PhenoProtect new-version Cracking (Zanzarah - German version)

GAME: Zanzarah – Das verborgene Portal (German version) [https://www.mobygames.com/game/7120/zanzarah-the-hidden-portal]
Protection: phenoProtect (new version)
Author: Luca D'Amico - https://www.lucadamico.dev - V1.0 - 27-Sep-2024

Cosa ci serve:
- Windows 10/11 VM
- isdcc 1.22 [https://archive.org/download/InstallShieldTools/isdcc122.zip]
- isdcc 2.10 with mkcrc.exe [https://ftp.area51.dev/pub/os2/hobbes.nmsu.edu/download/pub/windows/util/archiver/isDcc_2-10.zip]
- HxD [https://mh-nexus.de/en/downloads.php?product=HxD20]
- UniExtract v2 [https://github.com/Bioruebe/UniExtract2]
- Disco di gioco originale

Prima di iniziare:
Darò per scontato che hai già letto il mio precedente documento su phenoProtect (Carrera Grand Prix), quindi molti dettagli su questo DRM non verranno trattati nuovamente. Se così non fosse, ti consiglio di leggerlo prima e poi tornare qui.
Anche questa volta, verranno presentati due metodi differenti per bypassare questa nuova versione di phenoProtect.

NOTA IMPORTANTE (per il Metodo A): Quando ho iniziato la rimozione di phenoProtect dallo script di installazione, non ero a conoscenza dell'esistenza del più recente isdcc 2.10, quindi ho usato la versione 1.22 (del 1998). È stato di gran lunga più difficile identificare gli opcode dello script di InstallShield usando questa versione più vecchia. Quindi, per il bene della tua sanità mentale, usa la versione 2.10: otterrai uno script ricompilabile (non rilevante in questo caso specifico) e un bell'elenco degli opcode decodificati per ogni riga!

phenoProtect trigger: Otterrai un errore al 50% del processo di installazione quando viene usata una immagine del disco.

METODO A:
È un pò più difficile patchare questa versione rispetto a quella trovata in Carrera Grand Prix, perché lo script è in formato .ins. Non sono riuscito a trovare uno strumento simile a SID, quindi ho usato isdcc per decompilare lo script. Utilizzando isdcc versione 1.22, non sarai in grado di ricompilare questo script nella

sua forma binaria utilizzando InstallShield 5 (otterrai molti errori durante la compilazione). Pertanto, ho dovuto modificare il file in formato esadecimale dopo aver identificato gli opcode corretti.

È stato piuttosto impegnativo, perché non avevo alcuna conoscenza pregressa degli script InstallShield da una prospettiva binaria.
Per fortuna, ho trovato questo ottimo documento scritto da un cracker molto abile:
https://www.darkridge.com/~jpr5/mirror/fravia.org/natz51.htm
La parte interessante è nel capitolo riguardante il "secondo approccio".
Adesso sappiamo che un'istruzione "if" inizia con i seguenti bytes:
28, 01, 32
Nello script decompilato, se guardiamo attentamente, troveremo una stringa chiamata **"COPYPROTECTION"**.
Questa è la parte rilevante del decompilato :

```
begin
        Enable(4);
        StrLoadString("", "COPYPROTECTION", lString0);
        SetStatusWindow(0, lString0);
        Delay(2);
        Disable(54);
        StatusUpdate(1, 100);
        function109();
        lNumber2 = LAST_RESULT;
        lNumber2 = lNumber2 = 0;
        if (lNumber2 = 0) then
            goto label775;
        endif;
        StrLoadString("", "COPYPROTECTION_FAILED", lString0);
        MessageBox(lString0, -65533);
        number34 = 1;
        return(-1);

label775: //Ref: 00DF73
        StrLoadString("", "COPY_PROGRAM_FILES", lString0);
        SetStatusWindow(0, lString0);
        lNumber1 = 1;
        ComponentMoveData(MEDIA, lNumber1, 0);
        lNumber0 = LAST_RESULT;
        function112(lNumber0);
        Disable(4);
        return(lNumber0);
        return;
end;
```

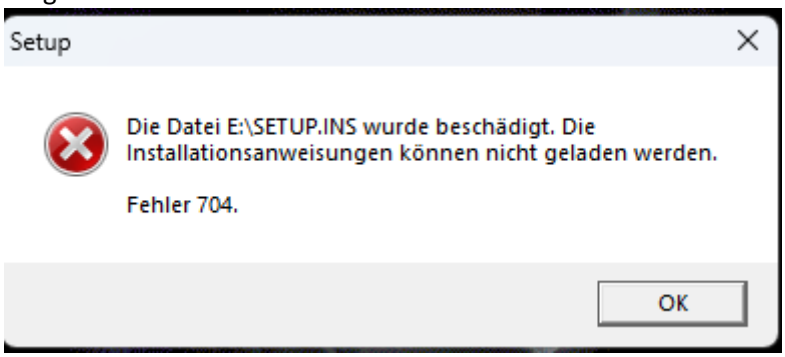
Presta attenzione alla condizione "**if (lNumber2 = 0)**": se questa non viene soddisfatta, otterremo l'errore "**COPYPROTECTION_FAILED"**.
Dobbiamo invertire questa condizione per bypassare il controllo. Per farlo, dobbiamo modificare la riga di codice subito prima facendola diventare:
lNumber2 = lNumber2 != 0;
Usando un editor esadecimale (**HxD**) e cercando la stringa **"COPYPROTECTION_FAILED"**, arriveremo qui:

```
0000DEF0  2F 01 B7 00 41 00 00 00 00 B8 00 00 00 10 00 B6  /.·.A.....¸.....¶
0000DF00  00 10 00 01 00 FF FF 00 00 03 00 00 00 01 00 41  .....ÿÿ........A
0000DF10  04 00 00 00 12 01 61 00 00 61 0E 00 43 4F 50 59  ......a..a..COPY
0000DF20  50 52 4F 54 45 43 54 49 4F 4E 52 9B FF 08 00 41  PROTECTIONR›ÿ..A
0000DF30  00 00 00 00 62 9B FF 0A 00 41 02 00 00 00 02 00  ....b›ÿ..A......
0000DF40  41 36 00 00 00 64 00 41 01 00 00 00 41 64 00 00  A6...d.A....Ad..
0000DF50  00 B5 00 80 6D 00 70 25 03 21 00 32 99 FF 42 00  .µ.€m.p%.!.2™ÿB.
0000DF60  00 28 01 32 99 FF 42 99 FF 41 05 00 00 00 41 00  .(.2™ÿB™ÿA....A.
0000DF70  00 00 00 22 00 70 07 03 95 42 99 FF 41 00 00 00  ...".p..•B™ÿA...
0000DF80  00 12 01 61 00 00 61 15 00 43 4F 50 59 50 52 4F  ...a..a..COPYPRO
0000DF90  54 45 43 54 49 4F 4E 5F 46 41 49 4C 45 44 52 9B  TECTION_FAILEDR›
0000DFA0  FF 2A 00 62 9B FF 41 03 00 FF FF 21 00 32 28 00  ÿ*.b›ÿA..ÿÿ!.2(.
0000DFB0  41 01 00 00 00 2F 01 B7 00 41 FF FF FF FF 00 00  A..../.·.Aÿÿÿÿ..
```

Se guardiamo con attenzione, individueremo la sequenza di bytes **28, 01, 32**: è l'inizio del nostro "if"!

```
0000DF50  00 B5 00 80 6D 00 70 25 03 21 00 32 99 FF 42 00  .µ.€m.p%.!.2™ÿB.
0000DF60  00 28 01 32 99 FF 42 99 FF 41 05 00 00 00 41 00  .(.2™ÿB™ÿA....A.
0000DF70  00 00 00 22 00 70 07 03 95 42 99 FF 41 00 00 00  ...".p..•B™ÿA...
```

All'indirizzo DF6A, è presente il valore 0x5. Se lo modifichiamo in 0x6, invertiremo la condizione.
Dopo aver applicato questa modifica, se decompiliamo nuovamente, otterremo:
lNumber2 = lNumber2 != 0;
Ottimo, adesso possiamo sovrascrivere il file setup.ins originale con quello appena modificato e creare una nuova ISO.
Proviamo ad installare il gioco e…. errore:



Usando Google Translate per tradurre il messaggio d'errore, notiamo che il programma di installazione non è contento perchè il nostro file setup.ins modificato è "corrotto".
Poichè abbiamo modificato questo file, probabilmente abbiamo rotto qualche CRC.
Ho impiegato un sacco di tempo per risolvere questo problema. Dopo ore, ho scoperto che i file .ins hanno un valore di CRC nel loro header.
A questo punto, leggendo un forum Cinese, ho scoperto che una nuova versione di isdcc (v2.10) non solo decompila gli script .ins producento un risultato abbastanza accurato da essere ricompilabile, ma include anche un eseguibile chiamato mkcrc.exe che calcola e aggiorna automaticamente il valore del CRC negli script modificati. Inoltre, questa versione fornisce un output 1:1 tra opcode e codice decompilato, così puoi evitare di perdere tempo cercando di capire quali opcode modificare.

Quindi lanciamo mkcrc.exe sul nostro script modificato e creiamo nuovamente una ISO.
Stavolta tutto funziona e possiamo completare l'installazione del gioco senza errori.

METODO B:

Qui non c'è nulla di nuovo. Anche su questa nuova versione puoi estrarre i file .cab usando UniExtract per ottenere i file di gioco. La cartella principale si trova nel file data1.cab. Ricorda di creare una cartella chiamata "save" (dove sono state estratte le cartelle Data, Resource e System), altrimenti non potrai salvare durante il gioco.

Credits:

Come nel documento precedente: Ringrazio sentitamente Codecult/Phenomedia AG per aver creato questa protezione unica nel suo genere ed interessante. Ringrazio anche gli autori degli strumenti usati per effettuare questa analisi e realizzare questo documento tecnico.

Conclusione:

Questa seconda versione phenoProtect non è stata particolarmente difficile da rimuovere, ma resta comunque divertente da studiare! :)

Grazie per la lettura!

Luca