Статья ULF/УЛЬФ - Унифицированый Формат Логов | Unified Log Format

xss.is/threads/70738

Привет всем! 

Наконец то руки дошли до **Ульф**а.

Немного предыстории.

В своё время писал простенький парсер для логов, а так же Приват/Паблик чекер.
Во время написания столкнулся с проблемой - нехватка единого стандарта для формата логов!
Тоесть каждый автор стиллера пилит свой формат, свой велосипед который естественно лучше других ~~(конечно же смена названия пары файлов то что нужно).~~
Соответственно для обработки разных форматов приходиться писать дополнительный код. Естественно его не много, всего лишь сменить имя файла, и разобраться с внутренним форматом. Но если подумать более глобально то можно выделить несколько проблем:

1. Первая и банальная - лишний код для обработки лога.
2. Вторая и основная - отсутствие единого формата, что не позволяет создать набор универсальных утилит или серисов для работы с логами.

Если в первом случае оверхед по количеству кода не большой, то вот со второй проблемой не всё так просто.
Для того что бы содать удобную маштабируемую инфраструктуру по типу парсера, различных чекеров, панелей, стилеров, нужно разрабатывать всё с нуля.
Соответственно возрастает стоимость и время разработки хороших продуктов.
Конечно для кодеров это плюс - больше работы больше денег, но для индустрии в целом это **стопор** *~~(есть такое слово?)~~* развития.

Что я предагаю.

Моя идея проста - создать **единый** стандарт для формата логов.
Возможно это не кажеться столь важной проблемой, но я считаю что это может стать **основой** для дальнейшей унификации отрасли)
Единый формат, единые утилиты для обработки, единая панелька для стилеров (по заветам Quake3 , и другого юзера ник которого не помню, но видел где-то в темке), единый формат базы данных для хранения логов и тд.
Унификация позволит создать набор универсальных инструментов которые в дальнейшем может использовать каждый.

Возможно это ударит по заработку кодеров, но как по мне это к лучшему - исчезнет необходимость по сто раз писать один и тот же код, панель упаковщик (тот что данные в архив пихает и на серв отправляет) на стороне стиллера, софт для отработки и тд. Кодеры смогут сосредоточиться на более интересных вещах, **двигать** нашу отрасль вперёд не отвлекась на банальную рутину.
А совместная опенсорс разработка позволит создать качественные инструменты(~~не придёться потом ныть о криворукости очередного C# кодера~~). Инструменты в Unix идеологии, маленькие тулзы для каждой задачи, написанные единожды но используемые на протяжении десятилетий.

Теперь немного о формате.

В качестве основы я выбрал **Json**, тк это удобно, да и большую часть потребностей он закрывает, плюс *нативная* интеграция с Document based ДБ.
Проанализировал различные логи, можно выделить основной набор данных которые мы можем извлечь из логов:
1. Данные с браузеров:
- Логин/Пароль/Юрл
- Кукисы
- Сохраненные СС
- Автозаполнение.

2. Системная информация:
- Айди, Айпи, Гео, Дата и тд.
- Железо юзера
- Список процессов
- Список установленного софта

3. Крипта:
- Файлы веб расширений(Metamask, Binance, Brave и др.)
- Wallet.dat файлы (Electrum)
- Файлы других десктоп кошелей (Exodus, Atomic)

4. Файлы других приложений:
- Steam
- Discord
- FTP

5. Файлы с файл-граббера.
На основе этого списка нам нужно построить универсальный формат.

Если на счёт данных с браузера и системной инфой проблем нет, то вот с криптой, файлграббером и другими приложениями есть вопросы.
[1]*Как выделить универсальные правила для их описания и хранения?*
Собственно наброски формата:

JSON:

```
{
    "id": "str",//Часть данных вывел в рут для более удобной сортировки
    "date": "str",
    "geo": "str",
    "screenshot": "base64_str",
    "browser_data": {
        "login_data": [
            {
                "login": "str",
                "password": "str",
                "url"
            },
        ],
        "cookies": [ //кукисы в формате Netscape
            {
                "domain": "str",
                "sub_domains": "boolean",
                "path": "str",
                "https_only": "boolean",
                "expires_at": "str",
                "name": "str",
                "value": "str"
            },
        ],
        "cc": [
            {
                "holder": "str",
                "type": "str",
                "number": "str",
                "expire": "str"
            },
        ],
        "autofills": [
            {
                "name": "str",
                "value": "str"
            },
        ],
    }
}
```

Подобный формат лога намного удобнее для автоматической обработки нежели построчные записи в стандартных лог файлах.

Немного проблем и вопросов.

Одна из проблем сейчас это формат для приложений и крипто файлов.
Тк они зачастую довольно разные и вывести единый формат довольно сложно.
Если взять к примеру Дискорд, то тут всё просто:

```
"discord": ["token1", "token2"],
```

Массив с токенами

Со Стимом сложнее, так как мы имеем набор с десятка файлов.
Из идей можно в Json занести информацию и список файлов, а сами файлы либо закодировать в Base64, либо хранить на диске, с путями в Json файле. Набор мета инфы позволит не считывая файлы с диска делать какой либо анализ.
В плане крипты есть идеи извлекать нужные данные с файлов и хранить в Json файле:

- Метамаск - хэш и соль для дальнейшего брута,
- Wallet.dat - прив кеи, сиды и адреса для дальнейшего вывода либо чека баланса.

Подобная предобработка позволит упростить дальнейшую работу с файлами кошеля и сэкономит время как юзера так и кодера пишущего софт для данного случая.
[2] Но вопрос остаётся открытым и выноситься на решения нашему комьюнити)

Несколько версий стандарта

Для различных целей могут понадобиться различные спецификации формата, на данный момент на ум приходят Три:

1. **ULF** - стандартный полно-имённый формат(названия полей в полном виде, основа формата Json объекты)
2. **ULFS** - simplified, имена сокращены либо полностью отсутствуют - меньше размер, но хуже читаемость.(без названий полей, основа - Json массивы), так же в дальнейшем на основе этого формата можно реализовать бинарную версию.
3. **ULFEX** - extended, расширенная версия для специфичных направлений, крипто, игры и тд. (Дополнительные правила описания специфичных данных, как раз таки доп поля для крипто кошельков)

Вообщем пока такие наброски по стандарту, приглашаю Quake3 , DildoFagins ,marmalade_knight@marmalade_knight и других заинтересованых кодеров к обсуждению

Програмная часть.

В дальнейшем так же необходимы несколько софтин:

- LogViewer - Гуй для работы с логами в этом формате
- LogProcessor - Бэкэнд часть занимающаяся базовыми операциями по типу сортировки, поиска и тд(возможно и не нужна, можно реализовать средствами БД), а возможно прикрутить тут АПИ, дабы была возможность написания плагинов и расширений. **[3]**
- LogDB - Собственно база данных, либо другой способ хранения логов, как вариант использовать MongoDB, либо другую Document based DB (плюс в том что можно все данные хранить как раз таки в Json)

PыSы0: Список утилит как и стандарт формата являеться тестовым, предназначенным для обрисовки самой идеии. Поэтому любые предложения, конструктивная критика и естественно опенсорс вклад приветствуется.

PыSы1: marmalade_knight , по поводу твоего предложения в посте, предлагаю после утверждения первой версии нашего скромного форумного стандарта, заняться реализацией первой Утилиты, а именно **ULF_LogUnifier***, которая как раз таки позволит конвертировать логи в единый стандарт.*
PыSы2: Давайте вместе сделаем мир малвари качественней и системней (Долой БлэкТимШарперов!!!)
PS: **[n] - вопросы для обсуждения**