMNR, ki je primarno uporabljen v Windowsu. mDNS se razlikuje tudi v tem, da IPv4 poizvedbe sprejema na naslov 224.0.0.251, IPv6 poizvedbe pa na FF02::fb. Kljub vsemu, novejše Windows različice za razreševanje pogosto uporabijo kar oba protokola, kot je razvidno iz spodnje slike prometa, zajetega s programom Wireshark, kjer smo izvedli poizvedbo po imenu (angl. hostname) “abc”.

10.0.2.15	224.0.0.251	MDNS	69 Standard query 0x0000 AAAA abc.local, "QM" question
10.0.2.4	10.0.2.15	MDNS	79 Standard query response 0x0000 A 10.0.2.4
fe80::d584:cf84:4b6…	ff02::fb	MDNS	89 Standard query 0x0000 AAAA abc.local, "QM" question
fe80::d584:cf84:4b6…	ff02::1:3	LLMNR	83 Standard query 0xbf24 A abc
10.0.2.15	224.0.0.252	LLMNR	63 Standard query 0xbf24 A abc

Slika 2: **LLMNR in mDNS poizvedba ter odgovor**

Na sliki prikazujemo, kako naprava pošlje mDNS poizvedbo prek IPv4 ter IPv6, nato pa isto stori še z uporabo protokola LLMNR. V tem primeru, že na prvo povpraševanje odgovori napadalec.

2.3 NetBIOS Name Service

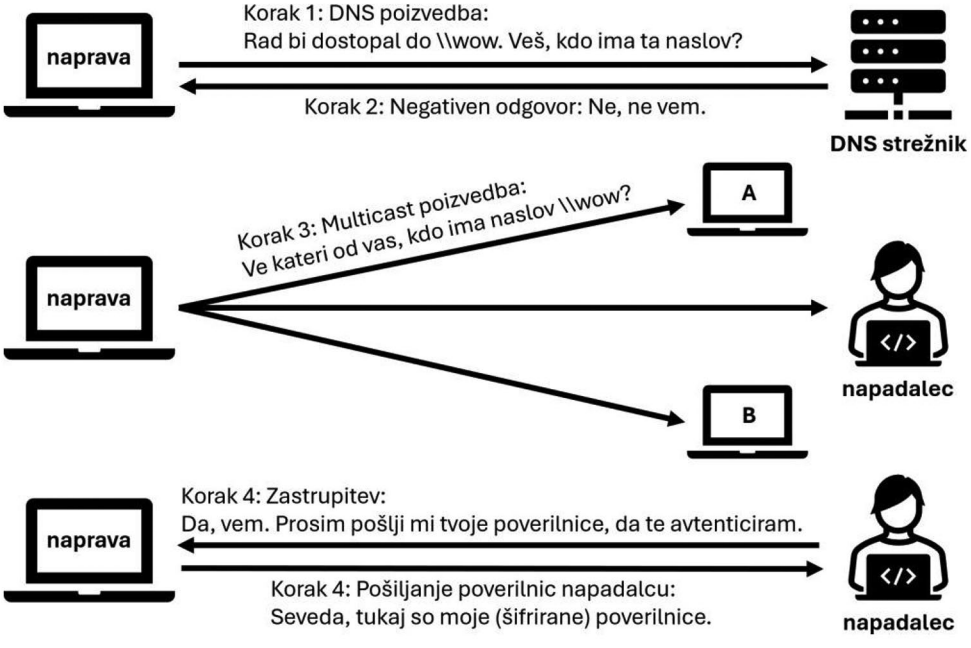
NBT-NS[10] je starejši protokol za razreševanje imen, ki je bil predvsem uporabljen v starejših Windows okoljih. Uporablja se za razrešitev NetBIOS imen v IP-naslove. Deluje preko UDP in uporablja vrata 137.

Za razliko od mDNS in LLMNR, ki sta bolj generična in delujeta na različnih operacijskih sistemih, je NBT-NS specifičen za okolja Windows. Deluje skupaj z drugimi protokoli, povezanimi z NetBIOS, kot je NetBIOS preko TCP/IP (NBT), in se uporablja predvsem zaradi povratne združljivosti v sodobnih Windows domenah/omrežjih. NBT-NS pravzaprav ne razrešuje domenska imena v IP-naslove, temveč v NetBIOS imena, ki so uporabljena za prepoznavanje Windows naprav ter storitev. Za razliko od ostalih dveh protokolov, NBT-NS podpira samo IPv4.

3 ZLORABA

V tem sklopu bomo najprej predstavili teoretično zlorabo omenjenih protokolov, nato pa prikazali praktični primer z uporabo orodja Responder. Kot potreben pogoj za izvedbo napada mora predhodno napadalec imeti dostop do omrežja.

Naprava, ki želi dostopati do nekega domenskega imena, katerega naslova še ne pozna, sprva pošlje poizvedbo na DNS strežnik. Če slednji ne poseduje ustreznega zapisa, naprava nato poplavi omrežje z vprašanjem po tem naslovu. Ta korak procesa predstavlja ključno ranljivost, pri kateri se lahko v komunikacijo vrinemo kot napadalec, ki na lokalnem omrežju posluša za tovrstnimi poizvedbami in ob prejeti poizvedbi sestavi "zastrupljen" odgovor, v katerem predstavlja sebe (ali drugo napravo) pod iskanim imenom. Poizvedujoča naprava nato od tarče zahteva poverilnice ali druge informacije, katere dobimo v obliki zgoščenih vrednosti ali pa celo kot golo besedilo (plain text).



Slika 3: **Diagram poteka "zastrupitve"**

3.1 Orodje RESPONDER

Za praktično izvedbo napadov smo uporabili orodje *Responder* [8]. Prikazali bomo šest tipov napadov z omenjenim orodjem v praktični obliki na simulacijskem okolju. Za simulacijsko okolje smo postavili tri virtualne naprave in jih povezali na isto virtualno omrežje (NAT network). Omenjene tri naprave so:

- Naprava Windows 11 s prizetimi nastavitvami — uporabljena kot tarča pri večini napadov,
- Naprava Kali Linux z orodjem Responder — uporabljena kot napadalec,
- Naprava Debian — uporabljena kot podpora napravam - na primer za Samba strežnik.

3.2 Zastrupljanje z uporabo SMB strežnika

Ko tarča poskusi dostopati do SMB strežnika, pošlje poizvedbo DNS strežniku, ki odgovori, da nima zapisa za to domensko ime. Tarča nato pošlje večvrstno poizvedbo po omrežju, ki jo lahko prestrežemo z uporabo orodja Responder *responder -I eth0* .

Responder tarči pošlje paket "Standard Query Response 0x0000 A", v katerem tarčo pozove, naj se avtenticira. Na Windows napravi se pojavi vpisno okno, kamor uporabnik vpiše poverilnice, in se pošljejo napadalcu. Napadalec prejme izpis v formatu:

```
Poisoned answer sent to <ip> for name
abc.local NTLMv2-SSP Client: <ip>
NTLMv2-SSP Username: <hostname>/<username>
NTLMv2-SSP Hash:
<username>::<hostname>:<hash>
```

3.3 Zastrupljanje z uporabo protokola WPAD

WPAD (angl. Web Proxy Auto-Discovery Protocol) je mehanizem za konfiguracijo omrežja, ki se uporablja predvsem v večjih organizacijah za samodejno odkrivanje posredniških (angl. proxy) strežnikov. Z WPAD protokolom lahko odjemalec poišče konfiguracijsko datoteko posrednika, ki se običajno nahaja na spletnem strežniku v lokalnem omrežju. Konfiguracijska datoteka vsebuje navodila o odjemalčevem dostopu do interneta, vključno s tem, kateri posredniški strežnik naj uporabi in katera vrsta prometa naj bo usmerjena

preko slednjega. Če uporabnik poskuša dostopati do neveljavnega URL naslova (npr. skozi brskalnik),

DNS strežnik ne bo vseboval imel zapisa za iskano stran. Brskalnik bo, če ima vklopljeno funkcionalnost "automatic configuration detection", poslal večvrstno povpraševanje po omrežju, v katerem povprašuje po WPAD strežniku.

To funkcionalnost brskalnika lahko zlorabimo z uporabo "-w" zastavice pri zagonu programa Responder, katera vzpostavi zlonamerni WPAD strežnik. Ko uporabnik zahteva konfiguracijsko datoteko, jo le-ta pozove za poverilnice. Tako zopet pridobimo NTLMv2(/SSP) zgoščene vrednosti poverilnic, v nekaterih primerih pa celo v golem tekstu.

3.4 Prisilna uporaba osnovne avtentikacije

Osnovna (angl. basic) avtentikacija uporablja golo besedilo za pošiljanje poverilnic — to je nešifrirana oblika, ki jo je mogoče neposredno prebrati. Dešifriranje lahko dolgotrajen in zahteven proces, zato bi bilo z vidika porabe časa in procesorske moči najlažje, da poverilnice izmenjujemo v goli obliki, kar pa predstavlja veliko grožnjo varnosti. Za ta primer nadaljujemo s prej opisanim napadom z strežnikom WPAD, ki mu v konfiguraciji orodja Responder dodamo zastavico "-b", ki prisili uporabnike v uporabo osnovne avtentikacije. Na tem mestu bi radi izpostavili, da to deluje v relativno redkih primerih. V primeru WPAD napada, ki se izvede skozi brskalnik, uporabnik prejme opozorilo, da se njegove poverilnice ne bodo varno prenesle, številni drugi programi in storitve pa avtomatsko zavržejo povpraševanje, če je avtentikacija nastavljena na osnovno. V primeru, da žrtev vpiše poverilnice, jih prejmemo v formatu:

```
Basic Client : <ip>
Basic Username : <username>
Basic Password : <password>
```

3.5 Prisilen spust šifrirnega algoritma iz NTLMv2-SSP na NTLM

Windows za šifriranje poverilnic in ostalih informacij privzeto uporablja šifrirni algoritem NTLMv2--SSP, ki je nadgradnja algoritma NTLM (angl. NT LAN Manager)[16] z dodatkom SSP (angl. Security Support Provider). Na sistemu Windows SSP predstavlja dinamično knjižnico (angl. Dynamic Link Library - DLL), ki ponuja vmesnik med operacijskim sistemom in različnimi avtentikacijskimi protokoli in tako omogoča okolju Windows razširjen nabor podprtih protokolov.

SSP je v kontekstu opisanega napada zanimiv z vidika funkcionalnosti ESS (angl. Extended Session Security), katera doda "SSP" zastavico v zgoščene NTLM vrednosti, in s tem podaljša SSP zgoščeno vrednost, zaradi katere je poverilnice težje dešifrirati.

Ta korak v več procesih poznamo pod imenom soljenje (angl. salting). V tem primeru napada orodje Responder konfiguriramo z zastavico "–disable-ess", s katero prisilimo tarčo, da poverilnice pošlje v obliki NTLMv2 zgoščene vrednosti, kar pomaga pri zmanjšanju časa, ki ga rabimo za dešifriranje.

V nekaterih primerih lahko dodatno omejimo kakovost zgoščevalnega algoritma z dodatno zastavico "–lm", s katero uporabnikovo napravo silimo v uporabo protokola NTLMv1, kar še dodatno zniža nivo varnosti. Vredno je tudi omeniti, da tovrstna prisila lahko privede do opozoril ali prekinitve seje s strani tarče, vendar je med našim testiranjem do takšnih opozoril prihajalo redko, za razliko od osnovne avtentikacije.

3.6 Zloraba posredovanja

Posredovanje (angl. relaying) je pogosto uporabljen način za nepooblaščen dostop do sistema. Deluje po principu posrednika, ki prejme veljavno avtentikacijo in nato to zahtevo posreduje drugemu strežniku ali sistemu ter se poskuša avtenticirati tem strežniku z uporabo prejetih poverilnic. Pred takimi napadi se lahko efektivno zavarujemo s podpisovanjem, vendar različni sistemi tega zaščitnega koraka ne uporabljajo [12], ali pa ga celo ne podpirajo. Princip takšnega napada smo testirali na Samba strežniku.

Uporabimo ukaz:

nmap -p445 –script=smb-security-mode <IP-naslov tarče>,

ki preišče vrata 445 (privzeta vrata za SMB) na tarči in preveri varnostno stanje konfiguracije Samba strežnika. Pridobimo odgovor, iz katerega lahko razberemo, da je podpisovanje izklopljeno.

```
Host script results:
| smb-security-mode
| account_used: guest
| authentication_level: user
| challange_response: supported
| message_signing: disabled (dangerous, but
  default)
```

Za napad nato uporabimo skripto "MultiRelay.py", ki jo lahko najdemo med seznamom orodij v orodju Responder. Skripti z zastavico "-t" nastavimo tarčo (kamor bodo poverilnice posredovane), ter z zastavico "-u" izvore, iz katerih sprejemamo poverilnice.

Vredno je tudi omeniti, da ta skripta ni bila posodobljena že od leta 2016 in je za njeno delovanje potrebna manjša prilagoditev — *thread.Daemon = True*.

Po zagonu skripte, ki je poslušala za poizvedbami na omrežju, smo uspešno izvedli posredovanje poverilnic Samba strežniku in tako pridobili dostop do strežnika.

3.7 DNS vrivanje v DHCP odgovoru

Če se v omrežju uporablja DHCP za identifikacijo IP-naslovov strežnikov, lahko orodje Responder v DHCP načinu tarči v odgovor podtakne lažni DNS zapis.

Orodje Responder lahko vzpostavi lažni DNS strežnik[5]. Ko žrtev poskuša dostopiti do naslova, najprej razreši ime z iskanjem DNS strežnika, kar stori s pošiljanjem DHCP zahteve. Responder odgovori na to zahtevo in v DHCP odgovor vstavi svoj IP-naslov DNS strežnika, in tako zastrupi odgovor. Ko žrtev to prejme, vidi IP-naslov lažnega DNS strežnika in z njegovo pomočjo poskuša dostopati do strežnika/storitve, vendar nevede dostopa le do napadalca.

Responder lahko zaženemo v DHCP-DNS poisoning načinu z zastavico "-D".

4 LLMNR AUTOMATION

Za izvedbo takšnih napadov lahko tudi avtomatiziramo opisane postopke in jih posledično tudi poenostavimo, na primer v primeru obsežnejših napadov z več tarčami. V ta namen smo ustvarili skripto LLMNRAutomation.sh in konfiguracijsko datoteko LLMNRAutomation.conf, ki se nahajata v našem javnem GitHub repozitoriju: https://github.com/Urichh/LLMNR-automation.

Skripta deluje v štirih korakih, izmed katerih dva slonita na drugih orodjih, ki sta potrebni za pravilno delovanje skripte. Ti dve orodji sta Responder (https://github.com/SpiderLabs/Responder) ter Hashcat (https://github.com/hashcat/hashcat). Obe orodji sta prosto dostopni. Omenjeni štirje koraki so:

1. Branje konfiguracijske datoteke in začetek poslušanja z želenimi nastavitvami.
2. Zajem zgoščenih poverilnic.
3. Organizacija zajetih poverilnic v logično datotečno strukturo za lažje dešifriranje.
4. Dešifriranje zgoščenih poverilnic.

V tem sklopu bomo razložili kako s pomočjo skripte avtomatizirano pridemo do dešifriranih poverilnic.

4.1 Konfiguracija

V repozitoriju se poleg LLMNRAutomation.sh skripte nahaja tudi konfiguracijska datoteka LLMNRAutomation.conf, v kateri so nastavitve s katerimi se nato zažene skripta. Datoteka vsebuje šest glavnih razdelkov:

- Vmesnik (angl. interface) – nastavitev vmesnika, na katerem skripta posluša in oddaja. Privzeta vrednost: eth0.
- Želeni strežniki – uporabnik vklopi ali izklopi lažne strežnike, ki jih skripta nato zažene. Privzeto so vsi vklopljeni.
- Preferirana avtentikacijska metoda - uporabnik nastavi šifrirni algoritem. Privzeta vrednost: NTLMv2-SSP.
- Lažni (angl. rogue) WPAD strežnik - uporabnik vklopi ali izklopi, če skripta zažene lažni WPAD strežnik za odgovore na DHCP poplavljanja. Ta nastavitev strežnika je ločena od drugih, ker se žrtev na WPAD strežnik ne povezuje neposredno, temveč je uporabljen v kombinaciji z drugimi. Privzeta vrednost: Off.
- DHCP-DNS vrivanje - uporabnik vklopi ali izklopi DHCP-DNS vrivanje, kot je razloženo v "DNS injection v DHCP odgovoru" razdelku poglavja o orodju responder. Privzeta vrednost: Izklopljena.
- Lažni zunanji IP-naslov - uporabnik lahko izbere lažni IP-naslov, iz katerega bo tarča prejela zastrupljene odgovore. Privzeta vrednost: None.

4.2 Organizacija zajetih poverilnic

Ko uporabnik ustavi skripto, se zajete zgoščene poverilnice shranijo v direktorij imenovan "hashes" znotraj direktorija v katerem se nahaja skripta. Te zgoščene vrednosti so urejene po IP-naslovih tarč, znotraj katerega so urejeni po protokolu in nazadnje, znotraj tekstovnih datotek, so ločene po uporabniških imenih, kot je prikazano v spodnjem diagramu:

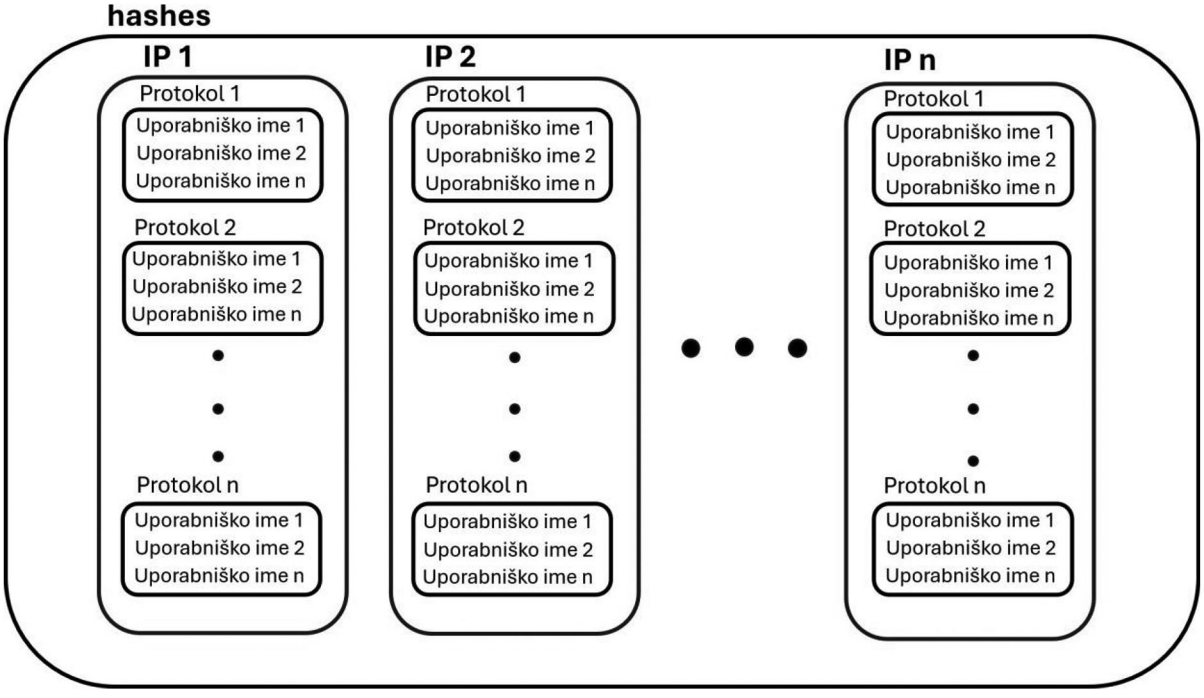
Ta datotečna struktura pomaga pri izbiri optimalnih zgoščenih vrednosti za dešifriranje, da se čim lažje prebijemo do želenih poverilnic.

4.3 Dešifriranje poverilnic

Ko zaključimo fazo zajema poverilnic, lahko preidemo na zadnjo fazo, ki je dešifriranje poverilnic. To storimo tako, da skripto zaženemo z zastavico "-c":

./LLMNRAutomation.sh -c

Če skripto zaženemo, dobimo izpis vseh zajetih poverilnic urejenih po IP-naslovu, ter nato po uporabniškem imenu. Npr:



Slika 4: **Organizacija zgoščenih poverilnic**

```
IP: <ip>
• uporabniško ime 1
• uporabniško ime 2
• ...
• uporabniško ime n
```

Nato lahko izberemo katero zgoščeno vrednost želimo dešifrirati, kar storimo z ukazom:

./LLMNRAutomation.sh -c -i <ip> -u <uporabniško ime>

Ta modul skripte nato dešifrira izbrane poverilnice z uporabo orodja Hashcat[7]. Za izbrano uporabniško ime, povezano z izbranim IP-naslovom, samodejno izbere najlažje zgoščene vrednosti za dešifriranje, ki si sledijo od najlažje do najtežje v slednjem zaporedju:

1. Golo besedilo (angl. plain text)
2. NTLMv1
3. NTLMv1-SSP
4. NTLMv2
5. NTLMv2-SSP

Hashcat poleg vhodne in izhodne datoteke zahteva še dodatni argument, ki predstavlja vrsto podane zgoščene vrednosti. Skripta to vrednost zazna avtomatsko, glede na spodnjo tabelo[9]:

Tabela 1: **Vrste algoritmov**

Algoritev	**Vrsta**
Golo besedilo	
NTLMv1	1000
NTLMv2	5600

Skripta nato začne z dešifriranjem poverilnic, vendar čas dešifriranja lahko močno variira glede na kompleksnost in dolžino gesla. Nazadnje skripta še shrani dešifrirane poverilnice v besedilno datoteko znotraj direktorija "cracked", ki je v istem direktoriju kot skripta.

5 DISKUSIJA IN ZAKLJUČEK

Čeprav so predstavljeni napadi lahko zelo nevarni, obstajajo številni obrambni mehanizmi, ki jih lahko preprečijo, ali pa vsaj minimizirajo posledice. Skrbniki IT okolij lahko, razen v primeru kjer narava organizacije to preprečuje, tovrstno razreševanje izklopijo, kar lahko storijo npr. kar preko upravljanja s politiko skupine (angl. group policy). V primeru, da si organizacija tega ne more privoščiti, pa lahko k varnosti pripomorejo z implementacijo omejenih dostopov do omrežja (ang. network access control) kot npr. protokol 802.1x. Poleg tega, se efektivnost takih zlorab lahko močno zmanjša s splošno dobrimi varnostnimi praksami, kot so podpisovanje zahtevkov/odgovorov na SMB strežnikih, preprečevanje uporabe zastarelih šifrirnih algoritmov, močna gesla, ki otežujejo dešifriranje gesel in ostale splošne dobre prakse infrastrukturne varnosti, kot so ločitev (segmentacija) omrežij.

V modernih računalniških sistemih je še vedno veliko vidikov, ki bodisi zaradi lahkote uporabe, povratne združljivosti ali drugih razlogov lahko predstavljajo varnostne luknje. Napadi, predstavljeni v tem članku po večini ne predstavljajo takojšnje neposredne grožnje za varnost računalniških sistemov, saj je za izvedbo takšnega napada potreben dostop do lokalnega omrežja, znotraj katerega tovrstni protokoli za razreševanje niso blokirani. Prav tako pa imajo napadalci ob uspešno izvedenem napadu pred sabo še mnogo ovir, kot so dejansko dešifriranje poverilnic, ki je ob ustrezno zahtevni kompleksnosti gesel in močnih šifrirnih algoritmih lahko zelo časovno potratno, poleg tega pa lahko zelo pomagajo tudi ostali preventivni ukrepi, kot so večstopenjska avtentikacija in podobni prijemi. Kljub vsemu pa je pomembno tem zlorabam posvetiti pozornost, saj za zagotavljanje varnosti vseeno želimo minimizirati potencialno ranljive vidike in tako zmanjšati število potencialnih vektorjev napada.

V tem članku smo predstavili delovanje protokolov za razreševanje imen v lokalnih omrežjih z uporabo večvrstnih poizvedb, ter kako lahko te protokole zlorabljajo napadalci z namenom zasega šifriranih poverilnic. Demonstrirali smo tudi delovanje orodij za tovrstne napade ter različne uporabe le-teh, kar smo nato nadgradili v lastno orodje za avtomatizacijo napadov in za konec predstavili še efektivne obrambne mehanizme. Hitra rast procesorske moči za namene dešifriranja poverilnic, neodpornost trenutnih šifrirnih algoritmov na kvantne računalnike, vedno večje kompleksnosti omrežij, večja uporaba mrežnih storitev namesto »tradicionalnih« namiznih programov in ostali dejavniki so razlogi, zaradi katerih menimo, da je zaščita pred takšnimi napadi v današnjem svetu ključnega pomena.

LITERATURA

[1] B Aboba, D Thaler in L Esibov. *RFC 4795*. English. Jan. 2007. URL: https://www.rfc-editor.org/ rfc/rfc4795.html (pridobljeno 5. 9. 2024).

[2] Iliano Cervesato. "Empirical Study of the Impact of Metasploit-Related Attacks in 4 Years of Attack Traces". English. V: *Advances in Computer Science - ASIAN* 2007. Doha, Qatar: Springer, dec. 2007, str. 198–211. ISBN: 3-540-76927-7. URL: https://link.springer.com/chapter/10.1007/978-3- 540-76929-3_19 (pridobljeno 21. 5. 2024).

[3] S Cheshire in M Krochmal. *Multicast DNS*. English. 2013. URL: https://datatracker.ietf.org/ doc/html/rfc6762 (pridobljeno 5. 9. 2024).

[4] R Cichocki. "State-Sponsored and Organized Crime Threats to Maritime Transportation Systems in the Context of the Attack on Ukraine". English. V: *TransNav. the International Journal on Marine Navigation and Safety of Sea Transportation* 17.3 (sep. 2023), str. 5. URL: https://bibliotekanauki. pl/articles/24811512.pdf.

[5] Maven Cybertech. *Using Responder to Capture Credentials. English.* Okt. 2023. URL: https :// systemweakness . com / using - responder - to - capture - the - credentials - a9d5a1013333 (prido- bljeno 5. 9. 2023).

[6] Davis Freimanis. "Vulnerability Assessment of Authentication Methods in a Large-Scale Computer System". English. Magistrsko delo. SCHOOL OF ELECTRICAL ENGINEERING in COMPUTER SCIENCE: KTH ROYAL INSTITUTE OF TECHNOLOGY, maj 2019. URL: https : / / www . diva - portal.org/ smash/get/diva2:1358687/FULLTEXT01.pdf (pridobljeno 5. 9. 2024).

[7] Radek Hranicky` in sod. "Distributed password cracking with BOINC and hashcat". V: *Digital Investiga- tion 30* (2019). Publisher: Elsevier, str. 161–172.

[8] William Hurer-Mackay. *LLMNR and NBT-NS Poisoning Using Responder.* English. Jun. 2016. URL: https : / / www . 4armed . com / blog / llmnr - nbtns - poisoning - using - responder/ (pridobljeno 5. 9. 2024).

[9] Nicklas Mortensen Hamang. "Effective Password Cracking". English. Magistrsko delo. Faculty of ma- thematics in natural sciences: University of Oslo, 2019. URL: https://www.duo. uio.no/bitstream/ handle/10852/73247/Nicklas_M_Hamang_ Master_Thesis.pdf (pridobljeno 5. 9. 2024).

[10] *NetBIOS over TCP/IPNetbio's NBT-NS Poisoning*. English. (Pridobljeno 19. 5. 2024).

[11] Mike O'Leary. "Attacking the Windows Domain". English. V: *Cyber Operations: Building, Defending, and Attacking Modern Computer Networks*. 2nd. Apress Media LLC, feb. 2019, str. 1151. ISBN: 978- 1-4842-4294-0. URL: https:// link.springer.com/chapter/10.1007/978- 1- 4842- 4294- 0_8 (pridobljeno 5. 9. 2024).[12] Alexander Oberle in sod. "Preventing pass-the-hash and similar impersonation attacks in enterprise infrastructures". V: *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2016, str. 800–807.

[13] Konstantinos Pantazis. "An External Red Team Assessment in a Corporate Environment". English. Doktorska disertacija. Department of Information in Electronic Engineering: International Hellenic University of Greece, 2022. URL: https : / / www . researchgate . net / profile / Konstantinos - Pantazis - 8/ publication / 364958274 _ An _ External _ Red _ Team _ Assessment _ in _a_ Corporate _ Environment / links / 63610c3a8d4484154a53def7 / An - External - Red - Team - Assessment - in - a- Corporate-Environment.pdf (pridobljeno 5. 9. 2024).

[14] Abdurrahman Pektaş. "Practical Approach For Securing Windows Environment: Attack Vectors And Countermeasures". V: *International Journal of Network Security & Its Applications (IJNSA) Vol 9* (2017). URL: https://papers.ssrn.com/ sol3/papers.cfm?abstract_id=3649907.

[15] U Steinhoff, A Wiesmaier in R Araújo. *The State of the Art in DNS Spoofing*. English. 2006. URL: https : / / citeseerx . ist . psu . edu / document ? repid = rep1 & type = pdf & doi = 7fdd734e684c6eb79a61864bb418ddc93a6ac751 (pridobljeno 21. 5. 2024).

[16] Nuno Tavares. *NTLM vs KERBEROS*. English. Apr. 2018. URL: https://answers.microsoft.com/ en - us / msoffice / forum / all / ntlm - vs - kerberos / d8b139bf - 6b5a - 4a53 - 9a00 - bb75d4e219eb (pridobljeno 5. 9. 2023).

[17] Zhihao Zheng in sod. "Best Practices in Designing and Implementing Cloud Authentication Schemes". English. V: *CS & IT Conference Proceedings*. Zv. 11. Issue: 3. CS & IT Conference Proceedings, 2021, str. 10. URL: https://www.csitcp.com/ paper/11/113csit07.pdf.

▪

Urban Dopudja je študent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Čas posveča strokovnim izpopolnjevanjem na področju kibernetske varnosti. Njegovi raziskovalni interesi segajo na področja spletne varnosti, omrežnih protokolov in nizkonivojske analize sistemov.

▪

Matevž Pesek je docent in raziskovalec na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je diplomiral (2012) in doktoriral (2018). Od leta 2009 je član Laboratorija za računalniško grafiko in multimedije. Od leta 2024 izvaja predmet Varnost programov.

Iz Islovarja

Islovar je spletni terminološki slovar informatike, ki ga že več kot 20 let ureja jezikovna sekcija Slovenskega društva INFORMATIKA. Slovar je javno dostopen za vpoglede in vnašanje novih izrazov na naslovu http://www.islovar.org

oblák -a m *(angl. cloud)*
programske rešitve, računalniška okolja in informacijska infrastruktura, ki so na voljo kot storitev (2) na internetu

oblákovno prográmje -ega -a s *(angl. cloudware)*
programje, ki je dostopno in se izvaja v oblaku

oblákovno skladíšče -ega -a s *(angl. cloud storage)*
storitev(2), ki omogoča shranjevanje podatkov v oblaku

omrežje oblákov -a -- s *(angl. cloud storm, cloud network)*
povezava več oblakov; prim. mreženje oblakov

piramída obláka -e -- ž *(angl. cloud pyramid)*
predstavitev različnih ravni računalništva v oblaku, kjer so posamezne ravni ločene glede na vlogo v oblaku, n.pr. infrastruktura, računalniško okolje, programje

prenosljívost med obláki -i -- -- ž *(angl. cloud portability)*
sposobnost selitve programja in pripadajočih podatkov med oblaki in ponudniki storitev v oblaku

računálništvo v obláku -a -- -- s *(angl. cloud computing)*
uporaba oblaka in z njim povezane tehnologije; sin. računalniški oblak; prim. storitveno računalništvo, igranje v oblaku

stándard obláka -a -- m *(angl. cloud standard)*
dogovor o uporabi in upravljanju oblaka

strežnik v obláku -a -- -- m *(angl. cloud server)*
navidezni strežnik, na razpolago kot storitev v oblaku

zunánje izvájanje v obláku -ega -a -- -- s *(angl. cloud sourcing)*
uporaba storitev v oblaku, ki jih omogoča zunanji izvajalec

Izpitni centri ECDL

ECDL (European Computer Driving License), ki ga v Sloveniji imenujemo evropsko računalniško spričevalo, je standardni program usposabljanja uporabnikov, ki da zaposlenim potrebno znanje za delo s standardnimi računalniškimi programi na informatiziranem delovnem mestu, delodajalcem pa pomeni dokazilo o usposobljenosti. V Evropi je za uvajanje, usposabljanje in nadzor izvajanja ECDL pooblaščena ustanova ECDL Fundation, v Sloveniji pa je kot član CEPIS (Council of European Professional Informatics) to pravico pridobilo Slovensko društvo INFORMATIKA. V državah Evropske unije so pri uvajanju ECDL močno angažirane srednje in visoke šole, aktivni pa so tudi različni vladni resorji. Posebno pomembno je, da velja spričevalo v 148 državah, ki so vključene v program ECDL. Doslej je bilo v svetu v program certificiranja ECDL vključenih že preko 16 milijonov oseb, ki so uspešno opravile preko 80 milijonov izpitov in pridobile ustrezne certificate. V Sloveniji je bilo doslej v program certificiranja ECDL vključenih več kot 18.000 oseb in opravljenih več kot 92.000 izpitov. V Sloveniji sta akreditirana dva izpitna centra ECDL, ki imata izpostave po vsej državi.

AGORA

Micro Team

Znanstveni prispevki

Strokovni prispevki

Informacije

ISSN 1318-1882
9 771318 188001