

Мониторинг игровых серверов МТА: Province
«ОНЛАЙН СЕРВЕРОВ (ТБРУ)»
by weekthor.ru

ДВОИЧНЫЙ-ТЕКСТОВЫЙ КОД
Для получения информации о серверах МТА: SA

ALL-SEEING EYE (ASE)

Екатеринбург
2021

ОГЛАВЛЕНИЕ

| | |
|--|----|
| Предисловие..... | 3 |
| Введение..... | 3 |
| Источники..... | 4 |
| Условные обозначения..... | 4 |
| СОДЕРЖАНИЕ КОДА..... | 5 |
| Содержание разделов..... | 5 |
| 1. Формат возвращаемых данных..... | 6 |
| 1.1. Код EYE1..... | 6 |
| 1.2. Код EYE2..... | 6 |
| 1.3. Код EYE3..... | 6 |
| 1.4. Версия..... | 6 |
| ДЕКОДИРОВАНИЕ ПОЛУЧЕННОЙ ИНФОРМАЦИИ..... | 7 |
| 1. Декодирование EYE1..... | 7 |
| 2. Декодирование EYE2..... | 9 |
| 3. Декодирование EYE3..... | 11 |

Олег Руинов, В. Двоично-текстовый код для получения информации о серверах МТА: SA : документация / [Виктор «Олег Руинов»]. — Екатеринбург, 2021. — 12 с.
— URL: fantastic-game.ru/blog/wp-content/uploads/2026/02/kod_ase_mta_eye1_eye2_eye3.pdf.
— Текст : электронный

Предисловие

Протокол «All-Seeing Eye» (сокращено: ASE) – протокол передачи информации о сервере (название, онлайн, список игроков и др.), который используется как внутриигровыми браузерами, так и различными мониторингами серверов.

Протокол ASE используется при получении информации в таких играх, как Far Cry, МТА: SA, МТА: Vice City.

Введение

Документация разработана для системы мониторинга игровых серверов МТА: Province «Онлайн серверов (ТБРУ)» (<https://vk.com/mtaprovinceonline>) и содержит информацию по декодированию двоичного кода ASE, отправляемого с игрового сервера МТА в трёх его вариациях: EYE1, EYE2 и EYE3

Документация по декодированию двоичного кода ASE серверов МТА разработана Виктором “Олег Руиновым” (<https://vk.com/weekthor>) для мониторинга состояния игровых серверов МТА: Province в рамках проекта «Онлайн серверов (ТБРУ)» (<https://vk.com/mtaprovinceonline>) и может использоваться в других проектах. Подготовлено на основе анализа исходного кода МТА: SA.

Источники

Multi Theft Auto. ASE.cpp : исходный код (C++) / Multi Theft Auto // GitHub : [сайт].

— URL: <https://github.com/multitheftauto/mtasa-blue/blob/master/Server/mods/deathmatch/logic/ASE.cpp> (дата обращения: 25.11.2021). — Режим доступа: свободный

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

$\boxed{0d}$ — отдельно взятый байт, значение которого равно $0d_{16}$.

СОДЕРЖАНИЕ КОДА

Двоичный код ответа можно условно разделить на 5 разделов:

Раздел 0 — указатель (идентификатор) кода ответа

Раздел 1 — основная информация о сервере

Раздел 2 — основная информация о сервере

Раздел 3 — дополнительная информация о сервере

Раздел 4 — список игроков

Содержание разделов

| Раздел | Содержание |
|----------|--|
| Раздел 0 | Идентификатор кода (4 байта) В текстовом виде: <i>EYE1</i> , <i>EYE2</i> или <i>EYE3</i> |
| Раздел 1 | Содержит поля с информацией о сервере. Первый байт каждого поля указывает длину n (в байтах) текстового содержимого с учетом первого байта. Значение байта равно $(n+1)_{10}$ |
| Раздел 2 | Является одним большим текстовым полем с информацией о сервере, разделенной на параметры. Параметры разделены между собой нулевым байтом 00_{16} Первый байт раздела указывает длину n (в байтах) всего раздела с учетом первого байта. Значение байта равно $(n+1)_{10}$ |
| Раздел 3 | Версия сервера (байт длины + текст), дополнительная информация в двоичном виде: наличие пароля, онлайн и т.п. |
| Раздел 4 | Список игроков. Содержание зависит от кода (формата) ответа из Раздела 0 |

Примечания:

- Содержимое разделов 1-4 варьируется в зависимости от используемого кода (формата ответа)
- Формат ответа зависит от отправляемого на сервер символа (пакета).
- В коде EYE1 **Раздел 2** и **Раздел 3** не включаются
- В коде EYE3 **Раздел 4** не включается

1. Формат возвращаемых данных

1.1. Код EYE1

Содержит основную информацию о сервере: название сервера, его настройки, количество и подробный список игроков (ники игроков, их опыт и пинг. В некоторых играх так же скин и команда). Является главным кодом для запроса информации о сервере по протоколу ASE. Отправляемый символ: s

1.2. Код EYE2

Содержит необходимую для внутриигрового браузера серверов МТА информацию: название сервера, его настройки, количество и сокращенный список игроков (исключительно ники игроков). Отправляемый символ: r

1.3. Код EYE3

Самый короткий код ответа. Содержит исключительно информацию о сервере (настройки и количество игроков). Отправляемый символ: x

1.4. Версия

При запросе этого типа данных в ответ придет лишь версия сервера. Отправляемый символ: v

ДЕКОДИРОВАНИЕ ПОЛУЧЕННОЙ ИНФОРМАЦИИ

Исходный ответ в шестнадцатеричной системе выглядит как:

45 59 45 32 **04** 6d 74 61 **06** 31 32 33 34 35 **07** 73 65 72
76 65 72 ...

Полностью перевести его в человеко-читаемый текст не получится. При переводе получим:

EYE2?mta?12345?server...

где на месте символов «?» находятся байты, которые не удалось привести к читаемому виду. Заменим их на исходное значение:

EYE2^[04]mta^[06]12345...

Эти байты, в большинстве случаев, обозначают длину следующего за ними текста (включая данный байт). Назовем их «байтами длины». Переводя значение первого выделенного байта в десятичную систему, получаем 4. И, действительно, длина сообщения ^[04]mta равна 4 байтам (1 символ = 1 байт).

Первые 4 символа ответа, EYE2 – это тип ответа. Длина этой части постоянна. Меняется лишь цифра. Поэтому, для декодирования следует, начиная с 5-го байта, получить десятичное значение «байта длины» (обозначим через n), а затем прочитать $n-1$ символов после него.

1. Декодирование EYE1

Пример содержания ответа:

Раздел 0:

EYE1

Раздел 1:

^[04]mta^[06]12345^[07]server^[05]type^[04]map^[04]ver^[02]P^[07]online^[0b]max_online

Раздел 4:

F_1F_1 $05nick05team05skin06score05ping05time$...

| поле может повторяться n раз

Раздел 1 состоит из 9 текстовых полей, содержащих основную информацию о сервере. Первый байт каждого поля указывает на длину текстового поля с учетом первого байта длины:

$04mta$ - 4 байта. Идентификатор игры (mta)

0612345 - 6 байт. Порт сервера

$07server$ - 7 байт. Название сервера (server)

$05type$ - 5 байт. Тип игры (type)

$04map$ - 4 байта. Название карты (map)

$04ver$ - 4 байта. Версия сервера (ver)

$02P$ - 2 байта. Наличие (P=1) или отсутствие

(P=0) пароля на сервере

$07online$ - 7 байт. Количество игроков

$0bmax_online$ -11 байт. Количество слотов

Раздел 4 состоит из списка игроков с различными параметрами. Число полей зависит от количества игроков.

Первый байт $F1F1_{16}$ является флагом, значение которого указывает на наличие или отсутствие параметров игрока. Затем в формате «байт длины» + «текстовое поле» передаются определенные параметры. Параметры игрока содержат следующую информацию в упорядоченном виде:

1. Ник игрока ($05nick$)
2. Команда игрока ($05team$)
3. Скин игрока ($05skin$)
4. Счет игрока ($06score$)

5. Пинг игрока ($05_{16}ping$)

6. Игровое время ($05_{16}time$)

Наличие или отсутствие каждого параметра определяется флагом $FlFl_{16}$:

| Двоичное представление флага $FlFl_{16}$ | Содержание параметров | Пример побитового сравнения |
|--|------------------------------|-----------------------------|
| $xxxx\ xxx1$ | Содержит ник игрока | $flag \& 1$ |
| $xxxx\ xx1x$ | Содержит команду игрока | $flag \& 2$ |
| $xxxx\ x1xx$ | Содержит скин игрока | $flag \& 4$ |
| $xxxx\ 1xxx$ | Содержит игровой счет игрока | $flag \& 8$ |
| $xxx1\ xxxx$ | Содержит пинг игрока | $flag \& 16$ |
| $xx1x\ xxxx$ | Содержит игровое время | $flag \& 32$ |

Например, если $FlFl_{16}=0001\ 0111_2$, то параметры игрока содержат ник, команду, скин и пинг. Поле игрока имеет вид:

$17\ 05_{16}nick05_{16}team05_{16}skin05_{16}ping$

Если передается только ник игрока и его команда, флаг имеет вид $03_{16} = 0000\ 0011_2$

Соответственно, флаг $FlFl_{16}$ принимает значения от 01_{16} до $3F_{16}$.

2. Декодирование EYE2

EYE2 04_{16} mta 06_{16} 12345 07_{16} server 05_{16} type 34_{16} map 00_{16} 10/15 00_{16} 1 00_{16} 0 00_{16} ping
status 00{16} netroute 00_{16} uptime 00_{16} http_port 04_{16} ver 00_{16} 00_{16} $0a_{16}$ $0f_{16}$
 05_{16} nick 04_{16} bot...

Пример ответа EYE2. Квадратами выделены «байты длины» и «байты-разделители».

В соответствии со схемой кода ответа:

Раздел 0:

EYE2

Раздел 1:

04mta061234507server05type

Раздел 2:

34map0010/1500100000ping_status00netroute00uptime00http_port

Раздел 3:

04ver 00 00 0a 0f

Раздел 4:

05nick ... | поле может повторяться *n* раз

Раздел 1 состоит из четырех полей. Первый байт каждого поля указывает на длину этого поля в байтах (с учетом первого байта). После первого байта следует текстовое значение поля

| | |
|---|-------------------------------------|
| 04 mta | - 4 байта. Идентификатор игры (mta) |
| 06 12345 | - 6 байт. Порт сервера |
| 07 server | - 7 байт. Название сервера (server) |
| 05 type | - 5 байт. Тип игры (type) |

Раздел 2 состоит из одного большого текстового поля, содержащего различные параметры, разделенные между собой нулевым байтом 00.

Первый байт раздела, $34_{16}=52_{10}$, указывает на длину всего раздела в байтах (52 байта). Раздел содержит следующую информацию в текстовом виде, разделенную только нулевым байтом 00_{16} и не содержащую байтов длины:

| | |
|-------------|--|
| map | - название карты |
| 10/15 | - онлайн на сервере (10 из 15) |
| 1 | - build type |
| 0 | - build number |
| ping_status | - <i>(назначение неизвестно)</i> |
| netroute | - <i>(назначение неизвестно)</i> |
| uptime | - время в секундах с момента запуска сервера |
| http_port | - http-порт сервера |

Раздел 3 состоит из сообщения о версии сервера и 4 байтов. Первый байт указывает на длину текстового поля, содержащего версию сервера, с учетом байта длины. Остальные 4 байта содержат информацию о сервере:

- 1) Установлен ли на сервере пароль (00_{16} – нет, 01_{16} – да).
- 2) Включена ли проверка серийного номера (аналогично).
- 3) Онлайн на сервере $00_{16}-ff_{16}$. Указывается в пределах $[0; 255]$.
- 4) Количество словтов $00_{16}-ff_{16}$, аналогично в пределах $[0; 255]$.

Раздел 4 содержит список игроков. В коде EYE2 список содержит только ники игроков, а сам список игроков неполный (ограничивается ~20-40 игроками). Перед каждым ником следует байт, указывающий на длину ника с учетом байта длины:

$\boxed{05}$ nick - 5 байт. Игрок с ником: nick

3. Декодирование EYE3

EYE3 $\boxed{04}$ mta $\boxed{07}$ server $\boxed{05}$ type $\boxed{0a}$ map $\boxed{00}$ 10/15 $\boxed{04}$ ver $\boxed{00}$ $\boxed{0a}$ $\boxed{0f}$

Пример ответа EYE3. Квадратами выделены «байты длины» и «байты-разделители».

EYE3 декодируется аналогично *EYE2*, но передается меньше данных:

Раздел 0

EYE3

Раздел 1:

04mta07server05type

Раздел 2:

0amap0010/15

Раздел 3:

04ver 00 0a 0f

Раздел 1 содержит только идентификатор игры, название сервера *server* и режим игры *type*. Порт сервера в коде EYE3 не указывается.

Раздел 2 содержит одно большое текстовое поле, разделенное на параметры с помощью нулевого байта 00_{16} . В коде EYE3 раздел 2 содержит следующие параметры:

1. Название карты: *map*
2. Онлайн сервера в текстовом представлении: *10/15*

Раздел 3 состоит из четырех полей:

1. Версия сервера *ver*, где первый байт – «байт длины».
2. Установлен ли пароль на сервере (00 – нет, 01 – да).
3. Онлайн на сервере $00_{16}-ff_{16}$. Указывается в пределах $[0; 255]$.
4. Количество словтов $00_{16}-ff_{16}$, аналогично в пределах $[0; 255]$.

Раздел 4 со списком игроков не включается.