

Animal Kingdom

(C) 1985-88 Unicorn Software

Crack Study

There are two known versions of protection on this game. We'll call them V1 and V2. V1 is a simple error block check, and V2 is more complex, reading in data from an "unused" area of the disk and "decrypting" it. Both are simple to defeat.

I'm going to guess that the more complex version came later, probably in the 1988 release. The error check was pretty common earlier on.

Also, to clarify, the original I received from GP in 2023 has the V1 protection, the version that surfaced recently in another group's release is the V2 version.

Cracking V1

A small loader stub loads to \$0120 and autoexecutes. It sends the command "B-E:2,0,35,4" to the drive, and waits for the error code to be returned (there is a soft error on the disk). If the error code is detected, it loads and runs the file "GH". Otherwise it jumps to a JAM instruction. As with all of these early error checks, you could crack them by nopping out the branch to the fail, but in this case it's even easier: simply scratch "UNICORN" and rename "GH" to "UNICORN" and you have a cracked disk.

Cracking V2

This version is somewhat more complicated, but the end result is the same. They have moved the "GH" file to "UNICORN" and inserted some protection code at the beginning of the program binary "OB.0".

This protection code reads the block at track 6, sector 3 in to \$2800 (the beginning of OB.0) in several passes until it knows it got what it expects, then eors the data in another several passes until it settles in again, then finally jumps to \$2800. They even seem to have messed with the sync on the disk to make the reads unreliable, presumably that's the reason for the multipass read dance.

The weakness here, of course, is that you can just wait until the protection runs and save out the result. With modern tools (VICE) simply breakpoint 2800 and the second time it hits it, save your "final" OB.0 and you have a working coyp. With traditional tools, modify the loader so it jumps to an endless loop at the end of the process, then freeze and save your "final" OB.0

LOADER STUB FOR U2, "encrypted trackdata"

```
      *=$0340
START  JSR  SENDCOMMAND
      LDX  #$03
      JSR  CHKIN
      LDY  #$00
      STY  $FB

LOOP1  JSR  IECIN    ;read a byte from the bus

FIRSTOPERATION
      STA  $2800,Y
      STA  $2800,Y
      CLC
      ADC  $FB
      STA  $FB
      INY
      BNE  LOOP1

      LDA  $FB
      CMP  #$80
      BEQ  DONE
      LDA  #$59 ;opcode for EOR $XXXX,y
      STA  FIRSTOPERATION ; will now be EOR $2800,y
      BNE  START

DONE   JSR  CLRCHN
      LDA  #$03
      JSR  CLOSE
      JSR  CLALL
      LDA  #$20 ;opcode for JSR $XXXX
      STA  $2800
      LDA  #$4C ;opcode for JMP $XXXX
      BIT  FIRSTOPERATION
      STA  FIRSTOPERATION ;will now be JMP $2800
      BPL  FIRSTOPERATION
      JAM

SENDCOMMAND ; sends the command to the drive
      LDA  $A2
      BMI  SENDCOMMAND ;delaying based on the jiffy clock
      JSR  CLRCHN
      LDX  #$0F
      JSR  CHKOUT
      LDY  #$00

LOOP2  LDA  COMMAND,Y
      BEQ  COMMANDSENT
      JSR  CHROUT
      INY
      BNE  LOOP2

COMMANDSENT
      JMP  CLRCHN

COMMAND
      .byte $55,$31,$3a,$33,$20,$30,$20,$30 ;"U1:3 0 06 03"
      .byte $36,$20,$30,$33,$0d,$00,$ff,$00
```



```

*= $0120
LDA # $08
JSR LISTEN
LDA # $F2
JSR LSTNSA
LDA # $23
JSR IECDUT
JSR UNLSTN
JSR UNTALK
LDA # $08
JSR LISTEN
LDA # $FF
JSR LSTNSA
LDX # $0C
LOOP1
LDA BLOCKCMD,X
JSR IECDUT
DEX
BNE LOOP1
LDX # $00
LDY # $00
DELAYLOOP
DEY
BNE DELAYLOOP
DEX
BNE DELAYLOOP

JSR UNLSTN
JSR CLRCHN
LDA # $05
DELAYLOOP2
INX
BNE DELAYLOOP2
INY
BNE DELAYLOOP2
SEC
SBC # $01
BNE DELAYLOOP2

LDA # $08
JSR TALK
LDA # $0F
JSR TALKSA
JSR IECIN
CMP # $31 ; "1"
BCC JAM ; NOP this is one crack
JSR UNTALK
LDA # $08
TAX
TAY
JSR SETLFS
LDA # $02
LDX # <FILENAME
LDY # >FILENAME
JSR SETNAM
LDA # $00
STA $9D
JSR LOAD
JSR $E544 ; clear screen
LDA $D021 ; background
STA $0286 ; char color
LDX # $05
STX $C6 ; tell keyboard buffer 5 chars are there
DEX
LOOP3
LDA RUNCMD,X
STA $0277,X
DEX
BPL LOOP3
LDA # $0A
STA $2E
JMP $A483 ; basic warm start
BLOCKCMD ; $01af
.byte $a4,$34,$2c,$35,$33,$2c,$30,$2c ; "4,53,0,2:E-B"
.byte $32,$3a,$45,$2d,$42
FILENAME ; $01bc
.byte $47,$48 ; "GH"
RUNCMD ; $01be
.byte $52,$55,$4e,$3a,$0d ; "RUN:<RETURN>"
JAM ; $01e0
.byte $02

```



U1 BAM



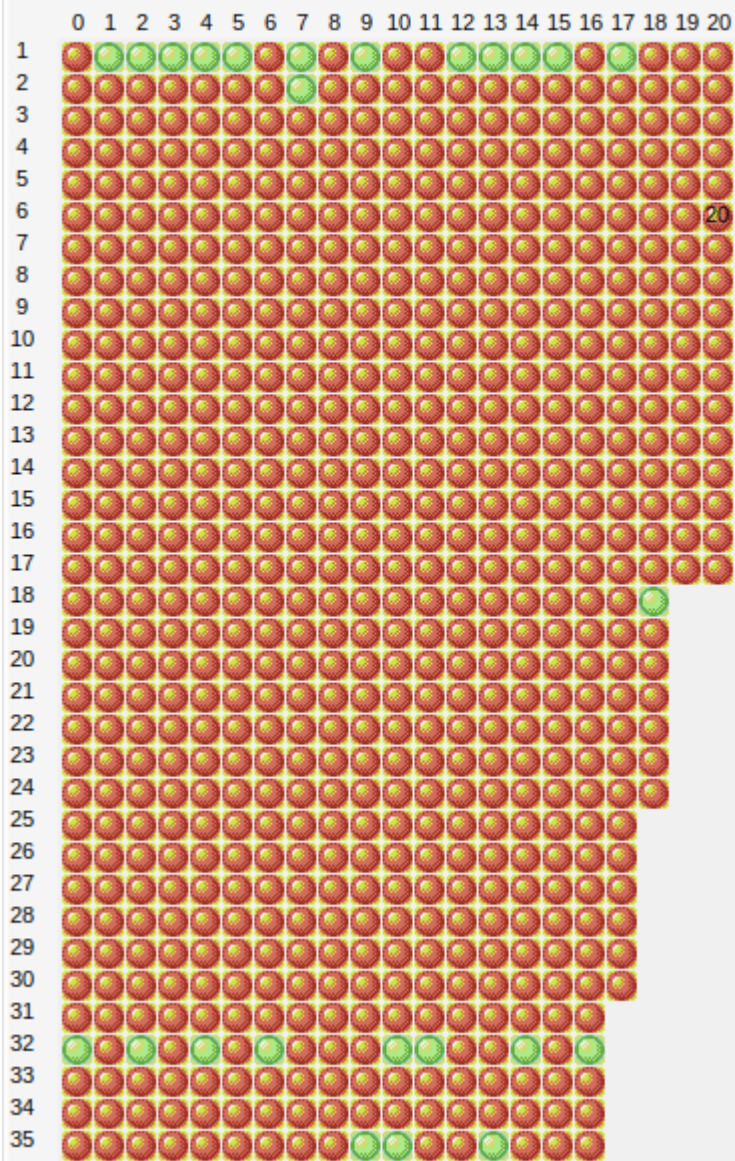
V2 BAM

NEWPIC

JC 2A

Free:  Used: 

Track: 35 Sector: 16



U2 forensic data

