



인스턴트 메시지 아티팩트 분석

System Software Lab

이근탁

CONTENTS



01

Electron

02

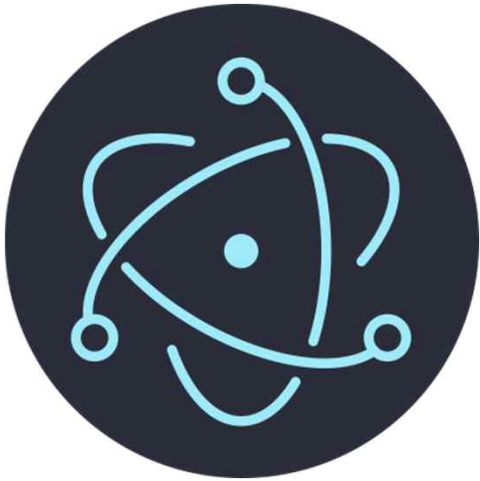
Element

C H A P T E R

01

Electron

01 Electron



OpenJS Foundation 개발

Chromium, NodeJS 기반

HTML, CSS, JS 이용하여
웹 제작하듯이
데스크톱 앱 만드는 플랫폼

역으로 데스크톱 to 웹도 가능



01 Electron

Accounts	2024-05-13 오후 3:55	파일 폴더
AutofillStrikeDatabase	2024-06-12 오후 3:02	파일 폴더
blob_storage	2024-06-12 오후 3:02	파일 폴더
BudgetDatabase	2024-06-12 오후 3:02	파일 폴더
Cache	Service Worker	2024-05-13 오후 3:44 파일 폴더
chrome_cart_db	Session Storage	2024-06-12 오후 3:02 파일 폴더
Code Cache	Sessions	2024-06-12 오후 3:02 파일 폴더
commerce_subscription_db	Shared Dictionary	2024-05-13 오전 9:59 파일 폴더
coupon_db	shared_proto_db	2024-06-12 오후 3:02 파일 폴더
databases	Site Characteristics Database	2024-06-12 오후 3:02 파일 폴더
	Storage	2024-05-13 오전 9:59 파일 폴더
	Sync App Settings	2024-05-17 오후 12:18 파일 폴더
	Sync Data	2024-06-12 오후 3:02 파일 폴더

Chrome 폴더

blob_storage	2024-06-12 오후 3:40	파일 폴더
Cache	2024-06-11 오후 7:11	파일 폴더
Code Cache	2024-06-11 오후 7:11	파일 폴더
databases	2024-06-11 오후 7:12	파일 폴더
DawnGraphiteCache	2024-06-11 오후 10:09	파일 폴더
DawnWebGPUCache	2024-06-11 오후 10:09	파일 폴더
Dictionaries	2024-06-11 오후 7:11	파일 폴더
EventStore	2024-06-12 오전 3:21	파일 폴더
GPUCache	2024-06-11 오후 10:09	파일 폴더
IndexedDB	2024-06-11 오후 10:09	파일 폴더
Local Storage	2024-06-11 오후 10:09	파일 폴더
Network	2024-06-12 오후 4:23	파일 폴더
Session Storage	2024-06-12 오후 3:40	파일 폴더
Shared Dictionary	2024-06-11 오후 7:11	파일 폴더
WebStorage	2024-06-11 오후 7:12	파일 폴더

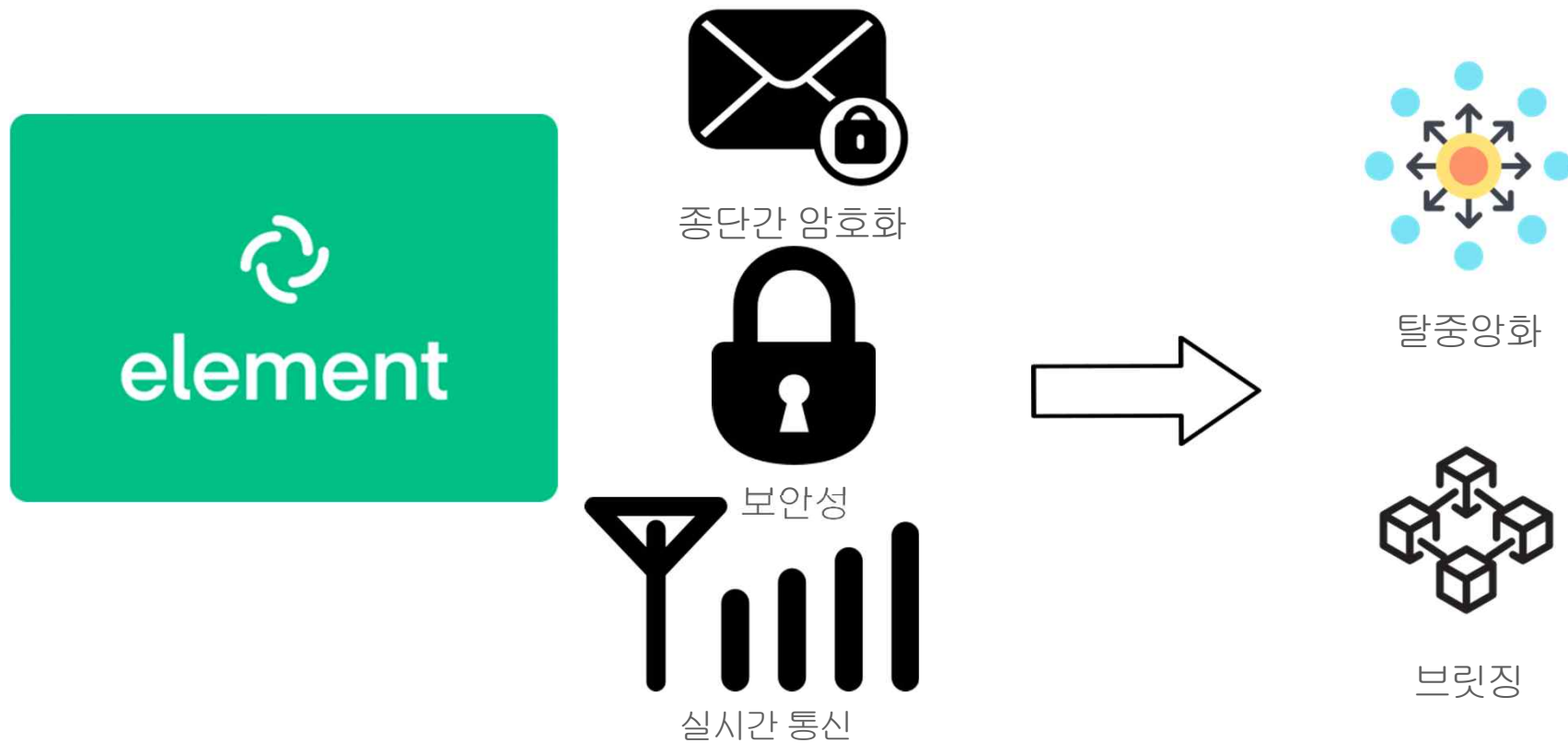
Electron 출신 친구들 폴더

C H A P T E R

02

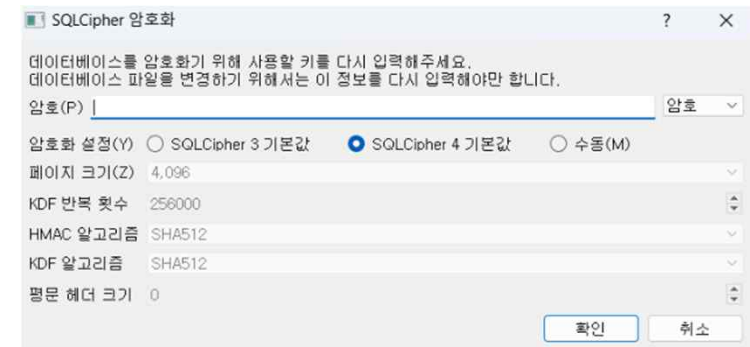
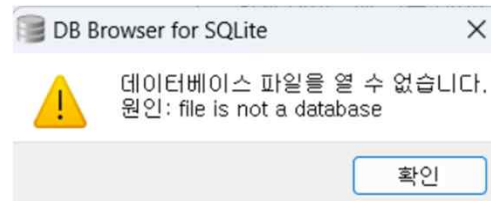
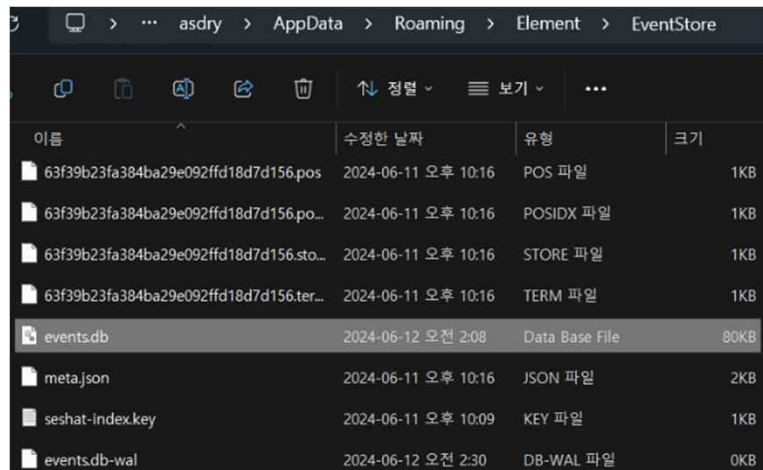
Element

02 Element -Matrix 프로토콜 이용



02 Element -DB 암호화

C:\Users\asdry\AppData\Roaming\Element\EventStore\Events.db



Encrypted.....

02 Element

오픈소스 프로젝트

```
async function getOrCreatePassphrase(key: string): Promise<string> {
  if (keytar) {
    try {
      const storedPassphrase = await keytar.getPassword("element.io", key);
      if (storedPassphrase !== null) {
        return storedPassphrase;
      } else {
        const newPassphrase = await randomArray(32);
        await keytar.setPassword("element.io", key, newPassphrase);
        return newPassphrase;
      }
    } catch (e) {
      console.log("Error getting the event index passphrase out of the secret store", e);
    }
  }
  return seshatDefaultPassphrase;
}
```

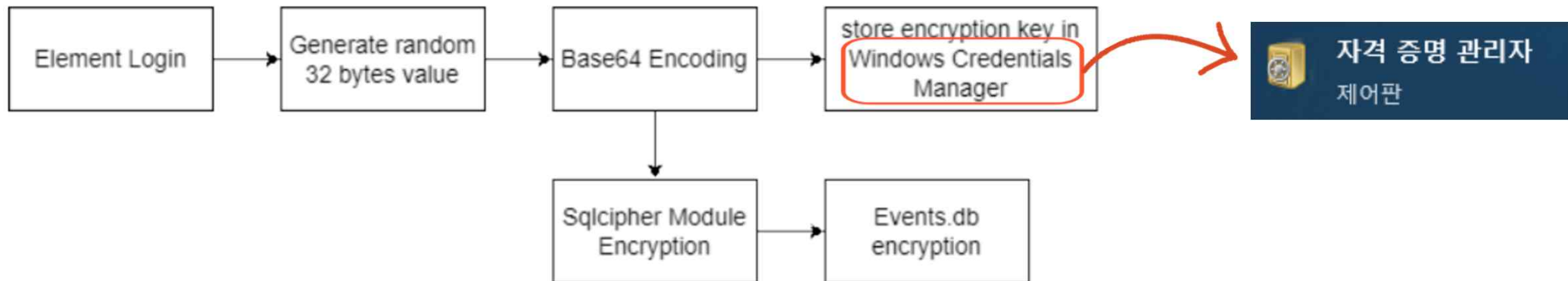
```
bool GetPassword(const std::string& service,
                 const std::string& account,
                 std::string* password) {
  std::string target_name = service + '/' + account;

  CREDENTIAL* cred;
  if (::CredRead(target_name.c_str(), CRED_TYPE_GENERIC, 0, &cred) == FALSE)
    return false;

  *password = std::string(reinterpret_cast<char*>(cred->CredentialBlob),
                          cred->CredentialBlobSize);
  ::CredFree(cred);
  return true;
}
```

02 Element

동작 구조



02 Element

C:\Users\Wsdry\AppData\Roaming\Element\Local Storage\W.log

일반 자격 증명 일반 자격 증명 추가

element.io/seshatj@56605341616:matrix.org|guest_... 수정된 날짜: 2024-06-11

인터넷 또는 네트워크 주소:
element.io/seshatj@56605341616:matrix.org|guest_device

사용자 이름: seshatj@56605341616:matrix.org|guest_device

암호:

지속성: 연터프라이즈

[편집](#) [제거](#)

```
00000000 14 41 04 F4 F1 02 01 01 00 00 00 00 00 00 0D .A.ñ.....
00000010 00 00 00 01 07 56 45 52 53 49 4F 4E 01 31 01 14 .....VERSION.1..
00000020 4D 45 54 41 3A 76 65 63 74 6F 72 3A 2F 2F 76 65 META:vector://ve
00000030 63 74 6F 72 0C 08 CD 8E 9B 81 8B A6 DE 17 10 D6 ctor..î>.<|p..ö
00000040 03 01 27 5F 76 65 63 74 6F 72 3A 2F 2F 76 65 63 ..'_vector://vec
00000050 74 6F 72 00 01 6D 78 5F 63 72 79 70 74 6F 5F 69 tor..mx_crypto_i
00000060 6E 69 74 69 61 6C 69 73 65 64 05 01 74 72 75 65 nitialised..true
00000070 01 1E 5F 76 65 63 74 6F 72 3A 2F 2F 76 65 63 74 .._vector://vect
00000080 6F 72 00 01 6D 78 5F 64 65 76 69 63 65 5F 69 64 or..mx_device_id
00000090 0D 01 67 75 65 73 74 5F 64 65 76 69 63 65 01 25 ..request_device.%
000000A0 5F 76 65 63 74 6F 72 3A 2F 2F 76 65 63 74 6F 72 _vector://vector
000000B0 00 01 6D 78 5F 68 61 73 5F 61 63 63 65 73 73 5F ..mx_has_access_
000000C0 74 6F 6B 65 6E 05 01 74 72 75 65 01 1B 5F 76 65 token..true.._ve
000000D0 63 74 6F 72 3A 2F 2F 76 65 63 74 6F 72 00 01 6D ctor://vector..m
000000E0 78 5F 68 73 5F 75 72 6C 21 01 68 74 74 70 73 3A x_hs_url!.https:
000000F0 2F 2F 6D 61 74 72 69 78 2D 63 6C 69 65 6E 74 2E //matrix-client.
00000100 6D 61 74 72 69 78 2E 6F 72 67 01 1D 5F 76 65 63 matrix.org.._vec
00000110 74 6F 72 3A 2F 2F 76 65 63 74 6F 72 00 01 6D 78 tor://vector..mx
00000120 5F 69 73 5F 67 75 65 73 74 05 01 74 72 75 65 01 _is_guest..true.
00000130 1B 5F 76 65 63 74 6F 72 3A 2F 2F 76 65 63 74 6F ._vector://vecto
00000140 72 00 01 6D 78 5F 69 73 5F 75 72 6C 12 01 68 74 r..mx_is_url..ht
00000150 74 70 73 3A 2F 2F 76 65 63 74 6F 72 2E 69 6D 01 tps://vector.im.
00000160 35 5F 76 65 63 74 6F 72 3A 2F 2F 76 65 63 74 6F 5_vector://vecto
00000170 72 00 01 6D 78 5F 6C 61 62 73 5F 66 65 61 74 75 r..mx_labs_featu
00000180 72 65 5F 66 65 61 74 75 72 65 5F 72 75 73 74 5F re_feature_rust_
00000190 63 72 79 70 74 6F 05 01 74 72 75 65 01 23 5F 76 crypto..true.#_v
000001A0 65 63 74 6F 72 3A 2F 2F 76 65 63 74 6F 72 00 01 ector://vector..
000001B0 6D 78 5F 6C 6F 63 61 6C 5F 73 65 74 74 69 6E 67 mx_local_setting
000001C0 73 12 01 7B 22 6C 61 6E 67 75 61 67 65 22 3A 22 s..{"language":"
000001D0 65 6E 22 7D 01 1C 5F 76 65 63 74 6F 72 3A 2F 2F en".._vector://
000001E0 76 65 63 74 6F 72 00 01 6D 78 5F 75 73 65 72 5F vector..mx_has_
000001F0 69 64 18 01 40 35 36 36 30 35 33 34 31 36 31 36 d..856605341616
00000200 3A 6D 61 74 72 69 78 2E 6F 72 67 01 4C 5F 76 65 matrix.org.._ve
```

02 Element

```
BOOL result = CredEnumerate(NULL, 0, &count, &credentials);
if (!result) {
    std::cerr << "Failed to enumerate credentials: " << GetLastError() << std::endl;
    return;
}

for (DWORD i = 0; i < count; i++) {
    PCREDENTIAL credential = credentials[i];

    std::wcout << L"Credential Name: " << credential->TargetName << std::endl;
    std::wcout << L"Type: " << credential->Type << std::endl;

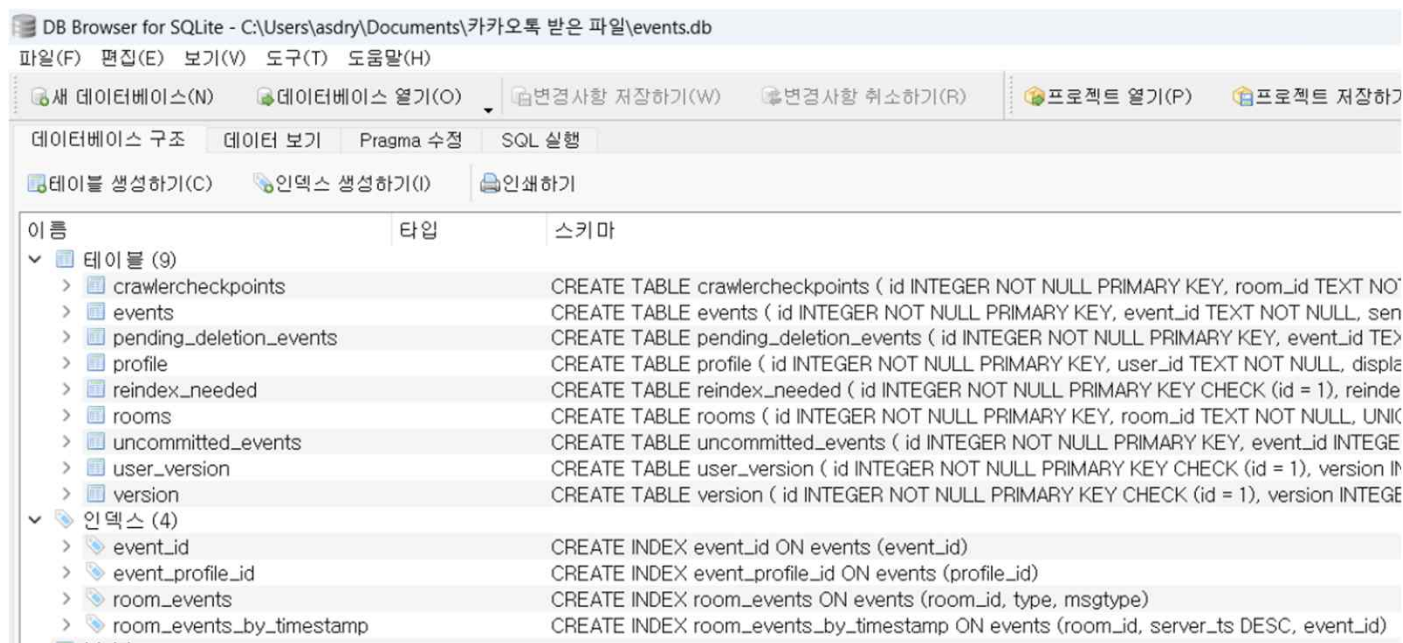
    if (credential->UserName) {
        std::wcout << L"User Name: " << credential->UserName << std::endl;
    }

    if (credential->CredentialBlobSize > 0 && credential->CredentialBlob != NULL) {
        // The password might not be stored in plain text; handling depends on the actual use case.
        std::wcout << L"Credential Blob: " << (char*)credential->CredentialBlob << std::endl;
    }

    std::wcout << L"Persist: " << credential->Persist << std::endl;
    std::wcout << L"-----" << std::endl;
}
```

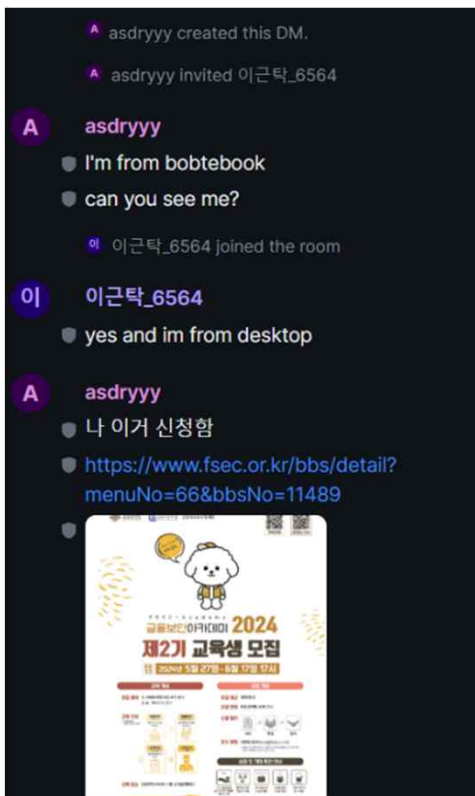
02

Element



복호화 성공

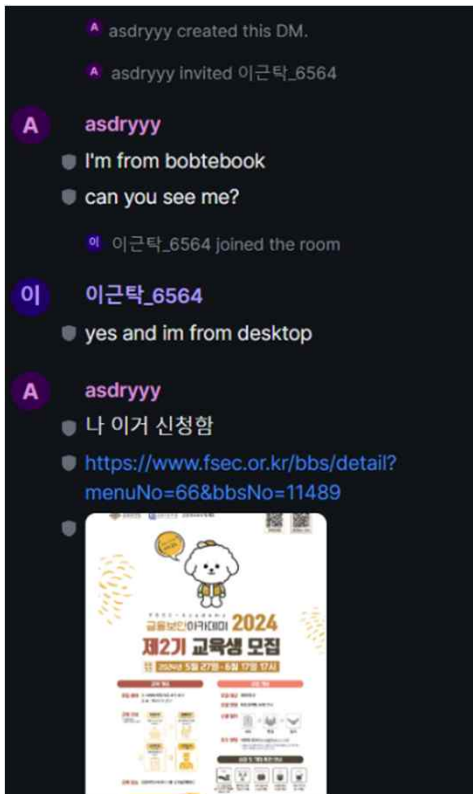
02 Element



테이블(T): profile

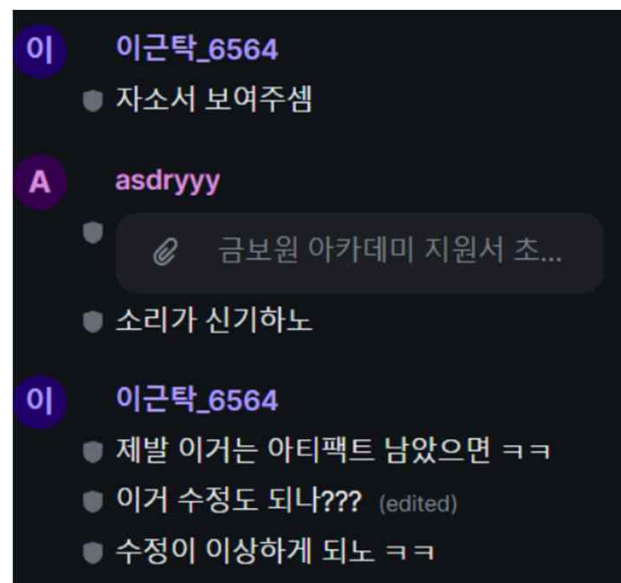
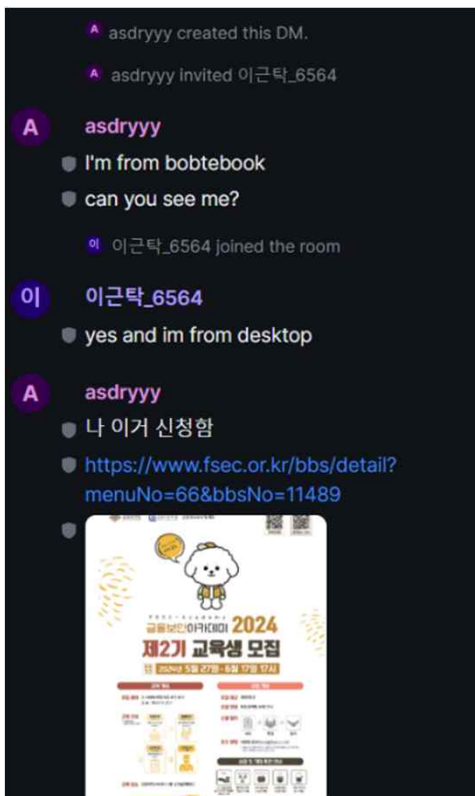
	id	user_id ▼1	displayname	avatar_url
...	필터	필터	필터	필터
1	1	@asdryyy:matrix.org	asdryyy	
2	2	@ggunnttaakk:matrix.org	이근탁_6564	

02 Element



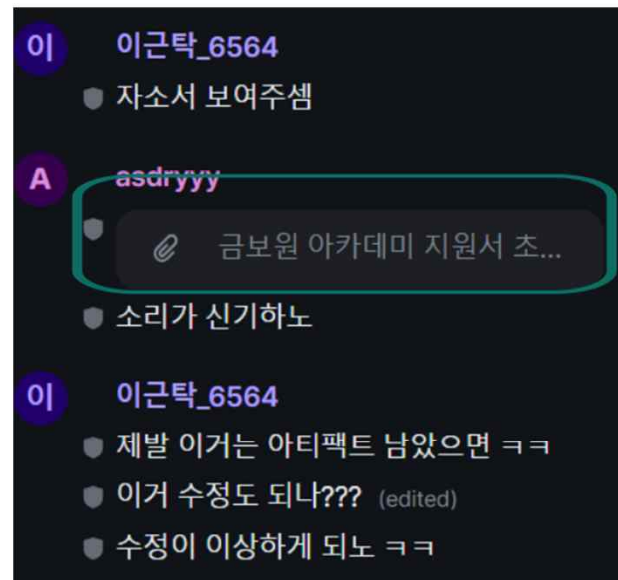
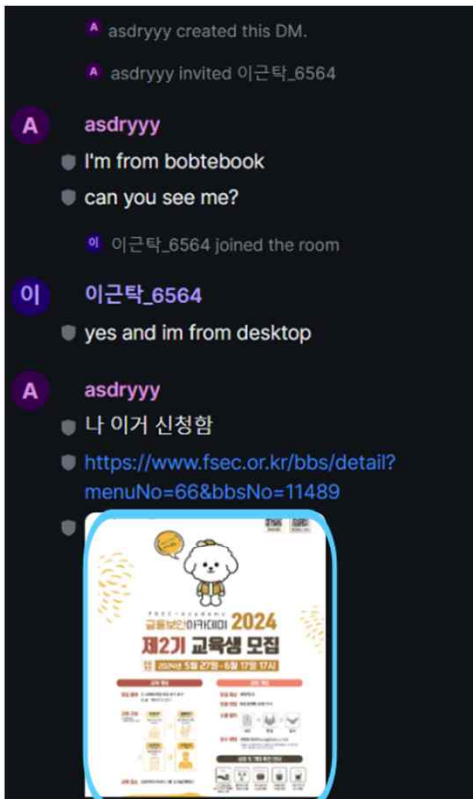
id	event_id	sender	server_ts	room_id	type	msgtype	source	profile_id
...	필터	필터	필터	필터	필터	필터	필터	필터
1	\$9ceD-...	@asdryyy:matrix.org	1718104838400	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-...	1
2	\$MpnmNo57r6jHYx8w_wzVO6NHBdVD...	@asdryyy:matrix.org	1718104839882	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-...	1
3	\$1JyiHjqOXLQKopuGxVLAsX7WDIm_8...	@ggunnnttaakk:matrix.org	1718104890450	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-...	2
4	\$k73iUQi4skkdsR3aw3h543xeUqxjDa7...	@asdryyy:matrix.org	1718104951854	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-...	1
5	\$LZbKLtR62a2xV0_3rxz_aphEZxK3U...	@asdryyy:matrix.org	1718104954200	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-...	1
6	\$pbl7f0psYK1mHTN_UitoTxV1MQo2l8...	@asdryyy:matrix.org	1718104971539	1	m.room.message	m.image	{"algorithm":"m.megolm.v1.aes-...	1
7	\$_D1Q_21qVPGbTWEjqUbT3J6XrbIVCd...	@ggunnnttaakk:matrix.org	1718104978470	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-...	2
8	\$vC9a9Qlnxs_QIN3tkzsfOvzeqGoPmb...	@asdryyy:matrix.org	1718105013083	1	m.room.message	m.file	{"algorithm":"m.megolm.v1.aes-...	1
9	\$U_8tmW-...	@asdryyy:matrix.org	1718105023989	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-...	1
10	\$diBLgTIS5MPLL0Z6E...	@asdryyy:matrix.org	1718105023989	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-...	2
11	\$cQZCZYpLZE0tRkkj...	@asdryyy:matrix.org	1718105023989	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-sha2", "content": {"body": "I'm from bobtebook", "m.mentions": {}}, "msgtype": "m.text"}, "curve25519Key": "8OsUwq1BC8DeyuYrXN7cSfpcCz6Iv0pMoRYVKpOaQGY", "ed25519Key": "jzwKWOZDfMkjIqCucgmk+EYc9cJpLyA/uuuFF8AEdH0", "event_id": "\$9ceD-BC9jmdM6vF8D2DXfvVF5bs7QOFA2AQ2_qOqt7M", "forwardingCurve25519KeyChain": [],	2
12	\$JPJbQjs1F0b7SEqb...	@asdryyy:matrix.org	1718105023989	1	m.room.message	m.text	...	2
13	\$7fk7uy8DMGMZdu...	@asdryyy:matrix.org	1718105023989	1	m.room.message	m.text	...	2

02 Element



```
{"algorithm":"m.megolm.v1.aes-sha2","content":{"body":"I'm from bobtebook","m.mentions":{},"msgtype":"m.text"},"curve25519Key":"8OsUwq1BC8DeyuYrXN7cSfpcCz6lv0pMoRYVKpOaQGY","ed25519Key":"jzwKWOZDfMkjiqCucgmk+EYc9cJpLyA/uuuFF8AEdH0","event_id":"$9ceD-BC9jmdM6vF8D2DXfvVF5bs7QOFA2AQ2_qOqt7M","forwardingCurve25519KeyChain":[],"origin_server_ts":1718104838400,"room_id":"!noZGILvRzUpswsyzmF:matrix.org","sender":"@asdryyy:matrix.org","type":"m.room.message","unsigned":{}}
```

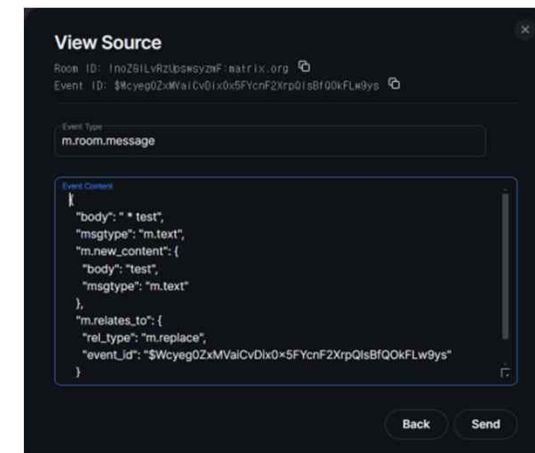
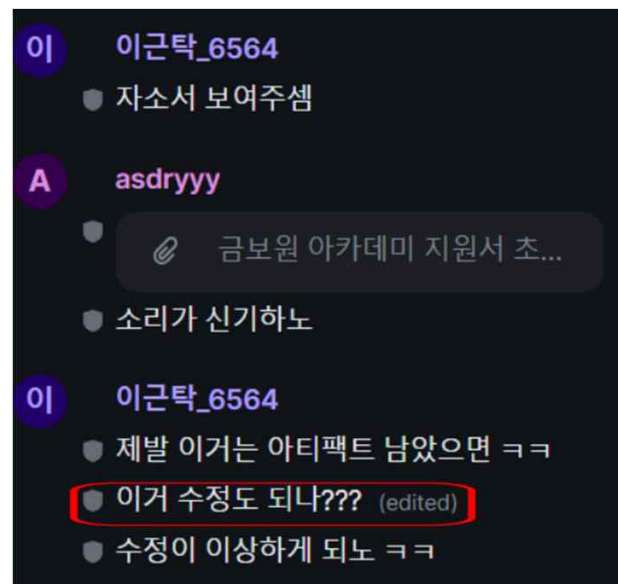
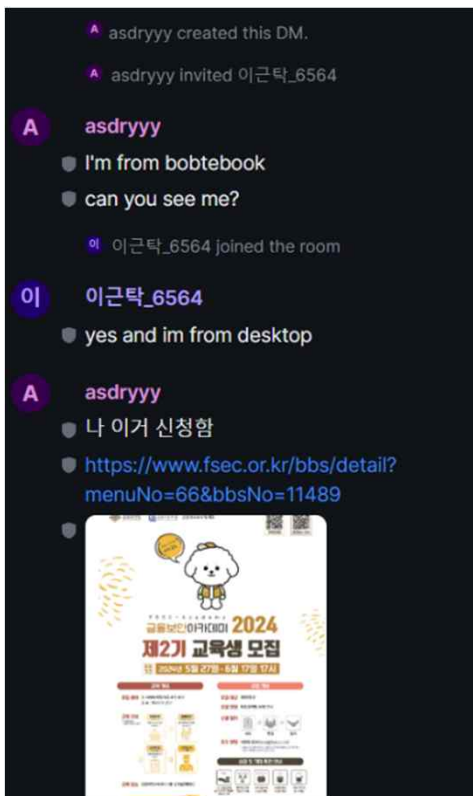

02 Element



"body": "이미지 파일명",
"msgtype": "m.image"
url : "mxc://matrix.org/ryJRYMzPWWPuMDCcgPGosOar"
sha-256 hash

"body": "파일명",
"msgtype": "m.file"
url : "mxc://matrix.org/bAjmnIciHUxDqmmsHZaNtiio"
sha-256 hash

02 Element



```
{ "algorithm": "m.megolm.v1.aes-sha2", "content"
: { "body": " * 이거 수정도 되나??", "m.new_content"
: { "body": "이거 수정도 되나???", "msgtype": "m.text"
}, "m.relates_to": { "event_id":
"$cQZCZYpLZE0tRkkI6nhppdNXiti735ER9RITB_-rRX0"
, "rel_type": "m.replace", "msgtype": "m.text" },
"curve25519Key":
```

02 Element – 한계점

1. 로그아웃 되어 있으면 획득 불가능 (Events.db X)
- 2.
- 2.
2. 로그아웃 후 다시 로그인하면 로그인 이전의 대화 내역은 소실됨
- 3.
- 3.
3. 크레덴셜을 획득하려면 Windows API를 써야하기 때문에 라이브포렌식을 해야함