The analyzed game modification, available at TerjeBruoygard/TerjeMods, follows the development model imposed by Bohemia Interactive and uses the Enforce Script language, provided by the maintainer of the game engine. The mod aims to allow members of the same group to see each other in the game interface and on the map through the display of visual markers.

## Context and Operation

The analyzed modification is the **Terje Party Mod** for the game DayZ, whose purpose is to allow the formation of groups (*parties*) among players within the game environment. The observed mechanism follows this flow:

- Player **A** interacts (key **F**) with Player **B**, sending a party invitation;
- Player **B** interacts (key **F**) with Player **A**, accepting the invitation;
- The party is created and the players begin to see each other through markers on the interface and on the map.

## Observation of Anomalous Behavior

During empirical testing, an inconsistent behavior was identified after a **complete computer format**, with removal of all local folders and data. In the observed situation:

- The analyzed player **no longer saw the party**, as if not being in a group;
- Another player still **saw the first player as a party member**;
- There was no explicit server-side action removing the group.

This asymmetry led to the initial hypothesis that **party information is not fully stored or validated on the server**, but depends on data persisted locally on the client.

## Technical Investigation

### File Monitoring

The DayZ process was monitored during party creation using system observation tools (Microsoft Task Manager / Resource Monitor / Process Explorer), in order to identify access to relevant local files.

As a result, recurring access to the following file was identified:

`TerjeParty.dat`

This file was opened and modified at the moment of party creation or update.

### File Analysis

The `TerjeParty.dat` file was analyzed using a **hex editor**, and a consistent binary structure was observed after party formation, indicating local configuration data storage.

#### File Structure

The file format follows a deterministic layout:

- **Initial header (4 bytes, int32, little-endian):**
  - Represents the total number of GUIDs stored in the file.
- **Sequential entries per GUID:**
  - 4 bytes (int32, little-endian) indicating the size **N** of the GUID string;
  - **N bytes** containing the GUID in **UTF-8** format, without any additional terminator.

Each GUID represents a player identifier and is stored as a **Base64 (URL-safe) string**, not as a fixed binary value.

The observed write behavior consists of simply **appending new entries to the end of the file**, requiring only the initial GUID counter to be updated to maintain format consistency.

### Observable Properties

- The file layout is **simple, sequential, and deterministic**, facilitating manual parsing;
- The structure based on `[count → size → string]` allows **direct insertions and modifications**, as long as the header is adjusted;
- Data is stored locally in readable format (Base64), **without encryption or integrity validation** embedded in the file itself.

These characteristics explain both the disappearance of parties after deletion of local data and the possibility of manually recreating them.

## Structure Exploration

During analysis, it was possible to identify within `TerjeParty.dat`:

- The total number of stored IDs;
- The complete list of GUIDs corresponding to party members.

It was identified that the GUID used by DayZ is derived from the player's **SteamID** through the following process:

`SteamID → SHA-256 → Base64 (URL-safe) → GUID`

Observed example:

- SteamID: `76561198012345678`
- SHA-256: `c991f9c7f37b12d7b3deddeb607ab338af88f4cbd253171c02cce3b...`
- GUID (Base64 URL-safe): `yZH5x_N7Etez3t3rYHqzOK-I9MvSUxccAszjtPnnRlc`

### Exploit / Proof of Concept

Based on the full understanding of the `TerjeParty.dat` file structure, it was possible to:

- Manually write GUID entries into the file;
- Create **custom parties locally** without legitimate in-game interaction;
- Insert players into a party **using only knowledge of their SteamIDs**.

After manually modifying the file, the game recognized the inserted players as valid party members, demonstrating that **there is no robust server-side validation** regarding group composition.

The proof of concept and experiments performed are documented at:

https://github.com/PSalleSDev/TerjeExp

## Editor's Note

This document was prepared with a **strictly technical, academic, and impartial character**, with the sole purpose of **analyzing and documenting observable behaviors** within the systems that compose the evaluated modification. The production of this material **has no commercial intent**, nor does it aim to improperly expose, discredit, or cause harm to the initiative or its maintainers.

The central motivation of this work is to **contribute to the strengthening of the mod**, supporting its continuous evolution and the maintenance of a more **secure, reliable, and stable** environment for all participants.

The disclosure of the observations and analyses presented here occurs in the context of **technical knowledge sharing**, with the intention of assisting the mod's maintainers in identifying areas for improvement, mitigating risks, and enhancing the practices adopted. It is based on the principle that technical transparency and constructive dialogue are fundamental tools for the sustainability and maturity of community-driven and collaborative projects.

This document should be interpreted as a **positive and collaborative contribution**, aimed at promoting sound security practices and delivering a quality gameplay experience to the community, grounded in **mutual trust** between maintainers and players.