Dreamhack CTF Season 6 Round #7

32기 송지윤

이번 CTF에서는 리버싱 문제에 도전해 보았다. 문제는 다음과 같다.

파일을 열어보니 html 파일이 눈에 띄었다.

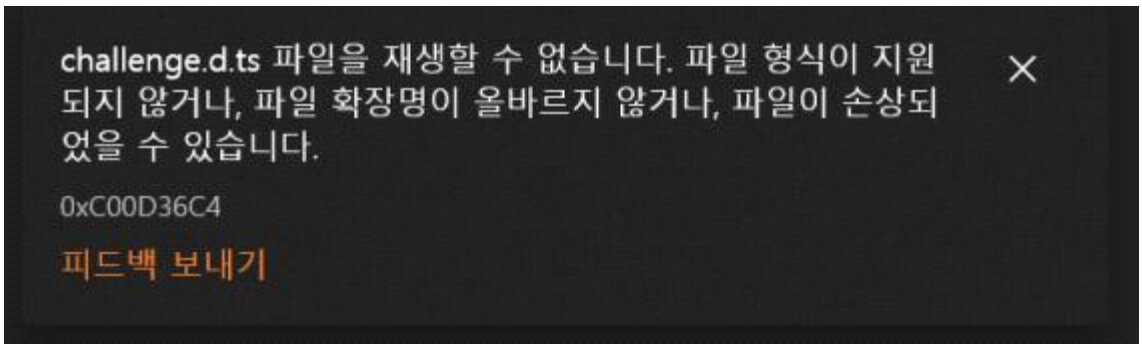
플래그를 찾은 뒤 이곳에 입력하는 것처럼 보인다.

또, 다음과 같은 파일들도 들어있었다.

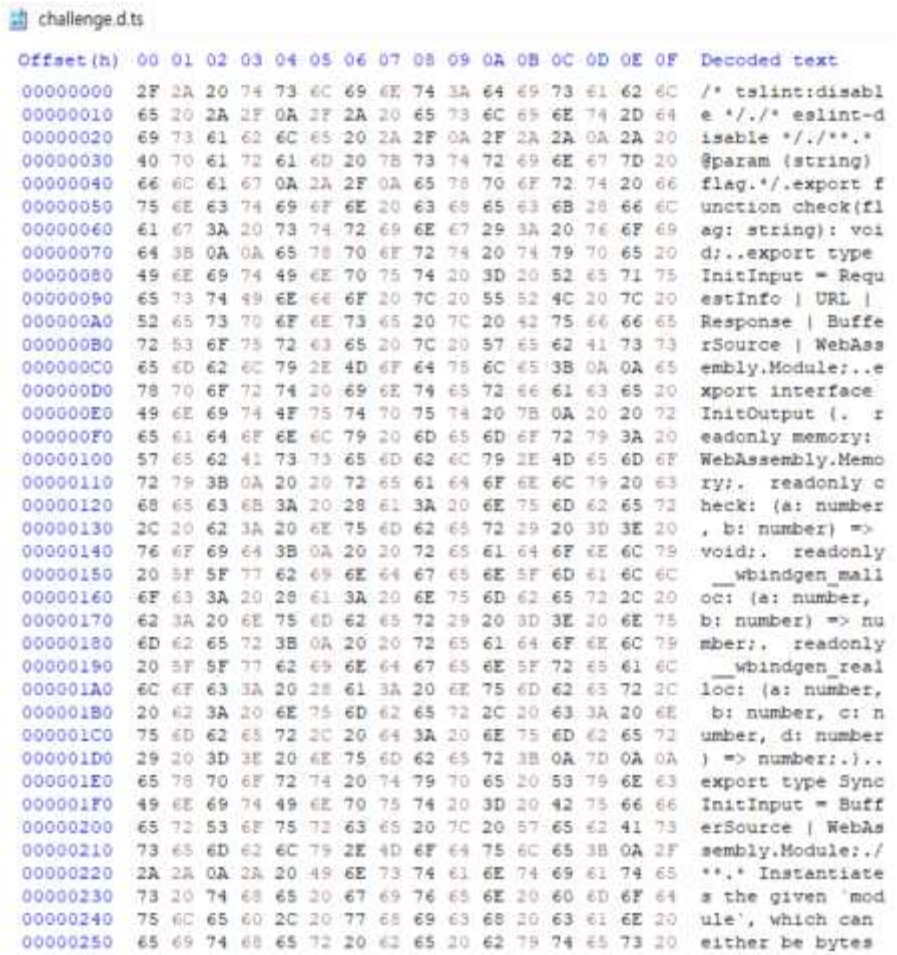
처음 보거나 잘 모르는 유형의 파일들이 있어 우선 파일 유형에 대해 찾아보았다.

• TS 파일: TypeScript 파일을 의미하며 TypeScript는 자바스크립트의 상위집합 언어이다.
• WASM 파일: WebAssembly 파일을 의미하며 이는 웹 브라우저에서 실행할 수 있는 이진 포맷 코드이다.
• JSON 파일: JavaScript Object Notation의 약자로, 데이터를 텍스트 형식으로 저장하고 교환하기 위한 경량 데이터 형식이다.

우선 challenge.d 파일을 열어보았으나 오류가 발생해 제대로 열어볼 수 없었다.



파일이 손상되었을 수 있다고 해서 혹시 ts파일에도 파일 시그니처가 있는지 궁금해 찾아보았지만, ts 파일은 기본적으로 텍스트 파일이기 때문에 파일 시그니처가 따로 존재하지 않는다고 한다. 우선 HxD로 파일을 열어보았다.


challenge.d.ts

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	2F	2A	20	74	73	6C	69	6E	74	3A	64	69	73	61	62	6C	/* tslint:disabl
00000010	65	20	2A	2F	0A	2F	2A	20	65	73	6C	69	6E	74	2D	64	e */./* eslint-d
00000020	69	73	61	62	6C	65	20	2A	2F	0A	2F	2A	2A	0A	2A	20	isable */./**.*
00000030	40	70	61	72	61	6D	20	7B	73	74	72	69	6E	67	7D	20	@param {string}
00000040	66	6C	61	67	0A	2A	2F	0A	65	78	70	6F	72	74	20	66	flag.*/.export f
00000050	75	6E	63	74	69	6F	6E	20	63	68	65	63	6B	28	66	6C	unction check(fl
00000060	61	67	3A	20	73	74	72	69	6E	67	29	3A	20	76	6F	69	ag: string): voi
00000070	64	3B	0A	0A	65	78	70	6F	72	74	20	74	79	70	65	20	d;..export type
00000080	49	6E	69	74	49	6E	70	75	74	20	3D	20	52	65	71	75	InitInput = Requ
00000090	65	73	74	49	6E	66	6F	20	7C	20	55	52	4C	20	7C	20	estInfo | URL |
000000A0	52	65	73	70	6F	6E	73	65	20	7C	20	42	75	66	66	65	Response | Buffe
000000B0	72	53	6F	75	72	63	65	20	7C	20	57	65	62	41	73	73	rSource | WebAss
000000C0	65	6D	62	6C	79	2E	4D	6F	64	75	6C	65	3B	0A	0A	65	embly.Module;..e
000000D0	78	70	6F	72	74	20	69	6E	74	65	72	66	61	63	65	20	xport interface
000000E0	49	6E	69	74	4F	75	74	70	75	74	20	7B	0A	20	20	72	InitOutput {. r
000000F0	65	61	64	6F	6E	6C	79	20	6D	65	6D	6F	72	79	3A	20	eadonly memory:
00000100	57	65	62	41	73	73	65	6D	62	6C	79	2E	4D	65	6D	6F	WebAssembly.Memo
00000110	72	79	3B	0A	20	20	72	65	61	64	6F	6E	6C	79	20	63	ry;. readonly c
00000120	68	65	63	6B	3A	20	28	61	3A	20	6E	75	6D	62	65	72	heck: (a: number
00000130	2C	20	62	3A	20	6E	75	6D	62	65	72	29	20	3D	3E	20	, b: number) =>
00000140	76	6F	69	64	3B	0A	20	20	72	65	61	64	6F	6E	6C	79	void;. readonly
00000150	20	5F	5F	77	62	69	6E	64	67	65	6E	5F	6D	61	6C	6C	__wbindgen_malli
00000160	6F	63	3A	20	28	61	3A	20	6E	75	6D	62	65	72	2C	20	oc: (a: number,
00000170	62	3A	20	6E	75	6D	62	65	72	29	20	3D	3E	20	6E	75	b: number) => nu
00000180	6D	62	65	72	3B	0A	20	20	72	65	61	64	6F	6E	6C	79	mber;. readonly
00000190	20	5F	5F	77	62	69	6E	64	67	65	6E	5F	72	65	61	6C	__wbindgen_real
000001A0	6C	6F	63	3A	20	28	61	3A	20	6E	75	6D	62	65	72	2C	loc: (a: number,
000001B0	20	62	3A	20	6E	75	6D	62	65	72	2C	20	63	3A	20	6E	b: number, c: n
000001C0	75	6D	62	65	72	2C	20	64	3A	20	6E	75	6D	62	65	72	umber, d: number
000001D0	29	20	3D	3E	20	6E	75	6D	62	65	72	3B	0A	7D	0A	0A	) => number;.}..
000001E0	65	78	70	6F	72	74	20	74	79	70	65	20	53	79	6E	63	export type Sync
000001F0	49	6E	69	74	49	6E	70	75	74	20	3D	20	42	75	66	66	InitInput = Buff
00000200	65	72	53	6F	75	72	63	65	20	7C	20	57	65	62	41	73	erSource | WebAs
00000210	73	65	6D	62	6C	79	2E	4D	6F	64	75	6C	65	3B	0A	2F	sembly.Module;./
00000220	2A	2A	0A	2A	20	49	6E	73	74	61	6E	74	69	61	74	65	**.* Instantiate
00000230	73	20	74	68	65	20	67	69	76	65	6E	20	60	6D	6F	64	s the given `mod
00000240	75	6C	65	60	2C	20	77	68	69	63	68	20	63	61	6E	20	ule`, which can
00000250	65	69	74	68	65	72	20	62	65	20	62	79	74	65	73	20	either be bytes

아래는 HxD에 쓰여있던 내용을 메모장으로 옮긴 것이다.

```
/* tslint:disable */
/* eslint-disable */
/**
* @param {string} flag
*/
export function check(flag: string): void;

export type InitInput = RequestInfo | URL | Response | BufferSource | WebAssembly.Module;

export interface InitOutput {
  readonly memory: WebAssembly.Memory;
  readonly check: (a: number, b: number) => void;
  readonly __wbindgen_malloc: (a: number, b: number) => number;
  readonly __wbindgen_realloc: (a: number, b: number, c: number, d: number) => number;
}

export type SyncInitInput = BufferSource | WebAssembly.Module;
/**
* Instantiates the given `module`, which can either be bytes or
* a precompiled `WebAssembly.Module`.
*
* @param {SyncInitInput} module
*
* @returns {InitOutput}
*/
export function initSync(module: SyncInitInput): InitOutput;

/**
* If `module_or_path` is {RequestInfo} or {URL}, makes a request and
* for everything else, calls `WebAssembly.instantiate` directly.
*
* @param {InitInput | Promise<InitInput>} module_or_path
*
* @returns {Promise<InitOutput>}
*/
export default function __wbg_init (module_or_path?: InitInput | Promise<InitInput>): Promise<InitOutput>;
```

열기는 했지만 TypeScrip에 대한 지식이 없어 해석하기는 힘들 것 같다.
자바스크립트 언어에 대한 지식이 많아야 문제를 푸는 데 큰 도움을 얻을 수 있을 것 같다.