

Общая структура и схема работы Universal Anti-Flood Script

User's script

Скрипт регистрируется на UAFS, получает свой ufh и далее работает с ним, передавая его как аргумент функции проверки на флуд.
Если при проверке выясняется, что лимит событий исчерпан для данного промежутка времени, функция проверки возвращает единицу и вызывает callback функцию (если она была указана).

Функции: (типа интерфейс)

- 1) callback function { ufh, nick, uhost, chan }

UAF script

Функции: (типа интерфейс)

Параметры в квадратных скобках являются необязательными.

- 1) регистрация пользовательского скрипта – предоставление ему уникального идентификатора (handle) – позволит из одного скрипта использовать разные настройки (параметры: floodsettings, [hostmask type], [ignore flags] , [callback_function], [strict mode], возвращаемое значение – unique flood handle (ufh))
- 2) Добавление события в список и проверка на флуд с вызовом callback функции (параметры: ufh, nick, uhost, [chan], [ignore flags]) callback вызывается с этими параметрами, чтобы можно было определить кого игнорить/банить и т.д.), либо по коду возврата – 1 – флуд обнаружен, 0 – нет.

Для чего нужен UFH (Unique flood handle) ?

(лучшего перевода слова “handle”, чем «манипулятор» я так и не смог найти)

Прежде всего для проверки на флуд из нескольких скриптов/триггеров – UFH служит идентификатором настроек (так же как user handle служит для идентификатором пользователя для eggdrop'a) для проверки на флуд – строк(событий) в секунду, callback функции, правила сравнения хостов (fullmask, [nick!*@host](#), [*!ident@host](#), [*!*@host](#)), флаги игнорирования user'a и режим проверки на флуд – позволяет задать настройки один раз для всего времени использования handle.

Подробнее читать в readme !