# PHASE 1: PROBLEM IDENTIFICATION

## EXFILTRATION USING DISCORD

Authors:

Ana Vidal 118408

Simão Andrade 118345

# Problem

Discord is a popular **web-based** communication platform used in non-corporate and corporate networks (e.g. Software Development companies)

Being a **legit application** that run in very **usual network ports**, it can be used to **exfiltrate** data

# Real Life Cases

Examples of data exfiltration using Discord:

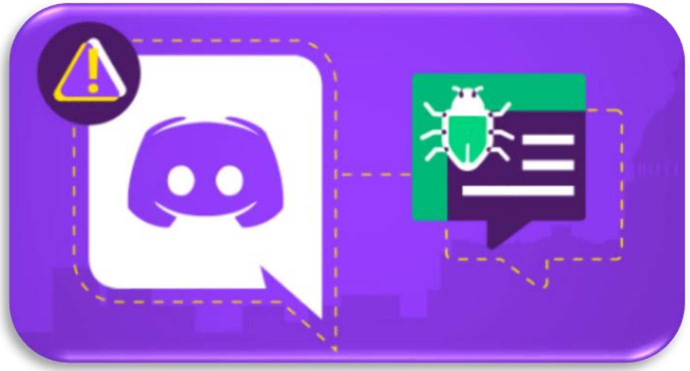
- The Hacker News - NS Stealer Uses Discord Bots to Exfiltrate Data (2024)
- Intel471 - How Discord is Abused for Cybercrime (2024)

# Why is it difficult to solve?

Discord uses HTTP/HTTPS to send data, therefore the data uses encryption during the communication

- This difficults the analysis for systems that perform DPI (Deep Packet Inspection) or based only on rules/policies

Although Discord made some updates regarding security (link), malicious users still take advantage of tools that allow development of plugins



07/11/2024

# Discovery

- It has a built-in function that enables automated messages sent to a text channel in the server (Webhooks)
- Allows the upload of a variety file types (e.g. PNG, PDF, MP4)
- The maximum file upload is 10MB

# Data Filtering

- IP Network: 162.159.0.0/16 (nslookup.io)
- Uses WebSockets over **TCP** for real-time communication
  - Destination Port(s): TCP/80 and TCP/443
  - Source Port(s): UDP/50000-65535
- Voice/Video communication and background syncronization is done using **QUIC**

# Data Agreggation

To perform the analysis, the following data will be extracted:

- **Group and Private Conversations** – the conversation type is obtained at the packet level (uploads/downloads)
- **Daily and Weekly message flow with various formats of files** – analyzing the timestamps of interactions (uploads/downloads)

# Data Collection In Testing Context

Tools of network analysis:

- Wireshark:
- TCPDump

Proxy tools for traffic capturing:

- Burp Suite

# Data Collection In Real Context

Tools to obtain data from devices, server, aplications, etc.:

- Syslog
- Agents

# Qualitative Data - Packet Level

- IP Source
- IP Destination
- Used Protocol
- Packet Length
- Timestamp (in seconds)

# Qualitative Data – Flow Level

- IP Source
- IP Destination
- Size of Exchanged Data
- Data Flow Start/End Timestamp (in seconds)
- IP Protocol Number

# Data Sampling – Sampling Interval

- In order to convert our qualitative data into quantitive data, we chosed to use observation windows of **0.1 seconds** and **1 second**
- This allows a balance between the level of detail needed to capture relevant events and the volume of data generated

# Data Sampling – Packet Level (1/2)

- **Mean and Median of Packet Length:** Attackers may try to avoid detection by sending data in irregularly sized packets.
- **Frequency of Packets per Millisecond (overall and per IP Source):** It’s an indicative of data being quickly transferred out of the network.
- **Number of Packets per Time Interval:** Can reveal irregular traffic patterns typical of exfiltration attempts.

# Data Sampling – Flow Level (2/2)

- **Mean and Standard Deviation of idle times:** Unusual gaps or consistency between flows.
- **Number of flows:** Indicating irregular usage patterns.
- **Size of exchanged data (Mean/Variance):** Changes in data size can point to unexpected or secretive data transfers.
  - Up/Down
- **Durations of Flows**

# Data Production

It will be done using **tree types** of bots:

- **Easy to Detect:**
  - **Size:** 10MB
  - **Frequency:** Periodically (40s)
- **Intermediate to Detect:**
  - **Size:** 1-10MB
  - **Frequency:** Same variance as a normal behavior
- **Hard (almost impossible) to Detect:** Through embedded images, using Discord CDN

Malicious Behavior

It will be done by performing **normal usage** of the application, made by:

- **Humans:** sending messages and files as usual
- **Bots:** made by plugins added to the server

Normal Behavior

# QUESTIONS?