Creating a framework for live data analysis of Discord servers

Filip Stenstuen Marthinsen

Haaga-Helia University of Applied Sciences
Bachelor of Business Information Technology
Bachelor thesis
2023

Abstract

Author(s)
Filip Stenstuen Marthinsen

Degree
Bachelor of Business Information Technology

Report/Thesis Title
Creating a framework for live data analysis of Discord servers

Number of pages and appendix pages
37 + 0

This thesis looks at how Discord intelligence could be done, discussing not only the technical aspects but also different practices that must be followed to be able to admit collected evidence to courts of law, with a focus on those in the European Union's Member States.

A rise in crime through Discord has resulted in tools for extraction of Discord logs from computers being made, but a framework for collecting evidence from live communications specialized to the use of intelligence is yet to be made. Thus, this thesis aims to address this by providing information to law enforcement and intelligence agencies that need to collect evidence from a live Discord server but are unsure about where to start.

The unsure future of intelligence for Discord is commented on, both from organizations with Discord's assistance (in the thesis, referred to as Discord intelligence with escalated privileges), as well as parties that for different reasons may not be able to acquire Discord's assistance and must perform intelligence as a normal end-user (Discord intelligence without escalated privileges).

Key words
E-evidence, digital forensics, Discord forensics.

Abbreviations

Abbreviation	Description
CA	*Certificate authority*. Centralized system that manages and distributes access certificates.
CLI	*Command-line interface*
CSAM	*Child sexual abuse material*
GUI	*Graphical user interface*
HTTPS	*Hypertext transfer protocol secure*. Secure protocol for the World Wide Web.
ISP	*Internet service provider*
PKI	*Public key infrastructure.* An infrastructure for encrypting and signing data, using asymmetric keys for decryption and / or validation.
OS	*Operating system*
RAID	*Redundant array of independent disks.* A system combining several physical disk drives to protect data from complete destruction by drive failure.
SSH	*Secure shell.* Protocol for remote CLI access.
URL	*Uniform resource locator*
VoIP	*Voice over internet protocol.* Protocol for voice communication over the internet.
VPN	*Virtual private network*

Table of contents

1 Introduction to the framework

In the digital age we currently are living in, the world is smaller than it has ever been before. A WhatsApp message from Finland can reach someone in California in less than a second, and one can even do video calls from anywhere in the world with just an internet connection and a phone or computer. This makes it easier than ever to cooperate with other businesses and persons across national borders, and one can even communicate with people or robots outside of planet Earth!

However, cross-border criminality has flourished as an effect of this, and investigations have become more complicated due to data servers or criminals being in a different country to where the victims are located. One prime example is tech call centre scams, where victims typically are in richer countries, and the call centres and their employees mainly are in poorer countries. The problem here is that the police in the victim's country has no right to do investigations in the country where the suspects are, and police in many poorer countries might not have the resources to organize large operations against organizations that are not a threat to their own citizens. Many call centres are also located in countries where jail sentences for scamming are relatively short, so call centres can easily get employees by offering them large monetary gains with low risks and repercussions.

Social media is also used to spread propaganda and recruit soldiers for terrorist groups. The article 'Cyber-Extremism: Isis and the Power of Social Media' (Awan 2017) discusses how Brits are enticed into aiding terrorist groups like Isis through propaganda spread on sites such as YouTube and Facebook, targeting very specific groups that might be easily influenced. One famous example is Isis' *one billion campaign*, a challenge for the Muslims of the world to post a total of one billion pictures, videos, or messages online in support for Isis. Isis even developed a free mobile application that would upload Isis-created content through the social media accounts of those who had downloaded it (Irshaid 2014).

The framework which the thesis explains automates evidence collection from live Discord servers, to help uncover and stop serious crimes before damage has been dealt. It also gives some recommendations for best practices on how to set up a system infrastructure for the framework code that gathers messages and other evidence collected from Discord chatting servers, such as how to do access control or protect the collected data to comply with most courts of law. There will be a focus on EU regulations for the law-side of the theoretical framework, as the thesis is done at Haaga-Helia University of Applied Sciences in Helsinki, Finland, and it was beneficial to select something specific to draw lines to whenever a law-discussion had to be made.

Online intelligence is extremely uncomfortable for most people – maybe even more so than old wiretaps, as they were only capable of listening in on vocal conversations. Complete access to a smart phone can result in the microphone and camera always being turned on without any signs, GPS signals would reveal exactly where one would be at any time, and it is possible to see everything else one might use their phone for; messaging, entertainment, internet searches or research, and even body vitals if a smart device monitoring those is connected.

As such, intelligence in this form is avoided when possible; but sometimes, the benefit of having knowledge prior to an event is too great to ignore. It is possible to return the money to the victim after a successful scam, but it is not possible to get back people that have died in terror attacks. In Susan Landau's book "Surveillance or Security? The Risks Posed by New Wiretapping Technologies", she writes "Privacy includes the right to control information about yourself, the right to associate as you wish, as privately as you wish, to share confidence, in confidence, the right to enjoy solitude and intimacy." (Landau 2010, pp. 10) She continues by acknowledging that it is not always possible to exercise these rights in modern society, both due to local jurisdictions and own circumstances, and sometimes law enforcement or intelligence agencies, with the power they have received from the state, decide that it is justifiable to infringe on those rights.

1.1 Motivation

In the video game Final Fantasy XIV, a quote is uttered multiple times throughout the story, sounding "For those we have lost. For those we can yet save." Several characters every player will come to love, die in the quest for saving the Source and all its residents, often after or before boss battles of great significance.

This quote really stuck to me and made me think of all the similarities between Final Fantasy XIV and our own world. There is so much meaningless suffering; innocent people dying in terrorist attacks, civilians caught up in wars between states, children that are abused for both monetary value and personal pleasure, and so much more. Some events are incredibly hard to stop; domestic violence and abuse might be hidden for years or decades, due to the victim's fear of repercussions and or social stigma around it. Even fear of not being believed by authorities, or a combination of these, might prevent a victim from speaking aloud about it.

I wanted to research how intelligence on Discord servers could be done, as it is one of the most used communication platforms today; both in general but also linked to several of these serious crimes. This thesis might not reach that many, and this framework might not be entirely useable in the future due to Discord's own decisions on the topic of intelligence, but I wanted to provide a start-point for how Discord intelligence could be done.

1.2 Scope of the framework

I have created a template code for running the evidence collection I will discuss later; this is open-source and available on GitHub (Watch-me-not code 2023). It has been made so that it easily can be further expanded upon, and to be easily scalable by splitting all data into individual fields and sending these to a local database for querying.

I will be discussing the politics around the field in EU, especially the e-evidence regulations and directives. I will also give some recommendations on both physical and digital security measures that should be taken to ensure that evidence is collected and stored in a forensically sound manner, in the event that the users of this framework do not have own frameworks for it.

Finally, I will be discussing the future of intelligence on Discord-servers, and how it might change due to laws or Discord's decisions on the matter. Without Discord's help, many things are incredibly difficult or impossible to do; thus, I highly encourage any law enforcement or intelligence agencies wishing to do investigations or gather evidence from live Discord-servers to contact them and ask for their cooperation.

1.3 The code and how it works

The main section of the code for evidence collection listens for specific events and does standardized actions based on what event is received. A Discord-event is a keyword sent to any users, bots or webhooks with sufficient permissions to receive the event in question and are caused by a Discord server member performing an action. An example of an event is *guildMemberAdd*, which is triggered and sent by Discord whenever a new user joins the server.

Events come with related information to what occurred; if a *guildMemberAdd* event is sent, it will be accompanied with the data of the member that was added. The framework code is designed to format and separate data into different variables, and then send these into a MySQL database for easier query and analysis.

For an investigator that needs to read the collected data, they would use SQL queries to find relevant information stored in the MySQL database. Usage of the framework and how the data arrives to the database is described in the figure below.

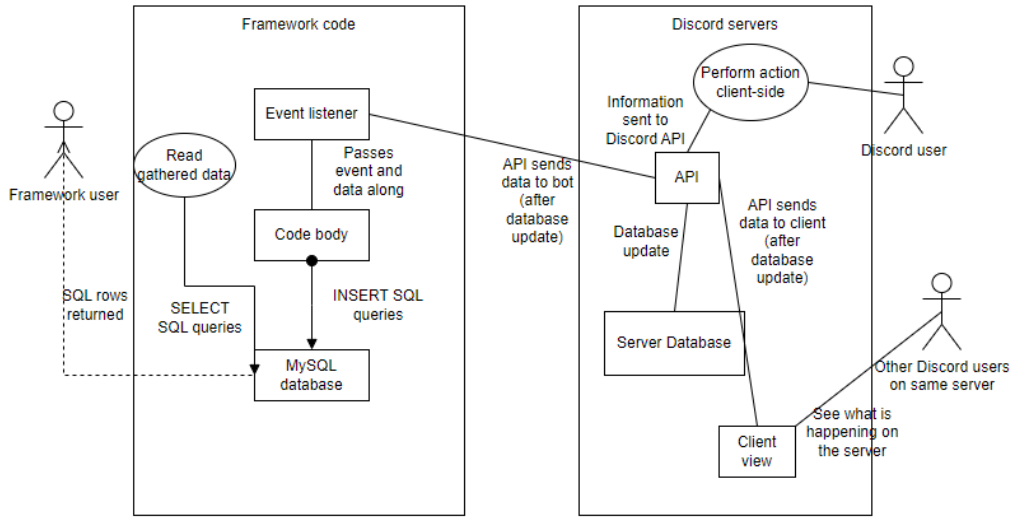

Figure 1. Functionality and usage of the code and MySQL database.

Some parts of the code are triggered on the event *ready*, which is a local event that is triggered when a bot is successfully connected to a server. The purpose of this code is to download, or fetch, all existing context: members that are already connected to the server, what roles exist on the server, what roles members have, what channels are available, and what emojis are available. To better understand exactly what the code does, I will include the code of what happens when the bot is fetching all server members and explain it in detail.

```
client.on('ready', async () => {
    // [...]
    var list = client.guilds.cache.get(process.env.GUILDID);
    await list.members.fetch();
    list.members.cache.forEach(member => {
        db_con.query('INSERT IGNORE INTO members VALUES (?, ?, ?, ?, ?, ?)',
[member.id, member.user.bot, member.joinedTimestamp, member.displayName,
member.nickname, member.pending]);
    });
    // [...]
});
```

Figure 2. Fetching guild members.

Line 1: the function on() of the Discord.js Client-object (nicknamed client) is called, with “ready” as the event keyword, is triggered whenever the bot is successfully connected to the server. Second parameter of the function is an async (asynchronous) arrow-function described in the later lines.

Line 3: A variable list is created, which contains a Map of what all properties the Guild has.

Line 4: The *await* keyword is used to ensure that all the members are fetched from the Guild Collection prior to performing any further code on them.

Line 5: A *forEach*-function is used to perform an action for each member in list.members.cache.

Line 6/7/8: A MySQL query is sent with member data of each member. INSERT IGNORE INTO allows the bot to be restarted without any problems, as by including the IGNORE keyword, any duplicate rows (primary key constraint) will be ignored. Placeholders are used so that MySQL’s Node.js-implementation can handle escaping the inputs. Escaping refers to formatting input in cases where any special characters are in place, such as an apostrophe (‘) which is used to denote the start or end of a string.

To make it easier for me to refer to the bot, I have given it the code name “Watch-me-not”: a wordplay on the flower myosotis, more known in the northern hemisphere as the forget-me-nots or scorpion grasses.

2 EU's E-evidence agreements and its benefits for digital evidence collection

On the 25th of January 2023, the EU Council approved the new e-evidence rules in a press statement, with the Swedish Minister of Justice, Gunnar Strömmer, commenting: "With this agreement we respond to a key request by our judicial authorities. More and more crimes are planned or committed online and our authorities need the tools to prosecute them as they do for crimes offline. The new rules will allow judges and prosecutors to quickly access the evidence they need, regardless of where it's stored, before it disappears." (European Council 2023)

The matter was first addressed in April 2015, when the Commission wrote the European Agenda on Security, which focuses on the importance of uncovering and shutting down terrorism, organised crime, and cybercrime (European Commission 2015, introduction to The European Agenda on Security). The Commission adopted the proposals for Regulation and Directives for cross-border sharing of electronic evidence for criminal investigations and expects to publish the final texts in the summer of 2023 (Migration and Home Affairs 2023).

2.1 Introduction to the agreements

With the cross-border aspects of most service providers offering their products to anyone regardless of their country of residence, and with USA reaching the highest evidence production requests/orders by far, the European Parliament saw a necessity to improve the methods and channels used by member states' law enforcement and service providers both inside of EU as well as other countries, most notably USA and Ireland.

Traditionally, most cross-border evidence production requests/orders would first have to be passed in the law enforcement's state, and then go through the court system of the service provider's state. In more urgent cases, police-to-police channels were used to shorten down the time necessary for a request/order to become active; however, this relied on police jurisdictions having established communication channels, and having a good history of prior cooperation. With the e-evidence regulations and directives, court decisions in the requesting/ordering law enforcement's state would be recognized by the service provider, or in the case of a service provider with a headquarters outside of the EU, a representative of the service provider within the EU, assuming domestic rulings applying to the service provider in their home country align.

2.2 Evidence acquired faster

Both in times of urgency and non-urgency, the regulations and directives aim to drastically reduce the time necessary to receive evidence from a service provider. Section C of the European Production Order Certificate (EPOC) includes a predetermined set of deadlines for compliance, where the non-urgent option is as soon as possible and within 10 days, and an urgent request with a deadline of 8 hours in cases where an imminent threat to life or physical integrity or safety of a person, or an imminent threat to critical infrastructure, is present (Regulation on Production and Preservation orders for electronic evidence in criminal proceedings).

2.3 Relevance to this framework

EU e-evidence is worded for and aimed mainly at evidence for police investigations in the aftermath of a crime. Police might be able to use this for intelligence in cases where crimes have already been committed in a platform such as Discord, and the possibility of a second, serious crime is high, and police would need assistance to for example locate the criminal prior to this. However, if a crime has not already been committed, EU e-evidence would likely not be applicable.

This could however change with inclusion of clauses for EU e-evidence to additionally cover intelligence. If it is allowed, EU e-evidence will surely prove detrimental for many intelligence missions in the future and using it frequently will help improving the information channels that are opened because of it.

3 Digital security of Discord evidence collection machines

The digital security of the evidence collection machines should be secure, but easily useable by all end users. This involves mainly using a remote connection to the evidence collection machine, such as SSH, to access data, and creating database clones on own analysis machines for investigations while the collection process is ongoing.

3.1 Intranet access only

This is perhaps self-explanatory, but in these times with a rising amount of home offices, restricting access to evidence should only be possible from direct intranet access. The risk of data leakage or tampering increases drastically if machines outside of the company building have access to it, as with only an IP address, a cyber adversary can start looking for and exploiting weaknesses in a server.

Allowing indirect intranet access through VPN is also a heightened risk factor, as the security given from having to pass through physical security at the company building is stripped away.

3.2 Role-based access control

Role-base access control is important for several reasons. Firstly, it allows for varying permission levels based on how the user will interact with the server, namely the principle of least privileges. It helps contain damage, both due to human error and rogue employees, as well as cases where login credentials are compromised by an unauthorized third party. Secondly, it allows for logging what each employee was doing when, and if no unauthorized people have access to an account, all actions can be attributed back to the owner of the user.

Microsoft advises to give roles and permissions to groups, and not directly to users (Microsoft 2022). This is due to several factors: firstly, it is easier to see what all permissions a group has, instead of having to click into every user. Secondly, in the case a permission is to be revoked, it is more reliable to remove that permission once from a group instead of several times from all users of a group.

3.3 Minimal services installed

Even if services are not given open ports, they can still tamper with the machine in different ways, both from a malignant version of a service, or a service that does what it is supposed to do but can have terrible effects on other facets of the server.

3.4 Usage of SSH certificates

SSH keys have often been the go-to way of strengthening the authentication process; however, they have a flaw in the fact that they do not expire. The solution to this is using PKI certificates, which can be simplified as being keys with expiration days built into them. After this period is over, the server will reject the certificate, requiring a new one to be made. The process of getting a new certificate is also a lot easier than with keys, as having a centralized management of access keys removes some of the complications and manual work that must be done for standard SSH keys.

The process for using traditional SSH keys was to first generate a keypair, then upload the public key to the server and storing the private key on the client machine. Whenever a SSH key would be updated, an admin would have to manually create a new keypair, remove the public key from the server, and give the client machine the new private key. This process is time consuming, often causing SSH keys to be renewed too infrequent; a big security risk, as the private key could be copied and stolen by an unauthorized user and remain useable for a long time. Therefore, using ephemeral certificates is recommended by the SSH company (SSH.com 2023, introductory paragraphs).

4 Physical security of Discord evidence collection machines

4.1 Access control

Accessing the server terminal directly should routinely only be done twice; first time for initializing the collection process, then a second, final time for stopping the collection process.

The default root user should be entirely removed, and no other single account should have full superuser permissions. The user to be used for the initialization phase should have permissions to start any programs that are to be used for the collection process, and the permissions to verify that both the programs and other required services are functioning normally. The user to be used for the ending phase should have enough permissions to stop the collection process, create forensic images, and extracting a RAM dump. Hashes are then computed for both the original drive and forensic images, to later ensure nothing in the analysis process altered any data stored on the forensic images.

The original machine could now be turned off, and the drive is to be collected and stored as evidence according to local laws, while the main forensic image would be used for analysing anything related to the case after the evidence collection process is finished.

4.1.1 The security risks of physical access to any server

Even without superuser permissions, there is a high risk involved in access to a physical server; Evil Maid Attacks being one of the main threats for this framework. Evil Maid Attacks is a group of methods that through tampering with the physical device can open it up for data leakages or unintended access methods. One likely method for tampering with collection machines is that a person with access to the physical server could open up a channel, not necessarily a new port, for remote communication or code execution with the machine.

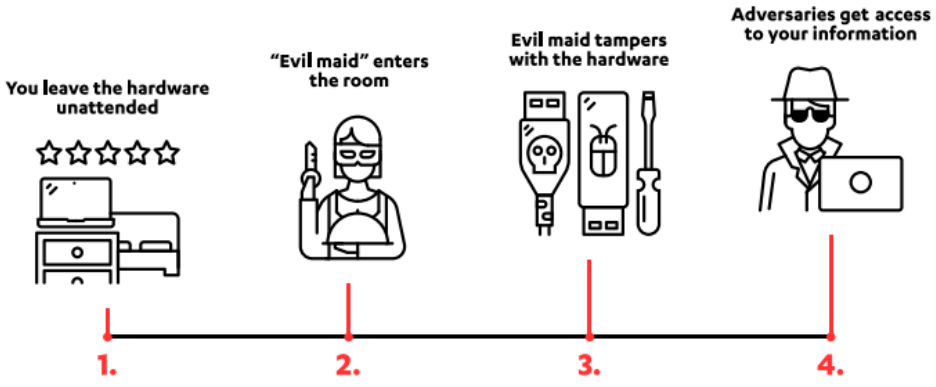

Figure 3. Evil Maid explained (F-secure 2018)

Defending against this could be done in some different ways, but one method is having witnesses that can testify that no foul play was done during physical access to the server; important points to look for is any foreign devices such as computers or memory sticks being plugged in, and for witnesses to log everything the technician did on the system, such as the different options ticked on or off during the OS installation process, any commands executed in the CLI, and what actions were done in the GUI. If all witnesses are happy with the process, one can be a lot safer that no Evil Maid Attack or similar tampering was performed during the physical access of the server. Additionally, the physical server should be inaccessible by anyone without proper authorization, and surveillance cameras should be installed.

4.2 Ensuring availability

4.2.1 Multiple machines

One of the main measures to justifiably secure availability is to have several machines collecting the same data. In the case that one were to go down, there would be another one running. There is a myriad of reasons to why a machine might momentarily shut down or lose connection to the API, but as long as the digital security of the machines is intact, some of the likeliest reasons would be connected to the machine's power supply or an ISP having problems. Due to this, it is recommended to have collection machines in two different locations with enough geographical distance that a normal power outage would not affect both machines, and if possible, use different ISPs.

One of the main prerequisites for evidence being admissible is that it should be the truth and the whole truth. A five-minute loss of connection to the Discord API could be catastrophic; a prosecutor's narrative of "We have all these deleted messages from the suspect explaining the case" could be destroyed if the defence would argue about unrecoverable evidence beneficial for the suspect that were made in a duration of downtime.

4.2.2 RAID storage and backups

RAID, or redundant array of independent disks, is a technique to store data across multiple physical disks, with the main goal of eliminating data loss for up to x amount of complete disk drive failures. There are multiple types of RAID, the most commonly used being RAID 0, 1, 5, 6 and 10, using striping (dividing data into blocks and separating them across two or more disks for higher read/write performance on hard drives), mirroring (copying data and putting it to two or more

disks), parity (bitwise operations to create parity bits for mathematical recovery), or some combination of these to achieve some level of redundancy, except for RAID 0 that only uses striping.

Even with RAID, making frequent backups is a must, as RAID does not protect against viruses, accidental data deletion or other human errors. Equally as important as taking backups, is practicing machine recoveries with them: creating a routine on how the restoration process should look like and regularly doing mock tests to ensure that there is minimal downtime between machine failure to normal operations. By doing this, any potential errors with the backup files, such as checksum mismatches or files being skipped, can be fixed before an actual situation arises.

For the evidence collection machines, a RAID level above 0 is absolutely recommended, and although it is not written in any EU regulations that it must be used, not using it might be a factor that could deem evidence inadmissible.

5 Ensuring collected evidence is court admissible

In addition to this framework being able to provide crucial information on a suspect's movements and activities so that crimes may be prevented, it should assist prosecution in court cases to prove that the surveillance and evidence collection was legally performed, and to help confirming whether the suspect is guilty of any crimes or not. Because of this, a section discussing court admissibility has been made, discussing topics of relevance to the framework.

5.1 Ensuring integrity

Integrity in the realms of computers can be defined as data being immutable; a breach in this could either be deletion or other alteration of the data in question. The Finnish witness affirmation sounds "I, N.N, pledge and assure, through my honour and conscience, that I will testify and tell all the truth in this matter without concealing or adding anything to it or changing it." (Oikeus.fi 2022). This applies to written evidence from prosecution and defence as well, even though they do not need to say an oral affirmation in court, and requires digital evidence to have well-documented integrity measures taken.

One of the most common causes of integrity breach is a threat actor getting access to the data and manages to either alter or delete all copies of it and removing any chance of forensically recovering it. This can be done either through digital, remote access to the data, or through physical access, where alteration or deletion can be anything from inserting a kill-disk into the machine holding the data in question, or physically destroying the data through critical damage to the storage hardware. Section 2, article 29 "Security of processing" in Directive (EU) 2016/680 lists several specific data security measures taken that law enforcement of Member States must comment on for the evidence to be admissible, such as 2.(i) "ensure that installed systems may, in the case of interruption, be restored ('recovery');" (Directive on the protection of natural persons with regard to the processing of personal data by competent authorities (EU) 2016/680).

If security measures to prevent any integrity breaches are not put in place, the defence can challenge the integrity of the evidence, and if prosecution does not have any additional evidence to back their evidence up, it can in the worst case be deemed inadmissible. So even though the data in theory is inaccessible from outside of an intranet, and even though the intranet itself is considered extremely safe, any evidence should still be strictly protected with proper access control and access logs, both physical and digital.

5.2 Ensuring verifiability

Verifiability is arguably not directly a criterion for evidence being court admissible, but it is extremely beneficial, as it helps backing up the integrity of evidence. Some evidence does not need verification, as its truth is common-knowledge or immutable; the current year as of writing the thesis is 2023, and our solar system is heliocentric. However, evidence that only exists in digital form is typically extremely ephemeral by essence. It is possible to alter data by simply overwriting it or preventing any chance of recovering data from a hard drive by putting it through a crusher.

In the case of this framework, verifiability of any messages or metadata can be assured without doubt by sending a preservation request directly to Discord. All that needs to be done is to compare evidence collected to what Discord has in their systems, which is no task at all with modern computers and their computing power. However, this does not cover any VoIP conversations or video sent over Discord voice channels, as Discord does not normally store these.

It might however be relevant for investigations in special cases to request Discord to do their own recordings of some investigations to later compare voice and video calls as well. Some things need to be figured out prior to this being doable; specifically, questions such as who will pay for the storage space those recordings will use, and if American law would allow for ordering Discord and similar companies to comply to these kinds of requests or orders.

5.3 Ensuring confidentiality

Directive (EU) 2016/680, article 29, 2.(b) mandates that each Member State should “prevent the unauthorised reading, copying, modification or removal of data media (‘data media control’)”, and 2.(h) demands to “prevent the unauthorised reading, copying, modification or deletion of personal data during transfers of personal data or during transportation of data media (‘transport control’)” (Directive on the protection of natural persons with regard to the processing of personal data by competent authorities (EU) 2016/680). Both these points indicate that data media should be confidential prior to the court case, thus becoming a criterion for ensuring court admissibility.

In addition to protecting the data against hacking attacks, access control should be put in place by the holders of the data so that only those with necessity to access the data may do so.

6 Introduction to Discord

Discord is a VoIP and messaging service released in 2015 by Discord Inc., located in San Fransisco, originally marketed towards video gamers. At the time of release, it mainly competed against Skype and TeamSpeak, and quickly managed to outperform both in the gaming world due to its simplicity, features, and appealing user interface. One could also argue that Discord outcompeted the market due to its modern and specialized features; Skype was in 2015 an old and outdated service targeted at businesses and small-sized group calls, and TeamSpeak required users to create and host their own VoIP servers, making it an over-complicated process for newer video game enthusiasts that simply wanted to quickly message or voice call with their friends.

6.1 Introduction to Discord as a platform

After creating a Discord-user, which is required to connect to the central Discord-servers, users are given a chance to interact with other users in two main different methods: through private one-to-one messages, or through Discord servers, also called guilds. One can also create groups that function similarly to Skype and WhatsApp groups, but what really appealed to gamers at the time of release, are the servers where people can be invited to, be given different roles and permissions to access server content, or administrating/modifying how users would interact with the server and other users.

On Discord servers, one can connect to a VoIP session through a single button press, and others can join just as easily. Discord users can almost just as easily share their screens or show camera feeds, which is another thing TeamSpeak did not offer. The Discord-hype eventually reached non-gamer communities, and it became the preferred client for many tech support groups, online Dungeons and Dragon groups, and even Microsoft eventually started providing Xbox Live users support through Discord (IGN 2018).

Discord uses HTTPS for all data exchange between user and server, but the data itself is not encrypted. This removes the at-times complicated key exchanges necessary for large group communications, which are not necessary for the purposes the app was designed for: normal communication between friends and otherwise people with the same interests.

6.2 Discord as a platform for criminality

The increasing fame and popularity of Discord caused many criminals, notably hackers, illegal pornography sellers, and hate criminals, to prefer this platform. Criminals connect to VPNs, create

burner-accounts, then communicate with clients or likeminded on Discord servers that quickly can all be deleted if suspicion arises.

6.2.1 Hackers

A simple Google-search reveals several websites compiling Discord servers with different exploits or hacking programs for sale, including many for popular video games such as Roblox, League of Legends, and Call of Duty, or Discord servers selling rootkits or different OS-targeted hacking tools.

Discord truly is a great place for hackers; most gamers or tech people, wherein many are at least slightly overlapping, have and use Discord almost daily. People prefer communicating through channels that they are accustomed to, maybe especially when they are doing something illegal. Thus, hackers have realized that one of the best platforms for marketing exploits or hacking tools is Discord, although they need to market that Discord server elsewhere, as there is no integrated server-searching function in Discord yet.

6.2.2 Illegal pornography

Discord openly allows some pornographic content on their servers: Discord only requires server owners to mark servers or channels wherein sexual content can be or is posted (Discord Guidelines 2023, under Respect Each Other, specifically point “Do not make sexually explicit content available to anyone under the age of 18”). However, Discord entirely prohibits CSAM, grooming, and otherwise showing sexually explicit or sexually suggestive content to people without their consent, as well as allowing any sexually explicit content to be available to users under the age of 18 (Discord Guidelines 2023, under Respect Each Other).

There was an enormous amount of Norwegian media attention in 2018 about girls, many underaged, who had found out that nude pictures of them were being spread around on Discord servers, often due to exes wanting revenge, so-called “revenge porn”, or people that merely wanted to make money through selling pornographic images of women or girls without their consent. Articles had headlines such as “Porno scamming and the spreading of nude pictures: - Many do not think that it will affect their jobs” (Sævold October 2018), “She discovered the pictures of Nora Mørk: «All girls I know, have sent intimate pictures of themselves»” (Bergens Tidende March 2018). These cases even reached America: “The gaming site Discord is the new front of revenge porn” (Cox January 2018).

6.2.3 Hate crime

Hate crime requires victims, which naturally leads perpetrators to places many people spend time at. Discord is a platform where strangers can get to know each other naturally and form long-lasting deep bonds, even without them ever meeting each other in real life.

One of the most notable hate crime cases in where Discord famously was used, was the May 2022 Buffalo supermarket shooting in New York, USA. The perpetrator had been planning out the crime on a closed Discord server since early March the same year and used Discord to share his findings with a small group of people prior to the attack. According to CNN (Morales et al. 2022), Discord had revealed that the perpetrator had been using the same Discord server as a diary to note down his plans and reasons for the attack.

6.3 Discord API

6.3.1 How to access

There is a myriad of different libraries that help with API communication, but the two most popular ones are discord.js for JavaScript, and discord.py for Python. In this thesis I have used discord.js and will talk about how it does things. However, almost all interactions and their limits are dictated by Discord, so the differences are not big.

To start a connection to the Discord API, one needs a bot-user that will interact with the API. A client user is not permitted to communicate with the API through scripts per Discord Terms of Services. Bots can be made by anyone through an application in the Developer portal, and upon completion, a bot token is given. This is somewhat of a master key; having access to the token allows one to fully control how the bot should work and what scripts it should run.

A bot is then added to a server and given permissions. It can then be started by activating the script, which in the case of discord.js is a Node.js program with some mandatory fields. Firstly, functionalities must be imported and piped into variables. An example is Client; this functionality allows for the bot to receive and send data to the API.

Then, Gateway Intents must be declared. These are a bot's own boundaries; if a Gateway Intent is imported, it may receive data or interact with its server in the ways defined within a Gateway Intent. If it does not have a certain Gateway Intent imported, it may not interact with anything the Gateway Intent governs. An example of a Gateway Intent is MessageContent; without it, the bot may not read the contents of a message, but if it has the GuildMessages Intent, it may receive non-content data about messages.

Lastly, the Client's login(bot_token) function is called. This effectively assigns the script to a certain bot and changes its server status from offline to online.

6.3.2 Snowflakes

Snowflakes are the IDs of Discord; every guild, user, channel, message, role, and emoji have their own unique identification number, all with built-in timestamps. The snowflakes vary in length depending on the creation of the resource; the resources created during the first months of Discord have 17 digits, then there were 18 digits until July of 2022, and any resources created now will have a Snowflake 19 digits long. The last snowflake 19 digits long has a timestamp equalling to 20th of July 2090, and if Discord still runs at that time, 20 digits will be adopted as the new length.

6.4 Previous work on the field of Discord forensics

If law enforcement has a computer and want to see all cached Discord communication, they would likely search the internet and stumble across DiscFor, a Python-based extraction tool created by Michal Motylinski (Motylinski 2022). This tool goes through local files that Discord store, organizes the collected data, and produces a report on the findings.

The main difference between this framework and DiscFor is where data is collected from; DiscFor collects data from a client machine, while Watch-me-not will gather its data from the Discord API. DiscFor is by far the easiest way to get evidence; no evidence requests need to be sent to Discord, there is no time spent waiting for responses back from them, and as I will get into in the next chapter, the Discord API sets limits onto what a normal person can do on the platform. But sadly, DiscFor has its own weaknesses; firstly, like other digital forensics tools, it requires physical access to the computer, and if full-disk encryption is in use, this needs to be removed either by the correct key or though brute-force methods. Secondly, if the owner of the computer has overwritten all Discord data and made it impossible to recover, no data can be collected with DiscFor. Lastly, a suspect might not ever have used Discord on their personal devices, instead using other devices such as an unknown friend's phone or a public library computer to connect to Discord.

Watch-me-not can in the correct circumstances overcome all these challenges, exactly because it does not rely on access to a user's device. It instead needs access to the server(s) the suspect is a member of and communicates in, and server administrator rights in case the suspect uses private, locked channels, as administrators cannot be locked out of any channels in a server.

7 About Discord evidence collection without escalated permissions

All data discussed in this chapter is in some form accessible as a normal Discord user; in other words, no special permissions or help from Discord is needed. Think of it as a normal customer that is merely using Discord to communicate with other like-minded people.

7.1 Messages

A Discord message is sent as a JSON-packet, with data about the message-related context, as in what channel it is posted in and what guild it is posted in, data about the message itself, data about the sender, and other reserved fields for different scenarios. Each message will have the exact same amount of fields and can be updated an indefinite amount of times after its creation.

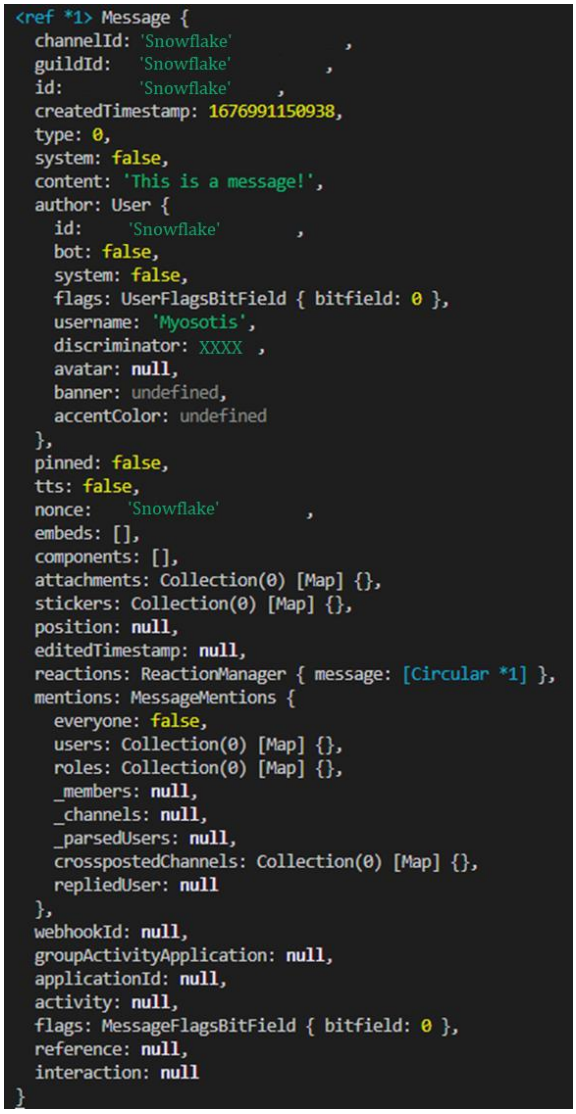

Figure 4. Discord API message

7.1.1 Edited messages

When a Discord message is edited by a user, its client will send a call to the API, which will send a *messageUpdate* event with both the old and new message.

Discord has not, to my best knowledge, confirmed to the public whether edited message versions are saved on their servers or not. However, for a normal API user fetching an edited message, they will only find the current version, and the previous. In other words, if a message has been edited twice, the original will be lost. For this reason, listening to the *messageUpdate* event and storing all versions is crucial for making sure no message versions are lost.

7.1.2 Deleted messages

A user deleting a message will result in the API sending a *messageDelete* event, which will completely wipe the message from the normal client API and from caches. Discord has confirmed that they by default delete messages but will in some cases retain deleted messages up to half a year (Discord 2022). However, prior to the message being deleted, it will be sent one last time to any clients or bots listening and can be stored offline.

7.2 Server members

There are several events related to members, such as *guildMemberAdd* when a new member joins the server*, guildMemberRemove* when a member leaves or is kicked from the server*, guildBanAdd* when a member ban is created*,* and *guildBanRemove* when a ban against a member is removed*.* These are of importance to understand the nature of the server itself; if new members routinely are swiftly removed, there might be little trust towards the members, and can support theories about criminal activity being performed on the server.

Additionally, if all suspects were to be kicked or banned from the server, it would mean that there would be a hiatus in the collection of evidence from them. However, it could also mean that admins on the server would follow the ban up with a discussion about why that was done, which could give investigators important clues.

7.2.1 Member information

Every user has some information displayed about themselves visible for everyone on the server, as for example the following screenshot of my own alternative user created for this project.

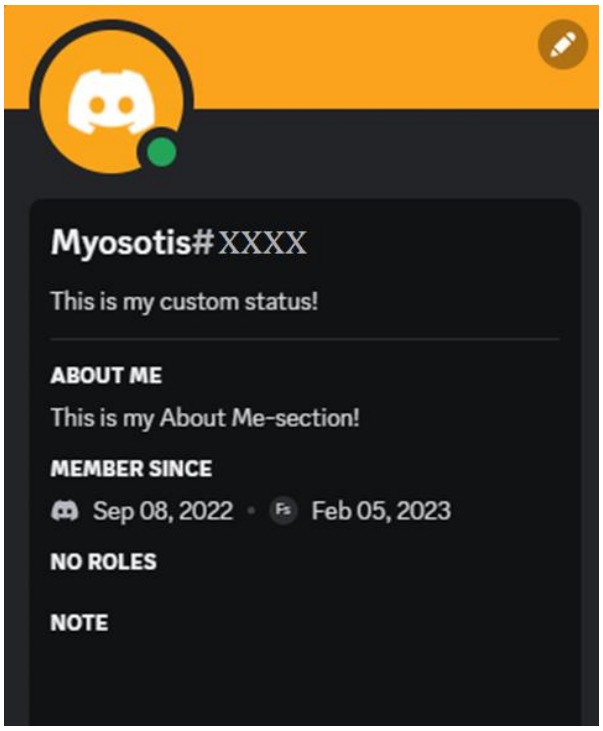

Figure 5. Member information

On the top, there is a banner that is customizable through a Discord Nitro subscription, and the circle with the Discord logo inside of it is the user's profile picture (here the default). Underneath, the username with a 4-number differentiator (censored), followed by either a custom status message, or an activity presence. Furthermore, there is an About Me-section, and a section displaying both the age of the Discord account on the left and how long it has been a member of the server on the right (if the user has left the server, the date on the right is reset). Finally, the roles assigned to the user are displayed, and a Note-section where users can keep entirely private notes about the user.

Relating to this information are two important events: *presenceUpdate and guildMemberUpdate*.

presenceUpdate is sent by the API when the activity presence of a user is updated, where the activity presence is a status of some program that the user has registered and is currently running. An example could be a video game, where the presence would read "Playing [video game name]", or if the user is listening to a song through Spotify, it would read "Listening to [song name]".

guildMemberUpdate is sent by the API whenever a user edits something else of their server information, and the fully updated GuildMember-information would be given alongside the event. An example of the event being called is when a member would change their server nickname;

here, the GuildMember JSON data would be identical to before, only with an updated *nickname* field.

An alibi could be produced for a suspect through their activity presence at a given time; with some video games, it is possible to get detailed information to what exactly the user is doing, such as if they are in a lobby or in an ongoing game. Thus, activity presences is one of the first matters prosecutors should check together with other Discord timestamps; was a suspect doing something on Discord, something that Discord caught up on and displayed to others, or were they seemingly inactive and might have done something in real life away from their device?

A relevant example to this is the case of United Sates v. Brendt A. Christensen, a lawsuit following an investigation which confirmed that Christensen had kidnapped and murdered Yingying Zhang. Christensen, under an initial interview with Police Department, told that he probably was playing a video game on his computer at the time Yingying Zhang was picked up by a car; however, it was later confirmed by digital forensics investigators that he had never played a video game that day due to logs from his computer, and this alibi was confirmed invalid (J C S 2021, 05:35 – 06:40).

7.3 Server roles

Server roles is the main method of grouping members into separate units, often to assign them server permissions, or merely to give them different descriptive tags such as “Staff”, “Client” or “Developer”.

From a forensics perspective, keeping track of available server roles and what server roles persons of interest have can be useful to see what roles they have on the server. Are they managing the server, or perhaps they have a tag that explains what they do inside of the server or in real life?

7.4 Voice calls, video calls and screen-sharing

This section of data is one of the main reasons for the creation of this framework; Discord normally never records any voice calls or video of either a screen or camera. These are also inconveniently some of the greatest treasures for prosecutions in terms of electronic evidence; anything said or done here will be a challenge for the defence to refute.

Discord, however, very much does not like scripts having access to any of these: it is impossible to autonomously record this while following Discord’s terms of service, as they are not accessible from the API, and self-scripting is prohibited by Discord, i.e. scripts performed by or through a Discord client user. This is a measure taken from Discord’s side to protect the privacy of its users,

but like many of the things that will be discussed later in this framework, this is technically speaking rather simple to do.

7.5 Timestamps

Almost everything done in Discord by a user conveniently includes a timestamp; embedded in every message snowflake, and as an extra variable together with JSON data of events. They are crucial pieces of information to take care of; not only to contextualize the order of which messages were sent and should be read back like, but also to refer to in court. Timestamps allow prosecution to say with confidence that a user sent a message while he was at home, and joined a voice call while the suspect claimed they went to the cinema.

7.6 Message attachments – media

Almost all Discord media is publicly available; when a user uploads a picture to what they believe is a private server, they are putting it on a Google Cloud-server, and when others on the server see the picture, their clients are merely looking at its URL link, downloading the media, and displaying it on their computers.

All media URLs follow the template of *cdn.discordapp.com/attachments/[CHANNEL_SNOWFLAKE]/[ATTACHMENT_ID]/[ATTACHMENT_NAME]* for media URLs,

and *media.discordapp.net/attachments/[...]* for their proxy URLs.

The media is thus available to anyone with the correct URL; however, guessing this URL is extremely impractical, as even knowing the channel ID, one must guess the correct attachment ID and the name of the uploaded media. If a script is a part of a server, it is extremely simple to retrieve any attachments; every message has a Collection including any attachments, so all that needs to be done is to check for a populated Collection in every message sent, retrieve all information from the attachment(s), and download the photo to keep a static copy of it in case a user were to later delete it from the server – attachments are also deleted from the content delivery network in a short time after a message/attachment deletion request is received.

7.7 Further challenges

7.7.1 Discord Terms of Services and API restrictions

As earlier stated, self-scripting is one of the main restrictions for this framework; as I do not have the privileges of being exempt from any points of the Discord terms of service, I cannot develop any script that for instance automatically starts a recording of a voice channel.

Another obstacle for anyone without “escalated privileges” is avoiding getting flagged or banned by the API due to too many requests. Fetching messages is one great example, where countless online forums strongly advise people to not send a message fetch request of anything larger than about 100 entities; the API is fine with responding to singular requests related to events, but due to a myriad of factors it is programmed to time-out, flag, or ban any bot user that sends too many fetch requests.

As such, the developed script available alongside the framework can fetch all members, channels, roles etc. (as these are normal requests for the API, even though many could be sent), and collect any data connected to events; but it is incapable of fetching message history prior to the bot joining the server.

8 About Discord evidence collection with escalated privileges

The term ‘escalated privileges’ is here used to discuss capabilities of data collection with Discord’s grace; a cluster of users, mainly targeted for law enforcement use, that can gather much more sensitive data than a normal bot can do even with full server administrator permissions.

8.1 Internet Protocol addresses

In these times of increased VPN usage, many might say that IP-addresses prove less useful to prosecution as the IP-address they receive might merely be one owned by one of the bigger VPN providers. However, if a suspect is not using VPN, or mistakenly send some packets with their VPN turned off, the geo-locating features of an IPs can result in a swift arrestation where this is deemed justifiable and necessary.

8.2 Email-addresses

Signing up to Discord requires an email-address, and while many criminals that come to Discord to do unlawful activities might use a burner email, others might use their personal email on their account, or perhaps someone in their household is using their personal email. Nevertheless, having access to a suspect’s email-address will allows for new ways to learn more about a suspect, or law enforcement could even be able to find a leaked password connected to that email.

8.3 Access to restricted channels

Bots often get server administrator permissions when added to a server, and in a lot of cases, people will not be too concerned as they might think that the bot would not function correctly if it would not receive all permissions it requested. However, a normal user asking for administrator permissions is to most a lot more suspicious; thus, getting a normal user to have them is often harder than tricking an administrator to invite a evidence collection bot with some other side features such as a music-player (these are very popular!)

With escalated privileges however, one does not need to receive it from another server administrator. For evidence collection purposes, the most important feature of being an administrator is that all restricted channels become visible. Sensitive discussions that a suspect would not want to risk leaking out would surely be performed in a restricted channel where they feel that their words are secret.

8.4 API limits removed

With API limits removed, fetching every single message on the server becomes a rudimentary task that the API will not blink at. It truly is an enormous perk; it is of course possible to code a janky script that scrolls up in every channel and copies all messages along its way, but this becomes a lot more resource heavy, as a script must now find the timestamp connected to the message, it must make requests for any reactions, and so on.

8.5 Invisibility for any bot- and client users

This last part is more of a wish from the author of this framework to Discord, as Discord has not (understandably) publicly told whether they do this or not, and if no law enforcement agencies have requested this from Discord before, it could be a lengthy discussion whether this should be allowed by Discord or not.

Having invisible users shifts a Discord server investigation from a strategic game of minesweeper where one must be careful to not seem suspicious while gathering as much evidence as possible, to a puzzle game where law enforcement is able to merely collect puzzle pieces as an invisible observer that no suspect could confirm being there.

This is practically speaking no challenge at all; Discord simply does not send any data about this invisible user to anybody, but silently passes along all data that is passed through the server to it. However, whether Discord would be willing to implement this for law enforcement or not is a whole other matter.

8.6 Recording voice and video

With Discord’s permission, self-botting is now possible, and scripts can be coded for clients to automatically enter calls and start recordings when user activity is discovered. Doing this from a client is likely the easiest method, opening a new Discord client for each user and recording voice and video for defined users.

9 Setting up the framework code

This section is written as a step-by-step tutorial to reduce the technical barriers for the code belonging to the framework, not including the securing of the evidence collection machine(s).

9.1 Tailoring the code

If variables not already gathered by the code would be relevant to an investigation, it is simple to add code for this in the watch-me-not.js file. As an example, if detecting and recording whenever a server role would be deleted would be wanted, this could be accomplished by looking through the Discord.js documentation to find the corresponding event *roleDelete*. The added event listener would look like this:

```
client.on("roleDelete", (role) => {
    // ...
});
```

Figure 6. New event listener.

There are also plenty of possibilities for fetching data on startup of the script. New startup code should be added under the *ready* event listener.

9.2 Setting up environment variables

Watch-me-not uses environment variables for all variables that vary from implementation to implementation. An example of this is the bot token, which is obtainable from Discord's bot application websites.

```
.env
1   TOKEN=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Figure 7. Environment variable file.

It is stored inside of the code's env-file, and used by first importing 'dotenv' and then referring to the variable by referring to process.env. and specifying one of its user defined variables.

```
require('dotenv').config();
```

```
client.login(process.env.TOKEN);
```

Figure 8. Using environment variables.

9.3 Setting up the database

For Watch-me-not's demo code, a MySQL Workbench Model file is included. To use this setup for a database with MySQL Workbench, open up the program and click File -> Open Model, and select the .mwb file. With the model now open, changes can be done to further customize it, or simply continuing to export it. To export the model, enter either the MYSQL Model or EER Diagram pane, click File -> Export -> Forward Engineer SQL CREATE Script. Customize the SQL options as required, click next, ensure that the Export MySQL Table Objects-option is ticked, and click Next. Finally, take the generated script and use this to create the tables in a new and clean database.

10 Further development possible on the framework

10.1 Tailoring to suit different purposes

The amount of data one can expect to download from a busy 20+ person Discord server in a month's time is astonishing, and law enforcement or national intelligence agencies often have restrictions on what data they are permitted to collect; thus, tailoring of this framework to remove or add features is a given.

Luckily, the Discord bot API is extraordinarily well documented, and the syntax of the Discord.js library is equally consistent. Therefore, any person with some previous coding experience should not find a great challenge if they wish to collect some data from events that are not covered in this framework.

10.2 Without escalated privileges

10.2.1 Future changes to Discord's API / Discord Bot Program

Watch-me-not without escalated privileges is dependent on Discord's API to not restrict any of its key features; one specific worry is a specific ban on bots whose main purpose is to log server information. Watch-me-not does currently not have any function other than staying in the Discord server, listening to events, and noting these down. However, bots were mainly created to provide additional functionality to servers, such as bots that can play music in voice channels, bots that implement support ticket systems, or bots that can assist in chat moderation such as deleting messages with offensive words. Watch-me-not can be programmed to have secondary functionalities, and pretty much requires it, as no servers would have a bot that seemingly does nothing.

But even having secondary functionalities will likely not be enough to fool Discord's systems if it wishes to shut down logging bots, especially not with the surge of new AIs we are seeing these days. Trying to avoid this detection system would nevertheless be a breach of Discord's terms of services, something that will make any evidence collected illegitimate and non-submissible to court. Thus, this simple change from Discord would mean the end of Watch-me-not without Discord's blessing. This choice might also not be made on Discord's own initiative; governments are getting more restrictive on the matter of storage and usage of personal data, first with the General Data Protection Regulation in 2018, and now recently, ChatGPT was banned in Italy due to privacy concerns raised by the Italian data-protection authority Garante per la Protezione dei Dati Personali (McCallum 2023). Discord would likely choose to comply and remove requested

features if the whole of EU or close to it would choose so, as they would lose too much of their customer base.

Discord letting server administrators have insight to bots' code is another worrying matter. Until now at least, Discord has allowed people to keep bot code secret, so they could not be pirated by other users. However, again with the surge of heightened focus and awareness on personal data handling, server owners might receive the right to go over what exactly the bot does, perhaps without seeing the whole code. This could be things like seeing what libraries the bot is using, what events it listens for, and whether any server communication is stored anywhere. One example of how this could be done, is that any libraries allowed to communicate with the Discord API are required to do these analyses on the source code of bots and send this to Discord on start-up.

This, of course, is not desirable for this framework as server owners would have full insight on how any bot works; in the case of Watch-me-not, they would be able to see that its function is to listen to events, format data, then send it to a MySQL server. This would ring alarm bells for most server administrators, and swiftly remove the bot in addition to telling the other server members about a potential data leak.

10.3 With escalated privileges

10.3.1 New American laws / law interpretations

One of the most important clauses of the EU e-evidence regulations and directives on how it should be used is written out in point 36 of the earlier discussed note 5448/23 sent from the Presidency to the Permanent Representatives Committee, the analysis of the final compromise text on the Regulation on Production and Preservation orders for electronic evidence in criminal proceedings: "It should only be issued if it could have been ordered under the same conditions in a similar domestic case and where it is necessary, proportionate, adequate and applicable to the case in hand." (Regulation on Production and Preservation orders for electronic evidence in criminal proceedings).

This in other words means that any use of EU e-evidence requires the courts of law that rule in the country of the service provider to agree with the courts of law in the country of an investigation that a production or preservation order is appropriate for the case on hand. In the case of Discord, no matter if one would be able to use the EU e-evidence channels to skip court processing in USA, the US laws and court(s) that preside over Discord Inc. must rule similarly to the EU Member State, making this framework and its capabilities rather influenced by American laws and their interpretations.

10.3.2 New local laws / law interpretations

Just as in 9.3.1, if a country suddenly would forbid any state-driven intelligence, an extreme example used for simplification, it would prevent them from using this framework even though USA would allow it. However, there are some cases where a national intelligence agency / police department might not have the legal authority to conduct investigations, but there still being a possibility for the case to be investigated by another country. One scenario is where there is reasonable evidence to suggest that a different country's national is using Discord for criminal activities on the same Discord server. Here, that country's investigative branch(es) may choose to open an investigation and might in some cases not only be able to assimilate enough evidence to not only incriminate their country's nationals, but also collect and, if police collaboration between the two countries allows for this, send incriminating evidence about the initial country's national(s), and allow this evidence to be used to sentence them.

Of course, cases that align perfectly like this are few and far between, but trying to find alternative solutions is a lot better than the fallback solution that Discord would need to deploy to satisfy American laws and their own Terms of Services, which sadly would be to remove any incriminating content and banning any Discord users found violating Terms of Services.

10.3.3 New Discord escalated privilege possibilities

An example of new information that can be extremely potent evidence that was added quite a long time after Discord's initial release is billing information, that was added as an optional data section due to the 2017 release of Discord Nitro, a paid subscription to get certain benefits over a standard free account. In the recent case of several leaked U.S. military classified documents being posted on a Discord server, the prime suspect by the time of writing had used their real name and home address for their billing information and was swiftly identified and arrested because of this (Bertrand 2023).

Thus, as Discord adds new data sections, either optional or mandatory, more information might be available to investigations. Updating one's knowledge of what evidence might be present could massively change the required duration of an investigation, as likely was the case just mentioned.

11 Conclusion

11.1 Outcomes

The product created is what was possible as a student that had no assistance from Discord, meaning that there were certain challenges that could not be overcome, such as the impossibility of showing an implementation of voice / video recordings and in general having to abide by the Discord Terms of Services, and there were plenty of times where prediction was the only way to comment anything on a matter.

Watch-me-not has not been coded to do everything that is feasible; however, this was never the goal of the thesis. Watch-me-not was created with the goal of being easy to understand, easy to expand, and easy to deploy, something I believe it fulfils, relatively speaking. Another goal was to keep this as relevant as possible to as many countries and their laws and regulations as possible: I decided to make the scope smaller and focus on the European Union and its member states, and with their quite restrictive but necessary regulations, I do believe most other nations will agree with measures that need to be taken for evidence admissibility.

11.2 The project as a thesis

The project had two main purposes: it was created to fill a general necessity and lack of solutions in Discord intelligence, but also to teach myself, the author, more about topics related to the thesis. These two goals might eventually lead to conflicts of interest, as the project itself might benefit more from longer discussions about certain themes that mostly or partly do not align under the study programme the thesis is written for. This can often lead to either a thesis that does not talk about all matters that are necessary for the project to fully succeed, or a project that is not focused enough on the field the thesis is being written for.

I found it challenging to balance this, as there is a fine balance between two much law, and too little law. In most software development-theses, more than just a little law is too much law, but as this thesis was related to data security and cyber security matters, I had both a leeway and a real reason to talk more in depth about regulations and directives that would affect the final product, as data security is tightly linked up to, and is a concept because of, real-world laws that must be followed.

As for whether I would write the paper differently if I were not creating it as a data security thesis: I am going to say probably not. That is partly because I do not have any formal education in law, so doing more in-depth research and reporting on EU regulations and directives is difficult for me and

out of my field, but also because I do not think it is necessary to talk more about EU regulations. The next step would probably be to write about the specific needs to get authority to start intelligence work, and this requires focus on one singular country and their laws and legal procedures, something I do not think would fit in this thesis.

11.3 Self-reflection and personal growth

In terms of matters I would have retrospectively done different in the thesis, would be to earlier accept that I cannot make the product completely “whole”. I wanted to have every feature foreseeably necessary by anyone created and availably be ready for anyone to use, but this ultimately ended up in me pretty much entirely redoing my schedule for the project. Not only would it realistically speaking be impossible to do in the number of hours reserved for the thesis, but it would likely lead to the demise of the project post-publication, as many features law enforcement and intelligence agencies would want is normally against Discord Terms of Services.

I have had several courses in university about data security, but they were limited in terms of speaking about law aspects. Being able to research this in depth here allowed me to link theory up to regulations, something that concretizes and makes my studies more relevant. Doing research on securing computers both physically and digitally also taught me a lot, and I am confident that I will have good use for the knowledge in the future.

Lastly, starting work on the thesis earlier than perhaps many, gave me more days to idlily think about the direction I wanted to go with the thesis and what I should and should not write about. I started work on my thesis in September of 2022, and finished it in May 2023, and going back, I would not reserve less time for the project. Working gradually on a project like this also taught me how less daunting it is to have leeway to have some days off, and then write more on other days.

11.4 Last words

The internet has ever since its public release on the 30th of April 1993 been a place for innovation, sharing, and friendships, but also a helpful and lucrative place for people with bad intentions. The same communication channels that allow families and friends to video call together from distances, allows criminals to plan and discuss crimes. The same web stores that allow people to pay for products and services from the comfort of their home, quickly became targets for hackers that want to earn big and easy money by stealing credit card information.

Crimes committed and evidence left through Discord might eventually drastically drop in numbers due to future changes in Discord’s policies or systems, or due to people simply finding new and

better service providers to fit their needs. Until that happens however, Watch-me-not and similar bots will be available tools to assist in fighting crime by acquiring evidence from Discord, and ultimately, be attempting to make the internet and the world to a little safer place for everyone.

References

Awan, I. 2017. Cyber-Extremism: Isis and the Power of Social Media. Soc 54, 138–149. https://doi.org/10.1007/s12115-017-0114-0. Accessed: 08 March 2023.

Bergens Tidende March 2022. Hun oppdaget bildene av Nora Mørk - «Alle jenter jeg kjenner har delt intime bilder av seg selv». URL: https://www.bt.no/amagasinet/i/p6Vz7E/mia-landsem-alle-jenter-jeg-kjenner-har-sendt-intime-bilder-av-seg-selv. Accessed: 26 February 2023.

Bertrand, N. 2023. Teixeira used his real home address in billing info on social platform Discord, court documents say. CNN. URL: https://edition.cnn.com/politics/live-news/pentagon-leak-jack-teixeira-court-04-14-23/index.html. Accessed: 16 April 2023.

Morales, M., Levenson, E., Beech, S. May 2022. 15 people joined suspected Buffalo shooter's Discord private chat shortly before the shooting. CNN. URL: https://edition.cnn.com/2022/05/19/us/buffalo-supermarket-shooting-court/index.html. Accessed: 26 February 2023.

Cox, J. January 2018. The Gaming Site Discord Is the New Front of Revenge Porn. The Daily Beast. URL: https://www.thedailybeast.com/the-gaming-site-discord-is-the-new-front-of-revenge-porn. Accessed: 26 February 2023.

Directive on the protection of natural persons with regard to the processing of personal data by competent authorities 2016/680. Directive (EU) 2016/680 on the protection of natural persons with regard to the processing of personal data by competent authorities for the purposes of the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, and on the free movement of such data, and repealing Council Framework Decision 2008/977/JHA.

Discord 2022. How long Discord keeps your information. URL: https://support.discord.com/hc/en-us/articles/5431812448791-How-long-Discord-keeps-your-information. Accessed: 21 February 2023.

Discord Guidelines 2023. Discord Community Guidelines. URL: https://discord.com/guidelines. Accessed: 26 February 2023.

European Commission 2015. Communication from the Commission to The European Parliament, The Council, The European Economic and Social Committee and The Committee of The Regions. The European Agenda on Security. Strasbourg. URL: https://home-

affairs.ec.europa.eu/system/files/2020-09/eu_agenda_on_security_en_0.pdf. Accessed: 26 February 2023.

European Council 2023. Electronic evidence: Council confirms agreement with the European Parliament on new rules to improve cross-border access to e-evidence [Press release]. URL: https://www.consilium.europa.eu/en/press/press-releases/2023/01/25/electronic-evidence-council-confirms-agreement-with-the-european-parliament-on-new-rules-to-improve-cross-border-access-to-e-evidence/. Accessed: 26 February 2023.

F-secure 2023. Evil Maid Attacken – wenn die Putzfrau den PC hackt. URL: https://blog.f-secure.com/de/evil-maid-attacken-wenn-die-putzfrau-den-pc-hackt/. Accessed: 2 April 2023.

IGN 2018. Microsoft Bringing Discord Support to Xbox Live. URL: https://nordic.ign.com/company/13908/news/microsoft-bringing-discord-support-to-xbox-live. Accessed: 26 February 2023.

Irshaid, F. 2014. How Isis is spreading its message online. BBC News. URL: https://www.bbc.com/news/world-middle-east-27912569. Accessed: 08 March 2023.

J C S 2021. The Case of Brendt Christensen. URL: https://www.youtube.com/watch?v=rudSWhe_KD0. Accessed: 22 April 2023.

Landau, S. 2010. Surveillance or Security? The Risks Posed by New Wiretapping Technologies, pp. 10. The MIT Press, Cambridge, Massachusetts & London, England.

McCallum, S. 2023. ChatGPT banned in Italy over privacy concerns. BBC News. URL: https://www.bbc.com/news/technology-65139406. Accessed: 07 April 2023.

Microsoft 2022. Best practices for Azure RBAC. URL: https://learn.microsoft.com/en-us/azure/role-based-access-control/best-practices. Accessed: 02 April 2023.

Migration and Home Affairs 2023. e-evidence. URL: https://home-affairs.ec.europa.eu/policies/internal-security/cybercrime/e-evidence_en. Accessed: 22 April 2023.

Motylinski 2022. DiscFor-Discord-Artefact-Extraction-Tool. URL: https://github.com/MichalMotylinski/DiscFor-Discord-Artefact-Extraction-Tool. Accessed: 07 April 2023.

Oikeus.fi 2022. As a witness in a trial. URL: https://oikeus.fi/tuomioistuimet/en/index/asiointijajulkisuus/kutsuttunaoikeudenkayntiin/asawitnessinatrial.html. Accessed: 15 March 2023.

Regulation on Production and Preservation Orders for electronic evidence in criminal proceedings. Regulation of the European Parliament and of the Council on European Production and Preservation Orders for electronic evidence in criminal proceedings and for the execution of custodial sentences following criminal proceedings 5448/23, 20 January 2023, pp. 108. URL: https://data.consilium.europa.eu/doc/document/ST-5448-2023-INIT/en/pdf. Accessed: 26 February 2023.

SSH.com 2023. Ephemeral Certificates and Ephemeral Access. URL: https://www.ssh.com/academy/iam/ephemeral-certificates-and-access. Accessed: 02 April 2023.

Sævold, H. October 2018. Pornosvindel og spredning av nakenbilder: - Mange tenker nok ikke på at det kan gå ut over jobben. Digi. URL: https://www.digi.no/artikler/pentester-mia-landsem-om-pornosvindel-og-spredning-av-nakenbilder-mange-tenker-nok-ikke-pa-at-det-kan-ga-ut-over-jobben/500659. Accessed: 26 February 2023.

Watch-me-not code 2023. URL: https://github.com/CapoFill/Watch-me-not.