

Interfacekulturens digitale infektioner

Computervirus' æstetik og kultur

Jacob Nielsen

DARC
DIGITAL AESTHETICS
RESEARCH CENTER

Specialeserie/Master Thesis

**Specialeserie/Master Thesis:
Digital Aesthetics Research Center
Aarhus University
Helsingforsgade 14, DK-8200
www.digital-aestetik.dk**

Jacob Nielsen
"Interfacekulturens digitale infektioner – Computervirus' æstetik og kultur"

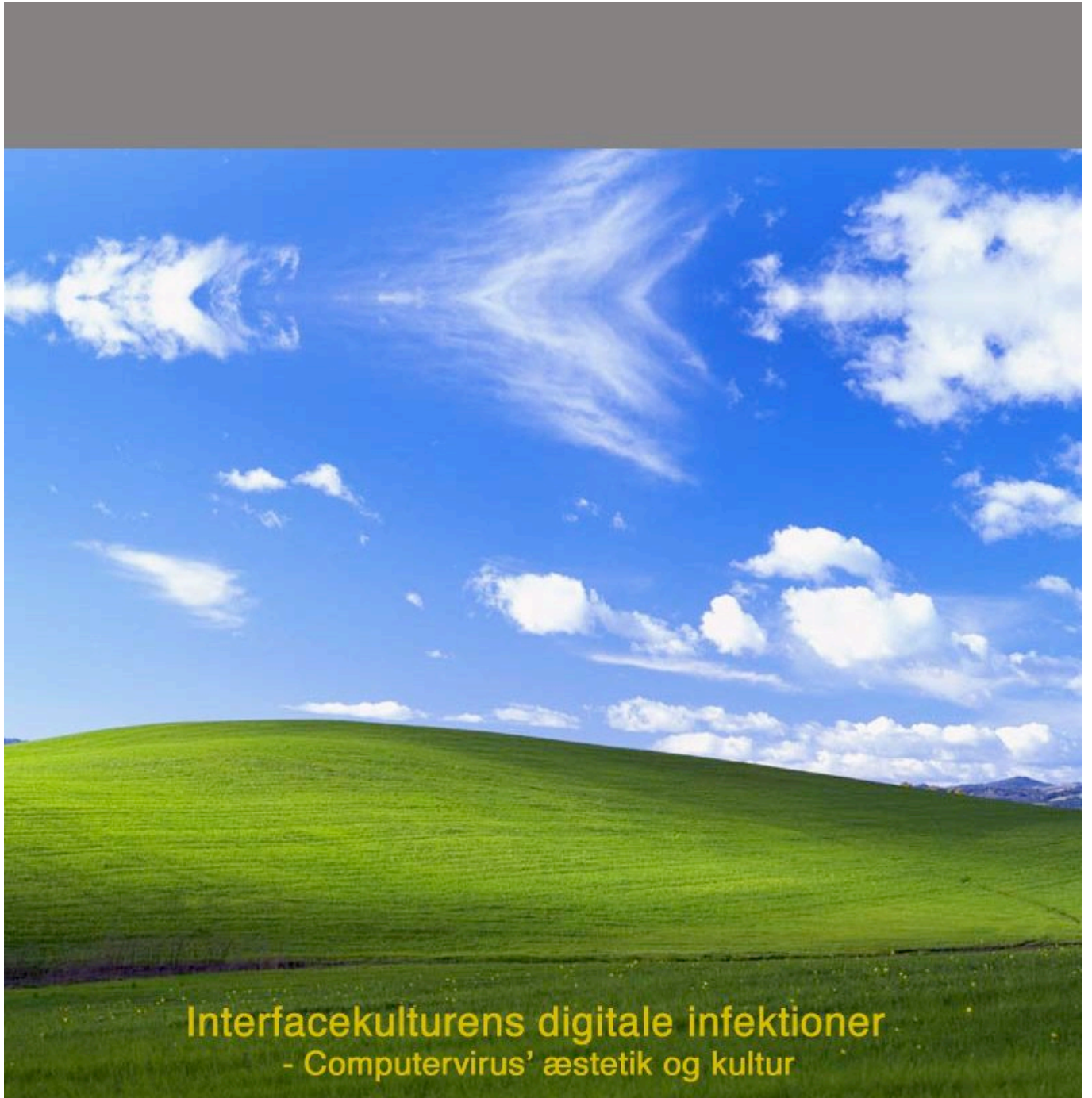
Udgiver: Digital Aesthetics Research Center (Aarhus University)

Copyright © 2009 Jacob Nielsen

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
<http://www.gnu.org/licenses>

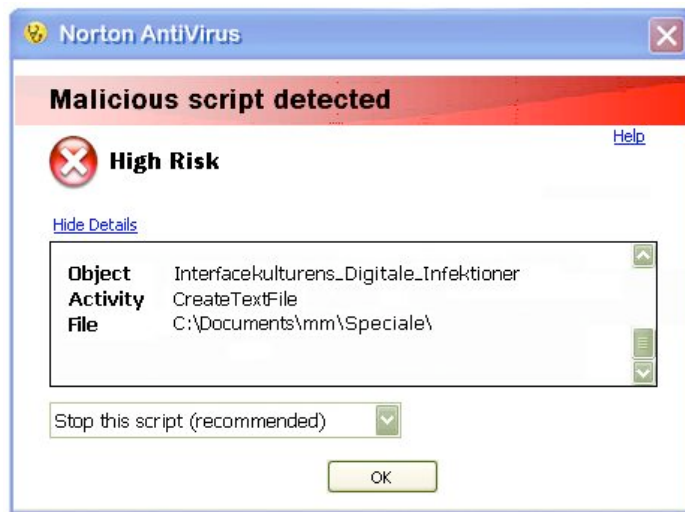
Bogen forefindes i en digital version på www.digital-aestetik.dk

ISBN 87-91810-11-6
EAN 9788791810114



Interfacekulturens digitale infektioner - Computervirus' æstetik og kultur

Speciale i Multimedier, IT-Vest, Århus Universitet af Jacob Nielsen 2008. Vejleder: Søren Pold



Indholdsfortegnelse

Indledning.....	1
Kapitel 1: Hvad er en computervirus?.....	5
Kode og effekt.....	5
Kulturelle kontekster.....	8
Gængse forestillinger.....	9
Subkultur: Vx-scenen.....	11
Kapitel 2. Den grafiske payload.....	13
Introduktion til analysen.....	13
2.1 Fejl og forhindringer.....	16
HPS: En formel intervention.....	16
Von Neumann arkitekturen.....	18
Transparens og brugervenlighed.....	19
Interfacet er en forhindring.....	21
Simulation af fejl.....	22
Fumble: Et drilsk tastatur.....	23
Balancen mellem magt og kontrol.....	24
Gullible: Et deformeret vindue.....	26
Om bruger-illusioner.....	28
2.2 Grafiske infektioner.....	29
Melting Screen: Et flydende system.....	29
Visualisering af systemets sammenbrud.....	30
Inspiration fra filmens fremstillinger?.....	31
Total simulation.....	32
Dematerialisering af software.....	34
Magistr.a: Et animeret kontroltab.....	36
Naturmetaforik og sikkerhed.....	37
Om virulent metaforik.....	39

2.3 Det fremmede i maskinen.....	40
Frygt og massehysteri.....	40
Den biologiske virus.....	41
Sulfnbk.exe: Tekst-virus.....	43
Den elektroniske graffiti.....	45
SilverRat: En suspekt gæst.....	46
Misledende grafisk repræsentation.....	47
Digitale monstre og binære aliens.....	49
Om parodi og paranoia.....	51
Kapitel 3: Computervirus-æstetik og -kultur.....	52
Æstetiske virkemidler.....	52
Computermediets effekt.....	54
Kritiske potentialer.....	56
Instrumentalisme og determinisme.....	57
Fra utopi til massekommercialisering.....	59
Interfacekritik.....	60
Digital modkultur.....	62
Modstand og aktivisme.....	63
Hacker-mentalitet og computervirus-æstetik.....	66
Kapitel 4: Net- og softwarekunsten.....	68
Software-kunst og kritik.....	68
Softwareværker: Window, Giant Cursor og WIMP.....	69
Jodi: OSS.....	71
GWEI - Google Will Eat Itself.....	72
Afslutning.....	72
Litteraturliste.....	75
Links til virusbeskrivelser.....	79

Interfacekulturens digitale infektioner

- Computervirus' æstetik og kultur

Indledning

Op igennem den digitale kunsts forholdsvis korte historie er computermediet primært blevet benyttet som et redskab i relation til værkproduktion og præsentation. I den forbindelse har det som oftest været målet at gemme den underliggende teknologi væk fra beskuerens åsyn, for i formidlingens eller sanseoplevelsens navn at fokusere på det kunstneriske udsagn værket producerer. Denne form for praksis kan ses som udtryk for fascinationen af de nye muligheder for interaktiv og multimedial repræsentation, computermediet har bragt, og den har manifesteret sig i et utal af værker i form af cd-rom udgivelser, interaktive installationer, virtual reality produktioner og meget mere. I modspil til denne tradition er der de senere år, med fremkomsten af netkunsten og senere softwarekunsten, opstået nye tendenser på den digitale kunstscene. I disse kunstretninger benyttes computermediet ikke udelukkende som et rent funktionelt *redskab*, men snarere som *materiale* i den kunstneriske produktion. I forbindelse med *Transmediale* festivalen i 2001, som var blandt de første digitale kunstfestivaler, der tematisk dedikerede sig til software-kunsten, lød det således i programerklæringen:

*"For us, software art is opposed to the notion of software as a tool (...)
Instead, software art has the potential to make us aware that digital
code is not harmless, that it is not restricted to simulations of other
tools, and that is itself a ground for creative practice."*¹

I den forbindelse kan man sige, at softwarekunstgenren omfatter værker, hvor softwareprogrammering eller konceptuelle aspekter af software spiller en vigtig rolle i værkets tilblivelse. Computerkode betragtes, dels som det æstetiske og kunstneriske materiale, men ligeledes som genstand for kulturel refleksion. Det vil sige, at der i mere eller mindre eksplicit grad fokuseres på, hvilken social, politisk og økonomisk betydning software har i den digitale kultur og informationsamfundet generelt. Dette udtrykker en erkendelse af, at efterhånden som vores hverdag i stadigt større

¹ Transmediale jury statement 2001: http://www.transmediale.de/01/en/s_juryStatement.html [02-08-2008]

udstrækning er blevet infiltreret af digital teknologi, så lever vi i en verden, hvor tekniske, ideologiske og kulturelle koder ikke længere kan adskilles fra hinanden. I dette perspektiv er software ikke kun neutrale redskaber, men ligeledes subjektive mekanismer, som må anskues og granskes i en bred samfundsmæssig kontekst. I publikationen til *Read_Me 2.3* softwarekunstfestivalen i 2003, formulerede kunstneren *Amy Alexander* den softwarekunstneriske strategi således:

*"The algorithm that generates the output is an important and subjective thing, and in commercial software, it often hides behind the veil of innocent, technological neutrality. Software artists approach the subjectivity of algorithms in different ways; some are more formal; many are more subtle. But because software art opens itself up to examination of its subjectivity (...) it can help us think about the broader software context."*²

Et af softwarekunstens potentialer bunder i, at den kan bane vejen for en kritisk undersøgelse af softwarens kodede subjektivitet, samt de formelle og indholdsmæssige rammer software, og hermed ligeledes den digitale mediekultur, er baseret på. Denne undersøgelse udføres ofte med kodede modifikationer af eksisterende software-produkter eller koncepter. Det er med andre ord en kunstnerisk praksisform, der forholder sig kritisk til den software vi omgiver os med og den kultur denne er konditioneret af.

Med den praktiske og teoretiske formulering af softwarekunsten er der i flere forbindelser draget paralleller mellem denne kunstgenre og computervirusfænomenet. En række kunstnere, har f.eks. direkte arbejdet med fænomenet i en kunstnerisk kontekst. Det mest kendte eksempel er sikkert *Biennale.py* virussen, der blev skabt i et samarbejde mellem kunstgrupperne [*epidemiC*] og *0100101110101101.org* og som blev sluppet løs på internettet i forbindelse med *Venedig biennalen* i 2001. Med *Biennale.py* projektet approprieredes computervirusbegrebet og introduceres i den kunst-institutionelle ramme. Det kunstneriske greb, som projektet benyttede sig af, handler om gennem re-kontekstualisering af et givent kulturelt fænomen at analysere og tematisere den kontekst, det oprindeligt udfolder sig i. I dette tilfælde søgte projektet at sætte fokus på den frygt og massehysteri computervirusfænomenet traditionelt har været omgivet af, samt at *afmystificere virussens aura*³ ved at gøre

² Shulgin & Goriunova, 2003, s.10

³ Beskrevet i Cornelia Sollfranks interview med kunstnergruppen 0100101110101101.ORG: "Biennale.py - The Return of The Media Hype", 2001

dens kildekode offentligt tilgængelig. Dette skete bl.a. ved at eksponere den på plakater, T-shirts, internettet og i massemedierne i forbindelse med værkets omtale.

Hvor *Biennale.py* projektet benyttede den digitale virus, som materiale i en kunstnerisk undersøgelse, så er der i andre sammenhænge foreslået et decideret slægtskab mellem computervirusfænomenet og softwarekunsten. Dette skete f.eks. i programerklæringen til tidligere nævnte *Transmediale* festival, hvor virussens til tider ondartede egenskaber benyttes som et slagkraftigt eksempel på, at software er baseret på subjektivt kodede instruktioner og intentioner, og derfor ikke per definition er hverken harmløs eller objektiv:

“Computer viruses might be seen as a critical form of software art because they make so-called users aware that digital code is virulent. Computer software is not simply a tool.”

Mens sammenligningen mellem computervirussen og softwarekunsten her er funderet i den fælles evne til at blotlægge, at software ikke bare er et formålsbestemt redskab, så rummer de digitale vira i mine øjne andre karaktertræk, som indikerer et langt tættere slægtskab mellem disse to kodede praksisformer. Det forekommer mig i den forbindelse overset, at computervira historisk set ikke alene har taget form som tekniske entiteter med ondsindede virulente intentioner, men ligeledes har været udtryk for en eksperimenterende kodepraksis, udsprunget af en virusskriverkultur med sine egne æstetiske udtryksformer. Det bemærkelsesværdige er, at disse æstetiske former ikke udelukkende udfoldes på det kodede niveau, men i mange tilfælde ligeledes manifesterer sig i den grafiske brugergrænseflade. Der findes med andre ord en kategori af vira - disse florerede primært op igennem halvfemserne - som formidler sin tilstedeværelse på den inficerede computer, via visuelle effekter og modifikationer af operativsystemets fremtoning og funktionalitet. Det er en gren af viruskreationer, der via interface-baserede virkemidler, kommunikerer til modtageren på mere spidsfindige måder, end med en decideret destruktiv eller ondsindet algoritmisk handling. De iscenesætter irrationel funktionalitet, grafiske deformationer og lignende effekter, ofte med et humoristisk kendetegn. Disse kreationers æstetiske og strategiske karaktertræk har i mine øjne en slående lighed med den værkpraksis, der senere har taget form i den softwarekunstneriske genre. I forlængelse af ovenstående påstand om at *computervira kan ses, som en kritisk form for softwarekunst*, vil jeg i denne tekst argumentere for, at denne lighed ikke kun er kendetegnet ved det *budskab* de digitale

vira bringer, men i mange tilfælde ligeledes gennem den *form* de giver sig til kende i. Med dette vil jeg hævde, at disse vira aftegner en interfaceæstetisk diskurs, som senere er blevet videreført indenfor softwarekunsten.

Omdrejningspunktet for nærværende tekst er at fremdrage og analysere omtalte vira, for her igennem at kortlægge den æstetiske diskurs de udstikker.⁴ Tilgangsvinklen er at holde fast i, at viraenes effekter ikke bare er skabt for at besværliggøre vores digitale liv og færden, de reflekterer ligeledes afsenderens syn på computerteknologien, og den kultur den er betinget af. Gennem analysen vil jeg søge at afkode, hvilke virkemidler og budskaber virus-effekterne bringer. Min påstand er, at der i disse virus initierede udsagn, kan spores både en fascination og en kritik af den computerkultur de udfoldede sig i. Det vil sige, at i lighed med de software-kunstneriske værker, så kan viraenes effekter udlægges som software reflekterende udsagn, der udføres med mediets egne midler. De tager med andre ord form, som en kritik af - eller kommentar til - softwarekulturens interne og eksterne strukturer. De adresserer softwaretekniske problemstillinger, der i et bredere perspektiv stiller spørgsmålstejn ved den digitale kulturs æstetiske, politiske og ideologiske fundament. For at sandsynliggøre denne læsning af fænomenet, vil jeg undervejs perspektivere tolkningen i den kultur og samfundsmæssige kontekst, viraene er opstået i. Dette gøres dels i forhold til virusskriverkulturen - som et modkulturelt fænomen - med udgangspunkt i interviews med virusprogrammører o.lign.⁵, og dels i forhold til den teknologiske og kulturelle kontekst aktiviteten er opstået i. Det vil sige, at der vil blive fremdraget udviklinger og tendenser i den digitale kultur, som jeg mener er relevante i forhold til analysen af pågældende viruskrivningsaktivitet.

Specialet vil falde i fire overordnede kapitler. Det første *Hvad er en computervirus?* vil søge at skabe en forståelsesramme for den følgende analyse. Jeg vil udkaste tekniske, historiske og kulturelle aspekter vedrørende computervirusfænomenet, som

⁴ De analyserede vira er opstøvet ved hjælp af virus-encyklopædier på antivirus-producenternes hjemmesider (primært Kaspersky Labs side: www.viruslist.com) eller i magasiner, bøger og lignende af og om viruskrivere. Links til tekniske beskrivelser af de vira dette speciale omtaler er vedlagt som bilag under: Links til virusbeskrivelser. Kendskabet til disse vira er tilegnet via tekniske beskrivelser, screenshots og virus-simulatorer. For sidstnævnte se følgende link, der rummer en række af disse: <http://vx.netlux.org/vx.php?id=sidx> [23-03-08]

⁵ Interviews og tekster med viruskriverne stammer dels fra virusscenens egne e-magasiner (f.eks: 40Hex, Xine, Alive, Vxtasy og Coderz) og dels fra mere traditionelle magasiner og blade, som Wired, Rolling Stones Magazine, New York Times o. lign. Viruskrivernes e-magasiner er primært hentet fra siden VX-Heavens (vx.netlux.org) og The Etext Archives (www.etext.org). Begge sider har et større arkiv af ældre magasiner relateret til virusscenen.

er relevante i forhold til den videre læsning. Det følgende kapitel *Den grafiske payload* er den analytiske del, som undersøger og afkoder en række virus effekter, virkemidler og budskaber. Tredje kapitel *Computervirus-æstetik og -kultur* vil søge at skabe en overordnet karakteristik af den optegnede computervirusæstetik, samt perspektivere praksissen i den teknologiske og kulturelle kontekst. Specialets endelige mål er at påpege den æstetiske arv, som jeg mener, samtidens digitale kunst og i særdeleshed softwarekunsten har i den skitserede del af halvfemsernes viruscreation. Afslutningsvis vil jeg derfor i fjerde kapitel understrege dette slægtskab ved at sidestille computervirussen med konkrete værkeksempler fra de net- og softwarekunstneriske diskurser.

Kapitel 1: Hvad er en computervirus?

Kode og effekt

Computervira er små parasitprogrammer, der vedhæfter sig til andre programmer med formålet at reproducere sig selv. En virus, som man i teknisk forstand traditionelt definerer den, er kendetegnet ved, at den vedhæfter en kopi af sin kode, ind i sin softwarevært. Gennem denne infektion, bliver virussens selvreplicerende kode eksekveret med værtsprogrammets kode, hvormed den potentielt kan formere og sprede sig uendeligt - på computeren, på tilknyttede datalagringsmedier og gennem netværksforbindelser til andre lokaliteter. Der findes dog også andre typer virus, som teknisk opererer anderledes. Her kan f.eks. nævnes computer-orme eller trojanske heste, som grundlæggende er selvstændige programmer og derfor ikke er afhængige af en vært for at kunne formere sig. En orm er et uafhængigt program, der spreder sig selv via sikkerhedshuller i digitale netværk, mens en trojansk hest fungerer således, at den, under dække af at være et harmløst program, lokker offeret selv til at eksekvere dets kode. Selvom disse typer programmer teknisk set adskiller sig fra den traditionelle computervirus, vil jeg i denne forbindelse bruge betegnelsen virus, som en generel betegnelse for computerprogrammer, der har viral karakter. Det vil sige programmer, der kan reproducere og sprede sig - selvstændigt eller ved hjælp af et værtsprogram.

En computervirus er som regel konstrueret således, at den indeholder følgende to programdele; dels en del, hvis formål er at sprede virussen, hvilket indebærer

mekanismer til replikation og gennemtrængning af operativsystemers og anti-virus softwarens sikkerhedssystemer, og dels en del, som udøver en given effekt på det inficerede systems tilstand. Det er oplagt at beskrive elementerne, via en reference til missil teknologi, som ”*a carrier and a warhead*”⁶. *Bærer* komponenten fører virussen frem, og dennes effektivitet og opfindsomhed definerer, hvor kontagiøs den er, hvilket vil sige, hvor hurtigt og bredt den kan sprede sig. *Sprænghovedet*, der er den anden komponent, er den påvirkning af systemet eller tilkendegivelse af virussens tilstedeværelse, som dens skaber har valgt at inkludere med bæremekanismen.

I virus-relateret jargon benyttes også betegnelsen *payload* til at beskrive de processer en given virus udfører, udover at replicere og sprede sig selv. En payload kan i de mest ondsksfulde tilfælde indebære ødelagte data, formatering af harddisken eller decideret systemsammenbrud. I andre tilfælde kan den bestå af tekstbeskeder, visuelt output eller forskellige former for manipulation af styresystemet. Nogle virustyper indeholder slet ikke nogen payload, og kan derfor helt ubemærket residere på brugerens computer i årevis, hvilket vil sige, at de udelukkende er skrevet med formålet at sprede sig selv. Modsat er det ikke ualmindeligt, at en virus indeholder flere forskellige typer payloads. I disse tilfælde kan de f.eks. bestå af en eller flere ondartede payloads, samt en række grafiske eller tekstbaserede payloads, der ofte på mere eller mindre kreativ vis henleder opmærksomheden på ophavsmanden – vel at mærke dennes dæknavn.

En given payload kan være programmeret således, at skader og lignende udføres gradvist gennem længere perioder, eller på en måde, så den forholder sig passivt og skjult over en tidsperiode, før den effektueres. Begge strategier benyttes for at give virussen større mulighed for at sprede sig, før den bliver opdaget. Disse tidsintervaller kan være ganske korte, eller i særlige tilfælde strække sig over flere år. Virustyper betegnet som *logical bombs* indeholder såkaldte *triggers*, som medfører at payload algoritmen f.eks. udløses på en bestemt dato eller efter et specifikt antal opstarter af computeren. *Logical bombs* kan være umiddelbart detonerbare, men kan også være defineret med så specifikke betingelser, at de højst sandsynligt aldrig udløses.

⁶ Ross, 1990 (ingen sidetal)

Det første registrerede tilfælde af integreringen af en payload i en virus, er formentlig Rich Skrenta's *Elk Cloner*⁷ program fra cirka 1982. Programmet inficerede Apple II computerens operativsystem og spredte sig via floppy-disk drevet til andre programmer (operativsystemet blev på dette tidspunkt indlæst fra diskette). Når den inficerede diskette havde været *loaded* mere end halvtreds gange siden infektionen, spærrede virussen adgangen til diskettens data, og der dukkede et digt op på skærmen. Digtet lød:

*"Elk Cloner. It will get on all your disks. It will infiltrate your chips.
Yes it's Cloner! It will stick to you like a fly on a glue stick. It will
modify RAM too. Send in the Cloner!"*

Virussens ophavsmand *Rich Skrenta* skrev *Elk Cloner* for at lave sjov med sine venner, men virussen spredte sig ude af kontrol, og anses for at være et af de første eksempler på en virus *in the wild*⁸.

Historisk set forekommer idéen om den destruktive payload, der ødelægger data og får systemet til at bryde sammen, som en utilsigtet og tilfældig sideeffekt fostret af - ofte dårligt programmerede - eksperimenter med selvreplicerende software. De tidligste eksempler på viruspåførte skader, var resultater af overbelastning af netværket eller den inficerede maskines processor, hukommelse og datalagringskapacitet, grundet det hastigt accelererende antal af opgaver, der skulle håndteres, når virussen spredte sig. Computeren frøs og pådrog sig sommetider betydelige skader, uden at programmøren direkte havde intenderet dette, men fordi softwaren spredte sig uforudsigeligt, effektivt og ude af kontrol. En af de første forekomster af en tilsigtet destruktiv virus(en trojansk hest), dukkede op i 1985 med virussen *EGABTR*⁹, som distribuerede sig selv via e-mail. Kamufleret som et program, der kunne forbedre computers grafiske egenskaber, lokkedes modtageren til at eksekvere programmet, hvilket resulterede i, at alle filer på offerets harddisk blev slettet og beskeden "*Arf, arf, Gotcha!*" dukkede frem på skærmen.

⁷ Beskrivelse findes her: <http://www.skrenta.com/cloner/> [28-07-2008]

⁸ Et begreb der stammer fra den ikke-kommercielle hjemmeside www.wildlist.org. Siden opdaterer løbende en liste over, hvilke vira der i øjeblikket spreder sig ukontrolleret på brugernes computere.

⁹ Beskrivelse findes her: <http://www.textfiles.com/virus/virus.txt> [28-07-2008]

Kulturelle kontekster

Computervirus begrebet blev akademisk defineret af *Fred Cohen* i 1984, med artiklen ”*Computer viruses – Theory and Experiments*”, hvor han som den første formulerede de risici der er forbundet med den digitale virus og anden selvspredende computerkode. Definitionen af en computervirus lød i *Cohen’s* artikel således:

”We define a computer 'virus' as a program that can 'infect' other programs by modifying them to include a possibly evolved copy of itself. With the infection property, a virus can spread throughout a computer system or network using the authorizations of every user using it to infect their programs. Every program that gets infected may also act as a virus and thus the infection grows.”¹⁰

Denne definition var den første, som drog analogien mellem digital selvreplicerende kode og den biologiske virus. Ideen om selvreplicerende algoritmer rækker dog længere tilbage i computerens historie, og kan spores tilbage til *John von Neumann’s* teori om *selvproducerende automata* publiceret i 1949. De første forsøg med selvreplicerende kode var da også forskningsbaserede, og deres primære formål lå i selve eksperimentet at producere dem. De kan ses som tidlige forsøg på at virkeliggøre grundidéerne bag forestillinger om intelligente maskiner og kunstigt liv.

Parallelt med *Cohen’s* definition og analogien til den biologiske virus opstod også opfattelsen af computervirusfænomenet som et teknologisk og samfundsmæssigt problem. Dette kan ses i lyset af, at der med den opkommende kommercielle udbredelse af computeren i starten af firserne opstod en ny type af virus-programmører, der ikke - som tidligere - eksperimenterede med virus i afgrænsede miljøer, men som spredte deres kreationer ukontrolleret. Disse programmører var fortrinsvis unge teenagere, og deres kreationer tog hovedsageligt form som mere eller mindre harmløse *practical jokes* rettet mod det netværk af computerspilentusiaster, som de via disketter byttede piratkopieret software med. Hvor disse tidlige virus frembringelser, på trods af deres til tider destruktive karakter, uden større besvær kunne rubriceres som drengestreger, blev fænomenet kompliceret markant med udbredelsen af internettet. Med internettet opstod muligheden for at sprede den digitale infektion bredt og effektivt, hvilket indebar, at en hel række af nye motivationer for at skabe virus opstod. Mange viruskrivere blev betaget af ideen om at kunne udbrede deres kode på verdensplan, og ud af denne fascination opstod en hel

¹⁰ Cohen, 1984 (ingen sidetal)

subkultur af inkarnerede viruskrivere, som udfordrede hinanden ud i at skabe de mest opfindsomme og raffinerede selvreplicerende viruskoder. Det var en periode, hvor der i udstrakt grad blev eksperimenteret med viruskodens mangeartede anvendelsesmuligheder i det gryende digitale netværkssamfund.

Efterhånden som de netværksbaserede digitale medier blev mere og mere infiltreret af økonomiske interesser, blev virusfænomenet det ligeledes. Virusprogrammer blev redskaber til at sprede *spam*, stjæle personlige oplysninger og til at narre penge fra de intetanende ofre. Denne tendens har udviklet sig frem til i dag, hvor gruppen af professionelle virusprogrammører, der beskæftiger sig med økonomisk kriminalitet, står bag halvfems procent af den viruskode, der bliver produceret.¹¹

Denne grovkornede gennemgang af hovedtrækkene i computervirusens historie, tjener til at underbygge min grundliggende arbejdshypotese i denne tekst, nemlig at computervira ikke udelukkende er tekniske entiteter, som er skabt for at besværliggøre vores computerbrug, de indeholder ligeledes karaktertræk, der afspejler den digitale kultur de er skabt i. Det vil sige, at de reflekterer, hvordan brugerne i et givent miljø og en given tidsperiode har brugt og opfattet computermediet. I dette perspektiv kan man sige, at de tidligere omtalte forskningsbaserede virus eksperimenter er udtryk for sin tids fascination og tro på den nye computerteknologis uanede muligheder. Teenage-programmørernes drillelystne virale udfoldelser aftegner helt præcist spilklulturens anskuelse af mediet; at inficere sin kammerats computer var sjov og ballade, på linje med at tæve ham i et computerspil. At hovedparten af samtidens virusproduktion er forbundet med økonomisk kriminalitet vidner, i dette lys, om i hvilken grad den digitale kultur er blevet inkorporeret i den kapitalistiske verdensorden – en sammensmeltning som kommer til udtryk på begge sider af loven.

Gængse forestillinger

Den altomfavnende generelle opfattelse af computervirus er dog, at den udelukkende er udtryk for en højst uetisk form for vandalisme, cyberterror eller organiseret kriminalitet. Det tages ofte for givet at vira er ondsindet af natur og at deres payload altid er skadelig. Computervirus er en trussel i det digitale informationssamfund, dens kildekode kan fungere som en et brækjern eller et våben, men samtidig kan man stille spørgsmålstegn ved, om disse egenskaber er kendetegnende for virusfænomenet af

¹¹ Ifølge viruslist: <http://www.viruslist.com/en/viruses/encyclopedia?chapter=153280553> [28-07-2008]

natur. Disse træk er vel i ligeså høj grad betegnende for selve computermediet, hvis udviklingshistorie er tæt knyttet til militære interesser og gøremål. Det er fristende at parafasere Ani DiFranco's kendte sætning "*Every tool is a weapon if you hold it right.*"¹² med "*Every code is a weapon if you program it right.*", for hermed at understrege, at en virus udelukkende har skadelige effekter, hvis dens skaber har valgt at inkludere slige i dets kode. Ser man nærmere på den mængde af virus, der gennem historien er produceret, vil man opdage, at det kun er et fåtal af dem, der indeholder direkte destruktive payloads¹³. Digitale vira har gennem tiderne bragt mange forskelligartede effekter og budskaber, og der findes sågar direkte venligtsindede vira, som er karakteriseret ved, at de søger at eliminere ondsindede digitale infektioner og udbedre sikkerhedshuller i operativsystemet¹⁴.

Som medieteoretikeren *Jussi Parikka* har uddybet med sin omfattende genealogiske kortlægning af computervirussens kulturelle historie, er billedet af virusfænomenet del af en kompleks samling af forestillinger og interesser, der har udfoldet sig gennem dets historie.¹⁵ Virus kan historisk set læses i mange forskellige diskurser, og der har ligget et utal af disparate motiver til grund for at skabe dem. At ondsindet software - også kaldet *malware* - i dag er til stor gene for det brede lag af computerbrugere, skal der ikke herske tvivl om, men alligevel kan man stille spørgsmålstegn ved, om den generelt unuancerede anskuelse af virusfænomenet, som ondsindet aktivitet udelukkende udført af sociopater eller hårdkogte kriminelle bander, er berettiget. *Fred Cohen*, som siden starten af firserne bl.a. har forsket i computervirussens gavnlige potentialer, giver en forklaring på den negative opfattelse af virusfænomenet.

*"(...) simple explanations are commonly used to avoid having to talk about the great breadth of issues involved in this field. It's a lot easier to sell fear when you can claim all Indians are evil than when you have to explain the difference between a Shawnee and a Mohawk. Another reason is that most people aren't very interested in mathematics or being very precise in what they do. Why bother to fully understand when you don't have to."*¹⁶

¹² Ani DiFranco er folk-sanger og frasen er en sangtitel.

¹³ I denne tekst har jeg valgt at definere en destruktiv payload som et stykke kode, der uigenkaldeligt sletter eller ødelægger brugerens data. Dvs. at en virus, der f.eks. modificerer operativsystemets grafiske arkitektur ikke falder under denne definition, fordi skaden kan udbedres ved at slette virussen eller geninstallere systemet. Definitionen kan selvfølgelig diskuteres, men er her trukket af praktiske hensyn.

¹⁴ De såkaldte "beneficial viruses". Se evt. Julian Dibbels "Viruses are good for you", 1995

¹⁵ I sin nylige bog "Digital Contagions – A Media Archaeology of Computer Viruses" fra 2007

¹⁶ Fra Alive e-mag. Stojakovic-Celustka, 1994 (ingen sidetal)

Ifølge *Cohen* er den gængse opfattelse, af hvad en virus er, fostret af den almene computerbrugers ignorance i forhold til dens tekniske aspekter, samt en generelt forsimplet anskuelse af fænomenet. Sidstnævnte er, ifølge *Cohen*, i høj grad fremavlet af antivirus-industriens kommercielle fremfærd og retorik. Det kan da også konstateres, at frygten for den digitale infektion kan omsættes til kolde kontanter i form af antivirus-pakkeløsninger. Fremkomsten af nye maliciøse virustilfælde, afføder som regel et heftigt opsving i den digitale sikkerhedsindustriens omsætning, hvilket f.eks. var tilfældet med *Melissa* ormen, som i 1999 spredte skræk og rædsel på nettet og i massemediernes overskrifter, for efterfølgende at øge antivirus-producenternes samlede salg af software med syvogtres procent i løbet af en uge¹⁷. Der er hermed ingen tvivl om, at antivirus-industrien kan skabe økonomisk vinding ved at pleje virushysteriet i den brede offentlighed. De fleste computerbrugere ved teknisk set så lidt om computervirus, at det er fristende for antivirus-industrien at hengive sig til dyster overdrivelse.

I den forbindelse spiller massemediernes ikke nogen neutral rolle. Når truslen om en virus med ekstra stærk medietække dukker op, iværksætter antivirus-selskaberne gerne et kapløb ud i at udsende virusvarsler og fange massemediernes opmærksomhed. Dette har ofte resulteret i, at nyheden om en særligt udspekuleret virus ofte har spredt sig hurtigere og bredere i massemediernes overskrifter end på ofrenes computere. Det mest oplagte eksempel på en virus, som via sit nærmest konceptuelt funderede modsætningsforhold, mellem dens umiddelbare kærligheds-erklæring og dens destruktive egenskaber, formåede stjernestatus i massemediernes, er *Loveletter* virussen (også kaldet *I Love You*). En virus, som frem til i dag, formentlig er den mest kendte computervirus i den brede offentlighed.

Subkultur: Vx-scenen

Op igennem halvfemserne eksploderede mængden af digitale virusforekomster. I 1989 blev det estimeret, at der eksisterede omkring halvfems forskellige Pc-vira, mens tallet steg til over fem tusind i 1995¹⁸. Forklaringen på denne stigning kan findes i, at udbredelsen af internettet gav muligheden for, at virusprogrammører verden over kunne organisere sig i netværk og udveksle erfaringer om deres aktiviteter. Der

¹⁷ Beskrevet her: <http://www.macobserver.com/news/99/april/990414/virusalesup.html>

¹⁸ Parikka 2007, s.299-300

opstod med andre ord ret så velorganiserede interessefællesskaber blandt computervirusentusiasterne, som i denne periode skabte indbyrdes platforme for kommunikation og udveksling – sammenfattet under betegnelsen Vx-scenen (virus exchange). Scenens indbyrdes kommunikation foregik primært via chat-kanaler og e-mail distribuerede magasiner. Virusskriverne organiserede sig i grupper og deres medlemmer var ofte bosiddende i vidt forskellige dele af verden. Skriverne skjulte sig bag aliasser og kendte sjældent hinandens virkelige identiteter. Udover at skabe viruskode, omfattede disse gruppers aktiviteter ligeledes produktion af magasiner med interviews, artikler og ikke mindst *tutorials* om virusprogrammering. Dette indebar, at kildekode blev byttet som spillekort via internettet og benyttet til at skabe nye virusmutationer. Virusskriverne gjorde deres arbejde og erfaring tilgængelig via deres mere eller mindre hemmelige netværk, hvilket ligeledes resulterede i at mange nye aktører involverede sig i miljøet. At være Vx'er kunne omfatte en række aktiviteter udover at skrive kode. Dette kunne bl.a. være at administrere virusfora, at formidle og studere virus, eller at være samlere af viruskode (*zoo-keeper*). I løbet af halvfemserne blev den kraftigt ekspanderende Vx-scene en subkultur med sine egne netværksbaserede kanaler for distribution, kommunikation og formidling - og ikke mindst med sine egne helte, idealer og regler.

De spilleregler, Vx-scenen opererede ud fra, var tæt knyttet til *hacker-etikken*¹⁹, som baserer sig på principper om åbenhed, decentralisering, mistillid til autoriteter og forestillingen om computerteknologien som platform for demokratisk udveksling og kommunikation. Desuden forankrer *hacker-etikken* sig i idéen om, at al information bør være fri og tilgængelig, uanset dens potentielt skadelige anvendelsesmuligheder. Dette var ligeledes devisen, hvad angik viruskildekode. I forlængelse af dette må det samtidig påpeges, at der indenfor Vx-miljøet syntes at være veldefinerede retningslinjer for adfærd. Det var officielt ikke velanset at kompilere destruktiv kode og sprede den ukontrolleret via internettet. Selvom scenens aktører eksperimenterede med alle aspekter og anvendelsesmuligheder af selvreplicerende kode, fralagde de sig samtidig ethvert ansvar for eksterne personers (mis)brug af deres kode og arbejde. Mange af scenens kreationer florerede da heller ikke *in the wild*, men udelukkende i virussamlinger og indenfor Vx-scenens egne rammer.

¹⁹ Den oprindelige hacker-etik blev formuleret af Steven Levy i bogen "Hackers: Heroes of the computer revolution" fra 1984.

Disse virusskriveres motivation synes i høj grad at være funderet i selve fascinationen af viruskodens muligheder²⁰. Mange af scenens aktører betragtede deres virale udskejelser som en forskningsorienteret hobby eller som en måde at udtrykke sig kreativt. Indenfor scenens rammer var der en stærk selvjustits; aktørerne måtte gøre sig fortjent til status og respekt indenfor miljøet, og det krævede arbejde og engagement at blive optaget i en af de mere - internt - prestigefyldte virus-skrivergrupper. Denne symbolske kapital kunne udelukkende erhverves ved at skrive elegant, effektiv og opfindsom viruskode. Endelig kunne der opnås agtelse indenfor miljøet ved at tjene fællesskabet på andre måder, f.eks. ved at teste eller raffinere andres viruskreationer, ved at bidrage til scenens aktiviteter via magasinskrivning eller ved vedligeholdelse af virusrelaterede internetfora og lignende.

Kapitel 2. Den grafiske payload

Introduktion til analysen

Læser man Vx-scenens magasiner og *tutorials*, vil man erfare, at viruskrivernes primære interesse er virussens bæremekanismer og de teknikker, det kræver at skabe effektivt selvreplicerende kode, som kan gennembryde eller overlister operativ- og antivirusystemernes sikkerhedsforanstaltninger. Payload effekter betragtes som en biting i virusproduktionen og uddybes sjældent nærmere i Vx-publikationerne. Alligevel kan man konstatere, at payloadens karaktertræk har udviklet sig gennem tiderne. De tidligste vira, som spredte sig *in the wild*, indeholdt typisk en – tilsigtet eller utilsigtet – skadevoldende payload og/eller en tekstbesked, der gjorde opmærksom på virussens tilstedeværelse. Der blev på dette tidspunkt ikke investeret den store opfindsomhed i at kreere payloads. F.eks. kan man nævne at *Dark Avenger*, som var ophavsmanden til nogle af de allerførste vira, der formåede at masseudbrede sig, under et interview svarede følgende på spørgsmålet om, hvorfor han implementerede en destruktiv payload i sin første virus: ”*As for the first virus, the truth is that I didn't know what else to put in it.*”²¹. Efterhånden fandt viruskriverne dog på langt mere opfindsomme sager at putte i deres payloads. De kunne f.eks.

²⁰ For viruskrivernes egne beskrivelser af scenen, samt motivationerne for at skabe virus se f.eks. tekster af: MidNyte 1999, Spanska 2000, Asmodeus 2000, DecimatoR 1992, Lord Jules 1999 og n0ph 1999.

²¹ Gordon 1993 (ingen sidetal)

benyttes til at udbrede tekstbaserede beskeder, i form af f.eks. vittigheder, skadefro kommentarer, smædebeskeder eller politiske budskaber.²² Med viruskode kunne man nå en bred skare af mennesker uden om officielle informationskanaler, hvilket blev udnyttet på et utal af måder.

Sideløbende med dette opstod der dog ligeledes en gren af vira, hvis payload-effekter og tilgrundliggende motivationer er langt sværere at gennemskue og kategorisere. Antivirus-producenten *Kaspersky Labs* skriver i deres online computervirus historie således:

“The first malicious programs may have shocked users, by causing computers to behave in unexpected ways. However, the viruses which started appearing in the 1990s present much more of a threat: they are often used to steal confidential information such as bank account details and passwords.”²³

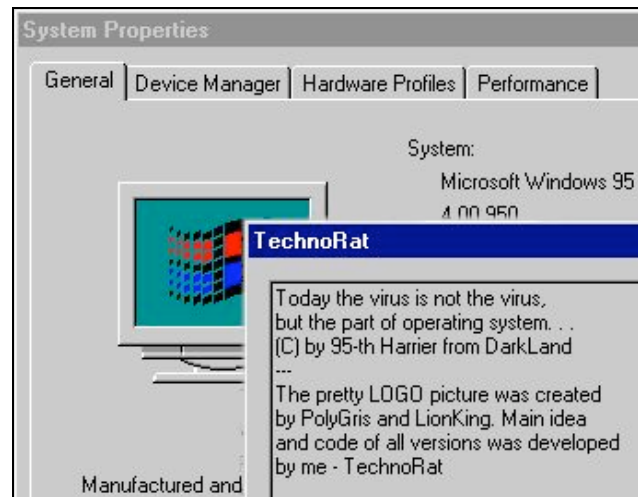
Omdrejningspunktet for dette kapitel er denne gren af vira, som *chokerede brugerne ved at få computeren til at opføre sig på uforudsigelige måder*. Fokuset vil blive rettet på en række vira, der er kendetegnet ved, at de sjældent ødelægger data, stjæler informationer eller udtrykker noget erklæret mål eller budskab. Deres payloads kan sikkert opfattes som sideeffekter implementeret for at give kreationerne et særpræg, der markerede deres tilstedeværelse og skiller dem ud fra mængden af virusforekomster. Denne type effekter – som betegnes *grafiske payloads* - er karakteriseret ved, at de udfolder sig i den grafiske brugergrænseflade og/eller via modifikationer af systemets funktionalitet. De iværksætter ofte irrationel computeropførsel eller grafiske effekter med et ”humoristisk” præg.

De tidligste forekomster af grafiske payloads manifesterede sig i firsernes kommandolinjebaserede interfaces. Her kan f.eks. nævnes *Australian Parasite* virussen, der vendte skærmens tekst på hovedet eller *Green Caterpillar* virussen, som viste en lille grafisk animeret larve - skabt ud af ASCII tegn - der kravlede hen over skærmen, mens den undervejs ”spiste” interfacets skrifttegn. Det var dog først med masseudbredelsen af operativsystemet med den grafiske brugergrænseflade(GUI), at der for alvor blev eksperimenteret med grafiske payloads. Med disse operativsystemer

²² Smædebeskederne var typisk rettet Microsoft, antivirusproducenter eller andre virusskrivere. Af vira med direkte politiske budskaber kan f.eks. nævnes *Mawanella* og *Staple-a*, som gjorde opmærksom på militære overgreb på civile i henholdsvis Sri Lanka og Palæstina, eller *Zafi-or*men, hvis tekst kritiserede regeringen i Ungarn.

²³ Findes her: <http://www.viruslist.com/en/viruses/encyclopedia?chapter=153280684>

fik virusskriverne en ny platform for deres viruskreation. Denne platform gav dels nye udfordringer i udforskningen af fejl, mangler og sikkerhedshuller i systemet, og dels en række nye muligheder for at udtrykke sig via den grafiske brugergrænseflade. Disse aspekter blev udforsket til fulde. De digitale vira blev, som *TechnoRat* virussen kækt proklamerer²⁴, en del af operativsystemet og udnyttede de muligheder og svagheder, dette introducerede.



Virus inficeret operativsystem. Af TechnoRat(1995)

Hvor udfordringen tidligere primært havde været at skrive kode, der kunne sprede sig effektivt, blev den nu, ligeledes at bryde ind i og modificere operativsystemets grundlæggende struktur og fremtoning. At skrive virus blev en kontekstuel praksis, som forholdt sig til platformen og de muligheder og begrænsninger denne indebar. Hvor virusprogrammering traditionelt opfattes som en aktivitet, der handler om at bryde ind i tilfældige menneskers computere, fik den på dette tidspunkt snarere karakter af at være en praksis, der handlede om at bryde ind i den kommercielle software, for her igennem at udfordre dennes rammer og konventioner, og i den forbindelse var det nærmest udelukkende Windows operativsystemet, der stod for skud. Ovenstående skift er, som jeg senere vil uddybe, centralt i analysen af den grafiske payloads æstetik og virkemidler.

Dette analytiske afsnit vil undersøge en række af de grafiske payloads, der dukkede op med de GUI-baserede operativsystemer. Min grundantagelse er, at disse vira ikke udelukkende er tekniske entiteter, som er udviklet for at besværliggøre vores

²⁴ TechnoRat virussens grafiske payload med beskeden "Today the virus is not the virus, but part of the operating system..." var integreret i menuen med Windows 95's systemindstillinger. (se billedet ovenfor)

digitale liv og færden, de udtrykker ligeledes karaktertræk, der afspejler og fortæller noget om den digitale kultur. I det indledende kapitel var jeg inde på, at virusproduktion historisk set udspringer fra vidt forskellige kulturer, der f.eks. er forankret i felter, som forskning, computerspilkultur, økonomisk kriminalitet osv. De vira, jeg følgende vil fokusere på, er formentlig alle skabt af VX-scenens aktører²⁵. Det vedkommende spørgsmål er så, hvad denne subkulturs virale udskejelser udtrykker i lyset af den kulturelle kontekst, den var omgivet af.

I den forbindelse er de grafiske payloads interessante, idet de giver sig til kende for computerbrugeren, og på denne måde frembyder en kommunikation, som kan fortolkes og analyseres. Min tese er, at disse payload-effekter, ikke blot er betydningstomme eller tilfældige frembringelser, der alene er skabt for at genere computerbrugerne, de bringer ligeledes budskaber, som reflekterer viruskrivernes syn på teknologien og den digitale kultur. De bærer implicit budskaber, som foldes ud via deres manipulation af computerens interfacebårne fremtoning og funktionalitet. De benytter virkemidler, som henvender sig direkte til modtageren af viruskoden. Målet med denne analyse er at afkode denne kommunikation og specificere det sprog, de grafiske payloads benytter.

Analysen vil falde i tre dele, som hver især repræsenterer en strategi, hvormed virussens tilstedeværelse tilkendes gives for computerbrugeren. Disse afsnit skal ikke opfattes som repræsentanter for klart adskilte virustyper, men er primært defineret for analysens overskueligheds skyld. Endelig skal det nævnes, at hovedparten af de vira, der her vil blive analyseret, florerede i midthalvfemserne. Den grafiske payload dukkede op i starten af firserne, havde sin glansperiode i halvfemserne, for følgende at forsvinde igen – fra internettet og *in the wild* lister - omkring årtusindskiftet. Årsagen til dette vil jeg komme ind på senere i denne tekst, men lad os indtil videre slå fast, at fænomenet allerede nu er blevet en del af computervirussens historie.

²⁵ Denne formodning kan for de fleste af analyseobjekterne eftervises, idet mange af disse vira er beskrevet i VX-scenens e-magasiner og tutorials.

2.1 Fejl og forhindringer

HPS: En formel intervention

*HPS-virussens*²⁶ triggerrutine aktiveres, hvis computeren inficeres på en lørdag. I så fald eksekveres en payload-algoritme, der gør, at de bitmap-billedfiler brugeren eller systemet åbner spejlvendes horisontalt. De oprindelige billeder overskrives på harddisken med de modificerede versioner, som ligeledes bliver stemplet med et kodet "vandmærke", således at virussen kan genkende dem og ikke spejlvender dem tilbage til normal tilstand. Da operativsystemet jævnligt benytter bitmap-filer i den grafiske brugergrænseflade, resulterer dette i en række besynderlige effekter, idet mange af de grafiske elementer, der udgør operativsystemets visuelle arkitektur - herunder opstarts- og nedluknings-skærbillederne - bliver spejlvendt. *HPS* modificerer ligeledes udseendet af andre af computerens programmer og de bitmap-filer, brugeren åbner i billedbehandlingsprogrammer og lignende, undergår samme behandling. *HPS* gør ikke opmærksom på sig selv eller virussens ophavsmand på andre måder og har ingen decideret destruktiv payload.



HPS af GriYo

HPS kan via sin payload betragtes som en formel intervention i systemets grafiske brugergrænseflade. Den trænger, via et simpelt billedmanipulerende greb, ind i computerens repræsentationer af funktionalitet og data, og i denne proces skelnes der ikke mellem, hvilke pixels der stammer fra software-arkitekturens og brugerens

²⁶ Navnet er taget fra den biologiske virus Hantavirus Pulmonary Syndrome(HPS), som overføres til mennesker, der har været i kontakt med gnavere. HPS kan give akut vejrtrækningbesvær.

domæne. Alle billeder, der er repræsenteret i bitmap-formatet, undergår samme modifikation.

Von Neumann arkitekturen

Det er i den forbindelse relevant at fremhæve, at et aspekt, der kendetegner GUI-baserede operativsystemer og slutbrugersoftware generelt, er, at de skaber en klar adskillelse mellem den fastlåste software-arkitektur og de manipulerbare data. Softwarens funktionalitet er fast defineret, mens de data brugeren behandler med den, er redigerbar og fleksibel på alle mulige leder og kanter. Ser man på, hvordan den moderne Pc teknisk set er bygget op via integreringen af *Von Neumann arkitekturen*²⁷, som er en implementering af *Turings universalitetsprincip*²⁸, så er denne adskillelse mellem redskab og materiale ikke kendetegnende for maskinens grundlæggende arkitektur. Det, der karakteriserer *Von Neumann* arkitekturen, er tværtimod, at maskinen er konstrueret, således at data og instruktioner lagres og processeres i samme fysiske domæne. Det vil sige, at både programmer og de data, programmerne behandler, gennemløber de samme elektroniske kredsløb i computeren og er repræsenteret via den samme sekvens af *tændt* og *slukket* tilstande, i den strøm af binær-information maskinen holder og behandler. Med *Von Neumann arkitekturen* er computersoftwarens ikke fastlåst fra producentens side - som det er tilfældet i andre mikroprocessorer f.eks. i en lommeregner - men åben for modifikation. Det er netop dette aspekt, der gør computeren sårbar over for funktionsfejl og virusangreb; det er denne mulighed for, at et program kan skabe eller modificere andre programmer, som en virus udnytter, når den inficerer og modificerer computerens software. Samtidig er det vigtigt at påpege, at det er *Von Neumann arkitekturens* egenskaber, der giver computermediet sit store teknologiske potentiale i forhold til tidligere medier og redskaber, fordi den via sin abstrakte fysiske arkitektur kan simulere enhver anden symbolmanipulerende maskine og oven i købet tilvejebringe funktionalitet, som ikke findes i noget andet medie eller redskab. Computerens software-arkitektur kan derfor principielt set formes på samme måde, som det materiale brugeren behandler med den, fordi dens funktionalitet er symbolsk defineret og hermed ikke indbygget i den

²⁷ Von Neumann Arkitekturen udsprang af matematikeren John Von Neumanns rapport om EDVAC-systemet: "First Draft of a Report on the EDVAC." fra 1945.

²⁸ Alans Turings teori om den Universielle Turing-maskine skitserer ideen om en abstrakt maskine, som kan læse en kodet beskrivelse af enhver anden symbolmanipulerende maskine, hvormed den, trods sin enkle opbygning, kan udføre ikke-trivielle beregninger og simulere den oprindelige maskines funktionalitet og logik. Turing 1950, s. 49-65

fysisk fastlåste arkitektur. Sagt på en anden måde, der er ikke nogen invariant grænse mellem maskinen og materialet der behandles²⁹.

Det er dette principielle fravær af en invariant grænse mellem maskine og materiale, *HPS* teknisk set blotlægger via sin intervention i systemet. Virussen manipulerer de elementer, der udgør softwarens visuelle arkitektur, som var de digitale billeder, der udsættes for *flip-funktionen* i et billedbehandlingsprogram. På denne måde afspejler virussen, med et simpelt algoritmisk greb, *Von Neumann arkitekturens* underliggende logik. Den viser, at software er symbolsk defineret og i princippet kan manipuleres på fuldstændig samme måde som det materiale, vi behandler med den. Hvad infektionens offer nok primært hæfter sig ved er dog, at virusens modifikation af den grafiske brugergrænseflade i høj grad saboterer systemets velkomponerede arkitektur og brugervenlighed. Hvad der gør denne intervention ganske spidsfindig, er den måde den forholder sig til den grafiske brugergrænseflades grundliggende logik. For at forstå dette aspekt, må vi tage et blik på nogle af de konventioner, der ligger bag moderne interfacedesign.

Transparens og brugervenlighed

Den grafiske brugergrænseflade har en tradition for at være metaforisk. Det vil sige, at man ved brug af metaforer forsøger at knytte interfacet til vores erfaringer fra den virkelige verden, for på denne måde at gøre interaktionen mellem menneske og maskine lettere forståelig. Moderne operativsystemer som *Windows* og *MacOS* er grundlæggende bygget op omkring skrivebordsmetaforen, der repræsenterer ideen om, at computerens data organiseres visuelt, som dokumenter og redskaber på et skrivebord.

De grundlæggende principper bag skrivebordsmetaforen, som designstrategi også kaldet WIMP-paradigmet (windows, icons, menus, pointers), blev udviklet op gennem 70'erne på *Xerox PARC* og var inspireret af tidlige eksperimenter inden for design- og forskningsdisciplinen HCI (human-computer-interaction). HCI beskæftiger sig grundlæggende med interaktion mellem computer og menneske, og den fokuserer på, hvordan man gennem kombinationen af fysiske input/output enheder og metaforiske grafiske grænseflader kan konceptualisere struktureringen af data på den mest umiddelbare og lettest forståelige måde. Den logik, der ligger bag HCI, og som nutidens styresystemer fundamentalt set er baseret på, bunder i forestillingen om, at

²⁹ Som formuleret af Finneman, 1998, s.49

computeren skal fungere som en naturlig augmentation af menneskets krop og intellekt. De vigtigste inspirationskilder i udviklingen af WIMP paradigmet var *Ivan Sutherlands Sketchpad*³⁰ system, der introducerede *direkte manipulation* af grafiske objekter, og *Douglas Engelbarts* præsentation af *NLS*³¹ projektet i 1968, hvor han demonstrerede grundelementerne til mange af de fysiske og grafiske interfaces, vi benytter idag – f.eks. computermusen, tastaturet og vinduer på skærmen.

Hvor tidligere tiders kommandolinje-baserede operativsystemer i høj grad krævede teknisk tilegnelse, indebærer metaforiseringen af interfacet muligheden for en langt mere umiddelbar og brugervenlig interaktion med computeren. Ved at lade brugergrænsefladens elementer og opbygning referere til allerede eksisterende genstande, arbejdsgange og medier, baserer man systemet på velkendte erindrings- og associationsmønstre. Den metaforiske brugergrænseflade hjælper på denne måde brugerne til på forhånd at forestille sig computerens funktionalitet, hvormed muligheden for en langt højere grad af intuitiv interaktion skabes.

Man siger, at efterhånden som den funktionalitet, metaforerne repræsenterer, vejer tungere i brugerens opfattelse af interfacet, end deres direkte refererende egenskaber gør, vil deres oprindelige nytteværdi forsvinde³². Det betyder, at referencerne opløser sig selv, til fordel for brugerens fortolkning af *dét computeren gør*. Indenfor HCI har interfacet da også traditionelt været anskuet som et mellemlid mellem maskine og menneske, som nødvendigvis må fremstå så anonymt og usynligt som muligt. Dette bygger på opfattelsen af, at *interfacet er en forhindring*³³, hvis materialitet må fortrænges til fordel for computerens funktionalitet og brugbarhed. Den moderne brugergrænseflades opbygning og formsprog reflekterer følgende ideen om en naturlig og transparent interaktion, som giver brugeren en fornemmelse af direkte umedieret adgang til de informationer, der præsenteres og bearbejdes. Interfacet er i den forbindelse designet til at skjule sig selv for brugeren, som gennem en naturlig augmentation kan interagere frit med arbejdsopgaven.

Det er relevant at nævne, at ideen om computeren som et transparent medie har spillet en central rolle i computerens historie i kommercielle henseender. Den massive udbredelse af hjemmecomputeren i halvfemserne blev i høj grad båret frem

³⁰ Sutherland, 1963, s. 109-127

³¹ Engelbart & English, 1968 (ingen sidetal)

³² Fuller, 2007, s. 81

³³ Usability-forskeren Donald Norman formulerede ”An interface is an obstacle”. Videre-citeret fra Pold & Bertelsen, 2003, s.98

af disse egenskaber og forestillinger. Samtidig har markedsføringsstrategien for nye interfacebaserede medier ofte bestået i at fremstille disse som udtryk for en endnu uset grad af brugs- og sansemæssig umiddelbarhed. Indenfor operativsystemsindustrien har dette en lang tradition, som specielt Apple har ført an, men som *Microsoft* nærmest per automatik har fulgt trop på. Her kan nævnes alt fra lanceringen af *Apple's* første GUI-baserede *Macintosh* computer i 1984, under anti-kommandolinje-sloganet ”*the computer for the rest of us*” til den seneste version af *MacOS Leopard*, som i høj grad markedsføres på de nye tredimensionelle navigationsmuligheder, operativsystemet indeholder.

Interfacet er en forhindring

Vender vi tilbage til *HPS-virussen*, kan man slutte følgende; da man stadig kan interagere normalt med computeren, selvom dele af operativsystemets grafiske arkitektur er spejlvendt, beskadiges systemet ikke på et direkte funktionelt plan. Brugerens interaktion med computeren forstyrres derimod på et perceptuelt plan. Interfacets elementer er stadig velkendte, men i deres spejlvendte udgave bliver de unaturlige, træder frem og stikker i øjnene. Den grafisk manipulerede brugergrænseflade nedbryder systemets velkomponerede fremtoning og bliver fragmenteret og irriterende. Brugerens arbejde med computeren besværliggøres, fordi interfacets elementer bliver så opmærksomhedskrævende i deres på en gang fremmede og velkendte fremtoning, at softwarens transparente egenskaber nedbrydes. Interfacet bliver, om ikke en forhindring, så en modspiller i brugerens interaktion med computeren.

Systemets brugervenlighed undermineres gennem denne dekonstruktion af interfacets overflade og transparens, idet brugerens fokus tvinges mod selve mediet, frem for de data der behandles med det. Virussen benytter med andre ord den modsatte strategi i forhold til de designprincipper, den grafiske brugergrænseflade er baseret på. Interventionen kan i dette lys opfattes som en kritik rettet mod specifikke tendenser i tidens interfacedesign; den tager sig ud som et nærmest ikonoklastisk angreb på brugergrænsefladens transparente facader. I et interfacekritisk perspektiv er det altså ikke en umotiveret handling, når *HPS* smadrer softwarens transparens. Angrebet på systemets visuelle arkitektur tager sig snarere ud som en ret så udspekuleret kritik af softwareindustriens hang til brugervenlige overflader. Da det netop er overskridelsen af softwarearkitekturens fast definerede grænser, der

resulterer i en afmontering af softwarens transparens, er det en ganske konceptuel intervention *HPS* udfører. Det konceptuelle ligger i, at den via ét og samme kodede greb simultant afspejler computerens (*Von Neumann arkitekturens*) *underliggende* logik og nedbryder interfacets *overliggende* logik. På denne måde opstår der et dikotomisk spil mellem computermediets indre og ydre beskaffenhed – mellem *det synlige* og *det usynlige*.

Opfatter man *HPS* som udtryk for en virusbåren interfacekritik, må man samtidig slutte, at den ikke eksplicit giver sig selv til kende som dette. Virussen lader ikke offeret vide, at computerens besynderlige opførsel er resultatet af en digital infektion, og modifikationen af den grafiske brugergrænseflade vil for de fleste formentlig blive opfattet som en systemfejl. De følgende analyser vil fokusere på virus, som via deres payloads mere eksplicit giver sig selv til kende, men da algoritmiske iscenesættelser af fejl er meget udbredt i forbindelse med payload-effekter, er det væsentligt indledningsvis at se nærmere på dette fænomen.

Simulation af fejl

Der har gennem tiderne ofte floreret rygter om digitale vira forekomster, der kan ødelægge brugernes hardware eller forårsage uoprettelig skade på computerens fysiske enheder. I den forbindelse kan det påpeges, at det generelt er meget svært at forårsage skade på hardware ved hjælp af software³⁴. Computere er designet således, at dette i princippet ikke kan lade sig gøre. Det er derimod muligt at forårsage skade på computerens BIOS³⁵, hvilket i mange tilfælde fejlagtigt bliver betragtet, som hardwarefejl. Eksempler på vira der er blevet opfattet således er *Magistr.a* og *Chernobyl*. Skader på computerens BIOS bliver sikkert forvekslet med hardware destruktion, fordi de kan indebære, at computeren har problemer med at starte op. Skade af denne art kan dog oftest repareres med BIOS-gendannelses software.

De eneste tilfælde, hvor virus reelt kan ødelægge hardware, er i forbindelse med produktionsfejl i specifikke hardware-komponenter. I den forbindelse har der været tilfælde, hvor vira kan benytte fejlen, til at udøve skade på hardware³⁶. Dette

³⁴ En undtagelse er dog, at man gennem længere tids virus-dirigeret overbelastning af en hardware komponent kan slide denne ned - f.eks. en harddisk.

³⁵ BIOS(Basic Input/Output System) En BIOS består af softwarekode, som er lagret på en chip i computeren. Dennes funktion er at genkende computerens hardware, således at computeren er forberedt til opstart og installation af programmer, hvilket først og fremmest vil sige operativsystemet.

³⁶ Mark Ludwig beskriver et tilfælde, hvor en monitor havde en produktionsfejl, der gjorde at den kunne ødelægges via software manipulation. Ludwig, 1995, s.548

forekommer dog yderst sjældent og kræver ligeledes, at programmøren på forhånd kender denne fejl og hermed specifikt kan adressere den i sin viruskode. Dette er måske forekommet i enkelte tilfælde, men generelt må man betegne hardware-destruerende viraføremster, som en myte.

Rygtene om ødelagt hardware kan derimod være afstedkommet af, at der findes en række vira, som simulerer hardwarefejl. Det vil sige, at de via deres kode iscenesætter fejl, som narrer brugeren til at tro, at en given hardwarekomponent er defekt. En strategi, der ifølge *Mark Ludwig*, er ligeså effektiv som at udøve reel skade på hardware:

Simulating hardware damage can be every bit as effective as actually damaging it. To the unwary user, simulated damage will never be seen for what it is (...) Furthermore, just about any hardware problem can be simulated.”³⁷

Ludwig giver eksempler på software-simulerede hardwarefejl, hvilket kan være at få computermonitoren til at opføre sig besynderligt, ved f.eks. at sende illegale værdier til grafikortet eller ved at programmere et kontinuerligt loop af skiftende skærmindstillinger, hvormed fornemmelsen af en defekt flimrende skærm opnås. Et andet eksempel er simulation af tastaturnedslidning, hvor registreringen af tryk på en eller flere af tastaturets knapper sorteres fra. Hyppigheden af denne frasortering kan eskaleres over en længere periode, således at defekten tager sig ud, som et resultat af tastaturnedslidning.

Ovenstående metoder til fejlsimulering skaber realistiske iscenesættelser af hardware/software defekter, hvis konstruerede karakter er nærmest umulige at gennemskue. Man kan sige, at hvad disse eksempler krystalklart demonstrerer, er, at ligeså vel som vi kan manipulere computeren gennem dets grænseflader, så kan computeren også manipulere med os gennem dem - vel at mærke langt ud over interfacets fysiske rækkevidde og i sidstnævnte tilfælde sikkert hele vejen ned i butikken efter et nyt tastatur.

Fumble: Et drilsk tastatur

I forhold til disse realistiske simulationer af hardwarefejl, findes der også tilfælde af vira, som benytter mere tvetydige virkemidler i deres iscenesættelser af fejl. Her er

³⁷ Ibid. s.549

det, i forlængelse af tidligere nævnte tastaturnedslidningsstrategi, oplagt at nævne *Fumble*³⁸ virussen. Denne simulerer, at computerbrugeren laver tastefejl ved at erstatte det tastede bogstav med det, der sidder ved siden af på tastaturet. Brugeren får fornemmelsen af at have ramt forkert på tastaturet, hvilket vil sige, at virussen skaber indtrykket af, at fejlen udføres af brugeren selv og ikke computeren. Dette aspekt gør *Fumble* virussen interessant, fordi den - i dette øjeblik forvirring – formår at få brugeren til at tvivle på sin egen fysiske ageren. Ved virussens fortløbende simulation af de fumlede tastefejl, vil brugeren formentlig ligeledes stille spørgsmål til sin egen dømmekraft; ”*Ramte jeg forkert, eller er tastaturet defekt?*” Ved nærmere undersøgelse vil det, på grund af tastaturdefektens temmelig utroværdige karakter, med stor sandsynlighed blive klart for offeret, at fejlen er simuleret.

Det, der adskiller *Fumble* fra tidligere nævnte metoder til fejlsimulation, er, at den antyder den tekniske defekts konstruerede karakter. Den iscenesætter en fejl, der er så fjollet og usandsynlig, at selv en bruger med minimal teknisk indsigt, kan gennemskue, at den er menneskeskabt. Hermed afslører virussen ligeledes sin egen tilstedeværelse og den effekt, den udøver på systemet. Der er med andre ord tale om, at virussen benytter virkemidler, der henvender sig til modtageren af viruskoden; den er designet til at give sig selv til kende og kommunikere til modtageren via dens manipulation af systemet.

Balancen mellem magt og kontrol

En computervirusinfektion indebærer grundlæggende et kontroltab. Når viruskoden påtvinger sig adgang til systemet uden brugerens accept eller vidende, fratvinges offeret en del af sin kontrol. Dette tab kan foregå, uden at brugeren er bevidst om det, idet virussens tilstedeværelse er skjult under brugergrænsefladens niveau. Denne egenskab af at være usynlig er meget nyttig, hvis afsenderens motivation eksempelvis er at samle personlige informationer eller lignende – hvilket ofte er tilfældet i forbindelse med cyber-kriminelle aktiviteter. I modsætning til denne strategi, findes der også en gren af vira, som via deres payloads meget eksplicit formidler den magt, virussen har tilegnet sig over maskinen; at slette offerets harddisk eller simulere et systemsammenbrud på realistisk vis, er magtdemonstrationer, som ikke er til at tage fejl af.

³⁸ *Fumble* er en tidlig DOS baseret virus, hvis payload kort nævnes og beskrives teknisk i Ludwig, 1995, s. 551.

Fumble virussens intervention i systemet er ikke på samme måde hverken en skjult eller en kontant magtdemonstration. Den er snarere en magtforskydning eller en forstyrrelse af brugerens kontrol over computeren. Brugeren kan jo i princippet tilegne sig tastaturets nye logik og på den måde genvinde kontrollen over maskinen. Man kan sige, at *Fumble* virussen iværksætter et erkendelsesforløb, hvor offeret bevæger sig fra at have en tro på, at tastaturdefekten er ægte, til at indse at den er fingeret. Brugeren tvivlen på sine egne evner til at ramme de rigtige taster, er samtidig bundet op på tiltroen til systemets stabilitet og rationalitet. Med erkendelsen af at situationen er algoritmisk iscenesat, forstyrres disse forestillinger, og offerets opmærksomhed rettes mod brugergrænsefladens irrationelle respons. I denne proces bliver det tydeligt, at grænsefladen ikke udelukkende er en neutral anordning, men derimod en mekanisme, der fortolker vores fysiske ageren, og at reglerne for denne fortolkning er konstrueret og ikke af natur gennemskuelig og rationel. Det greb, *Fumble* benytter, fremhæver et vigtigt kendetegn ved brugergrænsefladens natur, nemlig at den indebærer en magtfordeling. *Brenda Laurel* skriver i bogen ”*The Art of Human-Computer Interface Design*”:

“An interface is a contact surface. It reflects the physical properties of the interactors, the functions to be performed, and the balance of power and control.”³⁹

Som *Laurel* påpeger, indebærer computerens fysiske og grafiske grænseflader grundliggende en magtbalance mellem menneske og maskine. Man kan sige, at denne fordeling af magten forhandles i softwarens designproces og defineres i dens implementering. I forlængelse af *Laurels* definition kan man slutte, at en *Fumble* inficeret computer ikke længere afspejler brugerens fysiske egenskaber, hvilket indebærer, at den magtbalance, computerens software konstituerer, er blevet forskubbet. Denne magtforskydning fremhæver, hvad moderne interfaces sjældent giver til kende, nemlig selve kendsgerningen, at de konstituerer et magtforhold mellem menneske og maskine. Hvor softwareprodukter generelt søger at give brugeren en fornemmelse af fuldstændig kontrol, demonstrerer *Fumble*, at brugeren kun har den kontrol, som softwarens design og struktur giver adgang til. *Fumble* virussens indgreb peger hermed på, hvordan brugerens kontrol over computeren altid

³⁹ Laurel, 1990, s. xii

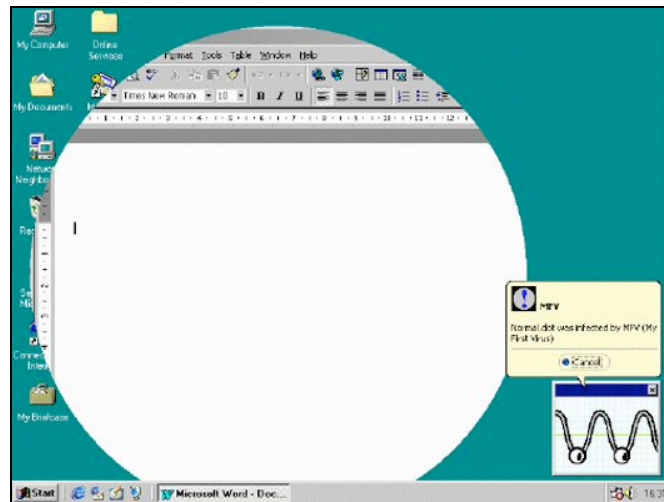
er iscenesat, fordi man ikke interagerer direkte med maskinens binære mekanismer, men med repræsentationer af dem. Det bliver konkret, at der foregår noget på den anden side af grænsefladen, som brugeren ikke har kontrol over, og som adskiller computeren fundamentalt fra den skrivemaskine, den simulerer via tastaturet. At *Fumble* virussen oven i købet kan få brugeren til at tvivle på sine egne fysiske egenskaber understreger ligeledes, hvilken magt computeren har over brugeren, når denne tilegner sig systemets logik og lader sig inkorporere i dets modeller for interaktion og erkendelse.

At en grænseflade involverer en magtfordeling, kan forekomme en smule banalt, når man taler om tastaturer, problemstillingen bliver der imod langt mere kompleks, når man taler om grafiske brugergrænseflader. Disse giver os muligheden for at udføre yderst avancerede former for interaktion, men er samtidig kendetegnet ved at være særdeles komplicerede konstruktioner, som kun giver et glimt af, hvad der foregår bag skærmen. Det næste eksempel, jeg vil fremdrage, forstyrrer ligeledes brugerens kontrol over computeren, men i dette tilfælde udøves dette kontroltab via den grafiske brugergrænseflade.

Gullible: Et deformeret vindue

Gullible er en makrosript-virus, som inficerer *Microsoft Word 97* dokumenter. Viruskoden eksekveres, når et inficeret *Word-dokument* åbnes, hermed lagres virussen i programmets globale *template*, hvilket betyder, at alle dokumenter, der åbnes eller skabes i *Word-programmet*, inficeres. Den 19. marts hvert år aktiveres virussens payload, som består i, at tekstbehandlingsprogrammets vindue visuelt beskæres, således at det tager form som en cirkel. Efter dette må *Word* tvangslukkes via operativsystemets nedlukningsfunktion, eftersom den del af programmets menu-bar, som indeholder nedlukningsfunktionen, ikke er synlig og tilgængelig.

Gullible virussen manifesterer sig via en fjollet effekt; en akavet omdefinering af programmets form og logik. Den giver sig, i lighed med *Fumble* virussen, til kende ved at iscenesætte en højst usandsynlig ”fejl”, hvis konstruerede karakter uden større besvær kan afkodes af infektionens offer. Da *Gullible* deformerer *Word-programmets* ellers fasttømrede visuelle arkitektur, kan indgrebet læses, som en kommentar til den overordnede brug af skrivemaskine-logik i forbindelse med tekstbehandlingsprogrammer. Den designmæssige idé, der ligger i at simulere en skrivemaskine



Gullible af MFV

ved hjælp af en computer, der i bund og grund er en langt mere avanceret maskine, trakteres med en subversiv ommøbning af softwarens visuelle arkitektur. Den ”firkantede” papirmetafor gøres rund - i konkret og symbolsk forstand, hvormed den strukturelle fleksibilitet computermediet, principielt set, rummer frembydes. Dette greb kan opfattes, som en komisk kommentar til software instrumentalisering og de begrænsninger, der latent ligger i at forankre et givent softwaredesigns funktionalitet og fremtoning i et andet redskabs kendetegn og egenskaber.

Et andet vigtigt aspekt er, at med deformationen af tekstbehandlingsprogrammet, nedbrydes den grafiske brugergrænseflades transparens, idet den metafordrevne følelse af at arbejde med et velkendt redskab forstyrres. Programmets nye form irriterer brugerens fordybelse i mediet, fordi den bryder softwareredskabets illusioner og gør opmærksom sin egen repræsentationelle karakter. I modsætning til tidligere omtalte *HPS-virus* indebærer *Gullible* virusens modifikation af det repræsentationelle lag, samtidig en direkte afmontering af dele af det funktionelle lag. Store dele af programmets funktionalitet er ikke tilgængelig, fordi den er ”skåret” væk fra skærmens overflade og hermed brugerens rækkevidde. Interfacet bliver helt konkret en forhindring i arbejdet med computeren. Dette påpeger et karaktertræk ved interfacet, som *Lev Manovich* formulerer således:

”Rather than being a neutral medium of presenting information, the screen is aggressive. It functions to filter, to screen out, to take over, rendering nonexistent whatever is outside the frame.”⁴⁰

⁴⁰ Manovich, 2001, s. 96

Denne formidlingsmæssige aggressivitet, som skærmen ifølge *Manovich* repræsenterer, fremhæves ved *Gullible-virusens* deformation af grænsefladen. Interfacets evne til at gøre det, der er udenfor grænsefladens synlige rækkevidde ikke-eksisterende, tydeliggøres, idet den afskærer brugeren fra funktionalitet, der burde være på skærmen. Med denne grafiske barrikade tydeliggøres det, at grænsefladen er et filter, som alene giver os de muligheder og erkendelse - eller den kontrol og magt - det gør tilgængelig for os.

Om bruger-illusioner

Som opsummering og afslutning på dette afsnit vil jeg konkludere, at der i disse virus-initierede interventioner ligger et kritisk potentiale, idet de baner vejen for en refleksion over, hvad en grænseflade grundlæggende er. I den forbindelse kan man pege på en udtalelse, som *Alan Kay*, en af nøglepersonerne i udviklingen af den personlige computer, fremførte i relation til arbejdet med at designe grafiske brugergrænseflader:

”At PARC we coined the phrase ‘user illusion’ to describe what we were about when designing user interfaces. There are clear connotations to the stage, theatrics, and magic”⁴¹

Grafiske grænseflader er helt grundlæggende funderet på at skabe illusioner. De er designet til at give os fornemmelsen af, at vi interagerer noget andet end den række af *tændt og slukket* tilstande maskinen holder og behandler. De digitale vira kan fungere som særdeles overbevisende spillere på denne ”scene af bruger-illusioner”, hvilket deres evne til at skabe realistiske fejlsimulationer vidner om. Alligevel synes de eksempler, der i dette afsnit er blevet behandlet, snarere at søge at nedbryde brugergrænsefladens illusioner, frem for at hengive sig til dem. Hvor disse viras formelle forskydninger af interfacets form og logik nærmest tager sig ud som ”trylleri” indenfor operativsystemets fast definerede rammer, så er det samtidig tricks, der helt fundamentalt provokerer den måde, vi ser og forstår computeren. De udfordrer gængse forestillinger om computerens beskaffenhed. Hvor de masse-udbredte softwareprodukter fundamentalt set har lært os at opfatte computermediet som et fornuftsbestemt redskab til transparent informationsbehandling, så fremhæver disse viruspåførte mutationer, at software ikke af natur er rationel, gennemskuelig

⁴¹ Kay, 1990, s. 199

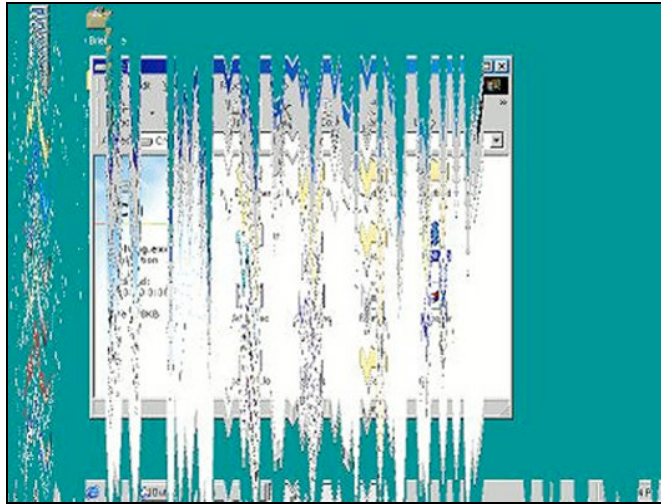
eller formålsbestemt. Samtidig minder de os om, at interfacet, på trods af de transparens forestillinger, der florerer omkring det, grundlæggende betjener sig af medieringer; at gennemsigtheden i bund og grund er en konstruktion, sat i scene via metaforer og algoritmiske rutiner.

2.2 Grafiske infektioner

Hvor foregående afsnit har koncentreret sig om payloads, der bryder interfacets form og logik, så er dette afsnit dedikeret til vira, som formelt set benytter den diametralt modsatte strategi. Følgende analysedel vil fokusere på vira, der, via deres grafiske payloads, formidler deres tilstedeværelse eller gøremål meget konsekvent indenfor de forståelsesrammer, computerens interfaces konstituerer. Da de analyseobjekter, jeg vil fokusere på, udfolder sig i den grafiske brugergrænseflade, kan man med andre ord sige, at de kommunikerer i kraft af den visuelle logik, disse flader benytter sig af.

Melting Screen: Et flydende system

Melting Screen er en blanding mellem en traditionel virus og en orm. Den ankommer til offerets computer som en fil, der er vedhæftet en e-mail besked. Hvis denne fil eksekveres, spredes virussen ved, at den henter adresser fra *Microsoft Outlook* for efterfølgende at videresende sig selv til disse. Virussen indeholder dels en grafisk payload og dels en payload, der består i, at den omdøber alle eksekverbare filer (.exe) til at have endelsen *.bin* (endelsen for komprimerede datapakker), hvilket resulterer i, at systemet ikke kan afkode og eksekvere filerne. Den grafiske *payload* er tidsindstillet efter et tilfældighedsprincip og manifesterer sig ved, at den grafiske brugergrænseflade, med de elementer den måtte indeholde på det givne tidspunkt, ”smeltes” visuelt. Efter at interfacets elementer er opløst - via en animeret sekvens - og gledet ud af skærmens ramme, fastlåses operativsystemet, således at man ikke længere kan komme i kontakt med computeren. Det er dog muligt at genstarte computeren, hvormed den opfører sig normalt, indtil den næste nedsmeltning indtræffer.



Melting Screen

Visualisering af systemets sammenbrud

Virusinfektionen illustreres ved et visuelt forfald; en for computerbrugeren synlig nedsmeltning af interfacets grafiske elementer. Virussen tilstedeværelse tilkendegives via en grafisk payload med metaforisk reference til smeltende materiale, og symboliserer på denne måde tanken om, at offerets kontrol og data opløses og forsvinder, som en isblok, der smelter i solen. Man kan opfatte det således, at virusprogrammøren iscenesætter grænsefladen som en skrøbelig fysisk størrelse, hvis materialitet markerer systemets sårbarhed overfor ydre påvirkning - i dette tilfælde i forhold til virusinfektionen.

En reel fejl eller dysfunktionalitet i computerkode vil sjældent manifestere sig visuelt i interfacet og i tilfælde, hvor dette sker, er det yderst usandsynligt, at defekten tager form som en metafor, der refererer til og kan afkodes i relation til den virkelige verden. Den visuelle manifestation af *Melting Screen* virussen er da heller ikke en naturlig følgevirkning af den ”spolering” af maskinens data, der ligeledes udføres. Den grafiske og den ”destruktive” payload er programmeret i separate algoritmer og har ikke nogen direkte forbindelse til hinandens gøremål. Det eneste, de har tilfælles, er, at de er kompileret ned i den samme kodepakke. Der er i stedet tale om, at virusprogrammøren, ved hjælp af den grafiske payload, iværksætter en visuel effekt, som henvender sig til brugerens forestilling om, hvordan data-destruktion og operativsystemets kollaps tager sig ud på skærmen. Virussens grafiske manifestation henvender sig, med andre ord, til en gængs forestilling om, at manipulation af data på maskinkodens niveau ligeledes afføder en synlig virkning på interfacets niveau.

Grænsefladen betragtes, i denne sammenhæng, som uadskillelig fra systemet på en måde, der relationelt indebærer, at interfacet præsenterer en direkte kausal sammenhæng i forhold til, hvilke processer maskinen udfører. Men hvor stammer sådanne forestillinger fra?

Inspiration fra filmens fremstillinger?

Denne type forestillinger er vidt repræsenteret i filmens fremstillinger af computermediet. Når computeren indgår i filmens univers - det vil i denne forbindelse sige, når computerens interface repræsenteres på filmlærredet - er fremgangsmåden for det meste, at computerens arbejdsopgave tydeliggøres via en visuel udpensling af den proces, der udføres. Søges der i en database, f.eks. efter dna-profiler eller fingeraftryk, som man jævnligt oplever det i kriminalfilm, illustreres den automatiserede søgningen ofte ved, at databaseelementerne gennembladres visuelt på computerskærmen. Dette afspejler en logik, der tilsyneladende indebærer, at databasens elementer må frem på skærmen, for at de algoritmiske processer - i dette tilfælde databasesøgningen og dennes sammenligningen af elementer - kan gennemføres. Denne type visualisering er selvfølgelig afstedkommet af, at computerens funktion og arbejde skal gøres forståelig gennem det ekstra lag af visuel mediering, som computerinterfacets repræsentation i filmmediet udgør. Der er tale om, at interfacets formelle logik skal tilpasses filmens repræsentationelle logik, men i denne transformation sker der for beskueren ligeledes en perceptuel sammensmeltning af computerens processuelle og repræsentationelle planer.

Et computervirusrelateret eksempel på denne type sammensmeltning kan findes i filmen *The Net* fra 1995⁴². Det væsentlige er i den forbindelse, at hovedpersonen i filmens klimaks slipper en virus løs, for på denne måde at destruere nogle belastende data, der er blevet påført hendes digitale identitet. Det interessante er måden, hvorpå denne virus manifesterer sig på computerskærmen – eller måske mere præcist på filmlærredet. Virusinfektionen illustreres som noget, der ligner et organisk forfald af interfacets grafiske informationer. Filmen skildrer, gennem den årsagsmæssige sammenhæng mellem destruktions af data og interfacets grafiske opløsning, en nærmest ”organisk forbindelse” mellem det underliggende system og brugergrænsefladen. Denne fremstilling bruger implicit referencen til den biologiske

⁴² Instrueret af Irwin Winkler. Som en sidebemærkning kan det nævnes, at filmen er fra samme år, som Melting Screen virussen dukkede op.

virus og det fysiske forfald. Etableringen af denne reference, og den symbiotiske forbindelse mellem maskinen og dens grafiske interface, fungerer som en effektiv katalysator i filmens formidling af virussens destruktive gøremål og systemets sammenbrud.

Man kan konstatere, at mange af Vx-scenens aktører har været inspireret af filmens skildringer af virus- og hackerfænomenet, f.eks. refererer mange af deres aliaser til cyberpunk-fiktion. Det er da heller ikke usandsynligt, at den visuelle formidling, som *Melting Screen* og en række andre vira benytter sig af, er inspireret af filmens virkemidler. Men, som jeg efterfølgende vil uddybe, er denne type årsagsbestemt visualisering af computerens tilstand og gøremål, ligeledes dybt integreret i softwareindustriens egne slutbrugerorienterede interfacedesigns.

Total simulation

Undervejs i udviklingen af den personlige computer frem til, hvad vi i dag betegner som multimediecomputeren, udviklede der sig, hvad man kan betegne som en realistisk diskurs i forbindelse med computerens grafiske interfacedesign. Jagten på den metafordrevne brugervenlighed indebar, at computerens kode og egenskab af at være en maskine - i modsætning til et redskab - i stadig højere grad blev skjult bag interfacets facader. Med denne tilsløring af maskinens grundlæggende mekanismer blev konceptionen af computeren først og fremmest defineret af, hvad brugeren så og sansede på skærmen. *Steven Johnson* skriver i relation til denne udviklingsmæssige tendens følgende i bogen *”Interface Culture”*:

”Somewhere along the way the good faith of user-friendly metaphors had been replaced by the hysteria of total simulation. The reasonable desire for analogies between the digital and the organic had given way to an all-encompassing quest for the fusion of the two.”⁴³

Transparensmetaforikken og den udbredte fascinationen af de virtuelle verdener, computermediet bragte, manifesterede sig ofte, for både producenter og brugere, i højest utopiske forestillinger om en fuldstændig sammensmeltning af det virkelige og det virtuelle. Skellet mellem det organiske og det digitale måtte nedbrydes til fordel for den immersive interaktion med computeren. Disse interfacerealistiske tendenser er, trods sine knap så utopiske nutidige manifestationer, stadig meget udbredte. Der

⁴³ Johnson, 1997, s.60

går på mange måder en lige linje fra halvfemsernes massive interesse for den *totale simulation*, som den f.eks. blev praktiseret med *virtual reality* teknologien, til den nylige lancering af den tredimensionelle skrivebordsmetafor med *MacOS Leopard*. De baserer sig begge på at skabe fornemmelsen af direkte umedieret adgang til det materiale eller den verden, computeren repræsenterer via sine brugergrænseflader.

Hvor det nok vil være en overdrivelse at påstå, at den personlige computers interfaces stringent baserer sig på idéen om *total simulation*, i hvert fald samme grad som f.eks. *virtual reality* teknologien, så er det alligevel forestillinger, der er dybt forankret i den grafiske brugergrænseflades grundlæggende designprincipper. Designstrategier rodfæstet i *direkte manipulation*, *WYSIWYG (What You See Is What You Get)*, *naturlighed og transparens*, baserer sig i bund og grund på at skabe følelsen af at interagere direkte med det medierede materiale. Som en sideeffekt fostrer disse designstrategier ligeledes fornemmelsen af en kausal *årsags og virknings* relation mellem grænsefladerne og det underliggende system.

Det er, som tidligere nævnt, denne relation *Melting Screen* baserer sin visuelle formidling på. Symptomerne på den ”syge” computer manifesterer sig via en organisk metafor, som eksplicit adresserer forestillingen om den fuldstændige fusion af det organiske og det digitale. Virussen kommunikerer med sin visuelle nedsmeltning indenfor den *totale simulations* forståelsesrammer. På samme tid forekommer dette greb ret så omvæltende, idet virussens letbegribelige – eller måske ligefrem brugervenlige – visualisering, anskueliggør en uventet destruktiv handling. Den brugervenlige metaforik benyttes til at formidle et ret så antagonistisk foretagende.

Virusprogrammører opfatter generelt sig selv, som højt avancerede computerbrugere, der har opnået et større indblik i maskinens sprog og processer, end de brugere, der udelukkende interagerer med computeren via den grafiske brugergrænseflade. Derfor kan denne type visualisering af virusinfektion opfattes som en sarkastisk bemærkning til lægmandsbrugers ”overfladiske” forestillinger om, hvordan en computer fungerer. Det er med andre ord ikke uden en vis ironi, at virusprogrammøren lader offerets skærbillede smelte ned. Det er en effekt, som netop kommunikerer på den uerfarne brugers niveau, men som samtidig bliver en humoristisk kommentar for den teknisk garvede bruger. Den smeltende grænseflade fremstiller computerens sammenbrud, men reelt ødelægger virussen ikke systemet, den omdøber blot nogle

filnavne, hvilket er en irriterende omstændighed, men ikke en fatal konsekvens for hverken systemet eller brugerens data. Den teknisk rutinerede bruger kan med andre ord forholdsvis hurtigt bringe sin computer tilbage til normal tilstand. Følgelig er det oplagt at opfatte nedsmeltningen som udtryk for en indforstået viruskriverhumor, som i høj grad spiller på den almene computerbrugers sparsomme viden om, hvordan computeren fungerer under den grafiske brugergrænseflades illusoriske facader.

Dematerialisering af software

Den centrale metode i softwareindustriens kultivering af brugervenligheden er implementeringen af det metaforiske interface, hvor den instinktive interaktion med computeren baseres på genkendelighed og association. Denne metaforisering foregår både på mikroplan, i form af ikoner og vinduer med mere, og ligeledes på makroplan, hvor hele softwarepakkeløsningers udseende og funktionalitet baseres på medieringer af fysiske redskabers fremtoning og egenskaber. Typiske eksempler på dette er skrivemaskinemetaphoren i tekstbehandlingsprogrammer, båndoptagermetaphoren i medieafspillere eller skrivebordsmetaphoren i operativsystemer.

Der er med disse designstrategier ofte tale om en direkte instrumentaliserings af software, idet dennes muligheder og begrænsninger defineres via det medierede artefakts fysiske karaktertræk og funktionalitet. Dette indebærer i mange tilfælde, at brugerens muligheder for at modificere systemet, er begrænset til *preferences* palettens valg af farver, lyde, baggrundsbilleder osv. Disse softwarepakkeløsninger tilbyder, med andre ord, primært rent overfladiske muligheder for ommøblering af systemets arkitektur. Muligheder, som folder sig ud i et fast defineret skema, nøje tilrettelagt af softwarens afsender⁴⁴. Følgelig er denne type software tænkt som et redskab, brugeren skal benytte til et bestemt prædefineret formål og ikke som middel til at udvikle nye anvendelsesmuligheder. Software skal for slutbrugeren *læses*, men ikke *skrives*, som *Florian Cramer* konkluderer om de kommercielle pakkeløsninger⁴⁵. Dette indebærer ifølge *Cramer*:

⁴⁴ Pold, 2008, s. 219

⁴⁵ Cramer refererer her til Roland Barthes læsbar/skrivbar terminologi. Roland Barthes skelner i essyet "S/Z" mellem skrivbare og læsbare tekster. Den skrivbare tekst tilskynder en aktiv læsning, der er åben for fortolkning, mens den læsbare tekst på forhånd har tilrettelagt et læse- og forståelsesforløb for læseren. Cramer, 2003, s. 100

”To be readerly, popular PC user software creates the illusion of being hardware visually and tactically disguising itself as solid analog tools.”⁴⁶

Man kan sige, at selv om f.eks. *Windows* operativsystemet ikke simulerer det fysiske skrivebords arbejdsgange og funktion fuldstændigt mimetisk - systemet tilføjer med andre ord nye mediespecifikke elementer og egenskaber og bliver på mange måder en selvstændig størrelse i brugerens bevidsthed - så simulerer det alligevel overordnet følelsen af, at man arbejder med et *solidt analogt redskab*. Brugeren får fornemmelsen af at interagere med en fast defineret anordning, hvis grundlæggende struktur ikke lader sig manipulere eller redefinere via interaktionen i den grafiske brugergrænseflade. Operativsystemet repræsenterer på den måde en solid og stabil størrelse i brugerens bevidsthed; en størrelse, som opfører sig ligeså forudsigeligt, som det fysiske skrivebord.

Ser man *Melting Screen* virussens grafiske payload i lyset af, hvordan software visuelt og taktisk skaber illusionen af at være solide analoge redskaber, tager den visuelle nedsmeltning sig ud som en dematerialisering af dette ”fysiske” materiale. Operativsystemets velkendte fremtoning og stabile softwarearkitektur bliver med detonationen af den grafiske payload, flydende og u håndgribelig.

Hvor en digital virusinfektion - teknisk set - grundlæggende, med sin modifikation af programkode og vedhæftning af den parasitære algoritme, vidner om, at Pc’ens software ikke er en fastlåst størrelse, er dette aspekt ikke videre gennemskueligt for GUI-brugeren. Virussens selvreplicerende algoritmer eksekveres under den grafiske brugergrænseflades overflader og giver sig ikke uden videre til kende. Derfor kan man sige, at virussens grafiske payload fungerer, som en visuel formidling af dette aspekt. Nedsmeltning af den grafiske brugergrænseflade illustrerer billedligt set, at den *binære-notation*, software er baseret på, er en fleksibel - eller ”flydende” - størrelse, som ikke partout er defineret af forudbestemt funktionalitet, som den metaforiske software giver os indtrykket af. Ved at lade operativsystemets fastlåste grafiske arkitektur blive profilløs, får beskueren et glimt af, at software ikke er begrænset til at være et færdigpakket redskab, men ligeledes kan være en åben og manipulerbar ”substans”.

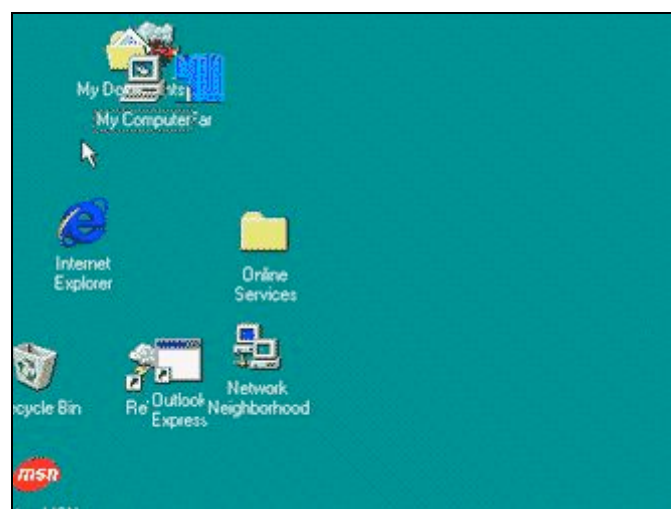
På denne måde kan man tolke *Melting Screen* virussens symbolske dematerialisering af interfacets solide grafiske elementer som en kommentar til den

⁴⁶ Ibid. s. 101

udbredte instrumentalisering af software. Idéen om den *solide analoge* software travesteres, idet virussen omdefinerer operativsystemets simulerede materialitet til en anden fysisk størrelse, hvis eneste egenskab og funktion er, at den kan smelte. Samtidig indebærer denne metaforiske transformation et kontroltab for brugeren, idet softwarens nye ”fysiske” materialitet er genstandsløs og u håndgribelig – ergo ikke interagerbar. Det næste eksempel, jeg vil se på, bruger ligeledes den grafiske brugergrænseflade til at dirigere og visualisere et kontroltab.

Magistr.a: Et animeret kontroltab

Magistr.a er en e-mail orm, hvis grafiske payload kommer til udtryk gennem en manipulation af *Windows* styresystemet. Når virussen via en tidsindstillet *trigger*-mekanisme udløser sin payload, resulterer det i, at brugeren ikke kan få adgang til skrivebordets ikoner. Dette sker ved, at virussen visuelt flytter ikonerne væk fra muse-cursoren, når den peger på disse. Dette giver en oplevelse af, at ikonerne forsøger at flygte fra cursoren. Virussen har ligeledes en direkte ondsindet payload, som udløses en måned efter infektionen, og resulterer i, at CMOS, Flash hukommelse og harddiskdata bliver slettet. *Magistr.a* virussen er med andre ord en yderst destruktiv virus, som da også formåede at sprede skræk og rædsel, da den dukkede op i 2001.



Magistr.a af Judges Disemboweler

Med *Magistr.a* virussens payload visualiseres den digitale infektion. Denne konkrete formidling fungerer samtidig som et varsel om den destruktive gerning der ligger i vente. Brugers kontroltab manifesterer sig, via de undvigende ikoner, som en

iscenesættelse af, at systemet har sin egen vilje. Ikonerne animeres grafisk, hvilket giver fornemmelsen af, at de er selvstændige ”væsener”, der ikke vil fanges, manipuleres eller eksekveres. Kontroltabet er med andre ord formuleret via den grafiske simulations retorik.

Sammenligner man den type payloads, som blev beskrevet i afsnittet *fejl og forhindringer*, med *Magistr.a* virussens indgreb, kan man sige, at de har det træk til fælles, at de begge forstyrrer brugergrænsefladens transparens. Offeret bliver i begge tilfælde opmærksom på selve mediet frem for den arbejdsopgave eller de data, det repræsenterer. Men hvor tidligere eksempler iværksætter dette skift med en dekonstruktion af softwarens form og forståelsesrammer, gør *Magistr.a* det i kraft af disse. Brugeren bliver opmærksom på grænsefladen, fordi den opfører sig uforudsigeligt og modvilligt, men samtidig er disse træk formuleret i forhold til - og inden for - systemets kommunikative rammer, idet modvilligheden udformes metaforisk og som tidligere nævnt med reference til undvigende ”levende skabninger”. For brugeren bliver introduktionen af det modvillige dog et nervepirrende brud med operativsystemets overordnede logik, da denne står i skærende kontrast til opfattelsen af, at systemet automatisk og velvilligt tjener brugerens ønsker.

Naturmetaforik og sikkerhed

Den metafordrevne ”naturlighed” som kommerciel software ofte er indkapslet i, kan, udover at fungere som en katalysator for den intuitive interaktion, udlægges til at tjene et andet formål; nemlig at overvinde brugerens angst for det eksekverbare. Med computermediet har vi ikke fingrene nede i det materiale, der bearbejdes, men med repræsentationer af det. En konsekvens af dette er, at softwareproducenterne må skjule computerens egenskab af at være en kodet maskine - der udfører et utal af automatiserede processer, vi ikke er bevidste om - i naturlighed, rationalitet og brugervenlighed. Dette er en forudsætning for, at brugerne føler, det er dem, der kontrollerer maskinen og ikke omvendt. Angsten for det eksekverbare er på mange måder det samme som angsten for at miste kontrollen over maskinen. Man kan sige, at da operativsystemet er udgangspunktet for vores digitale færden i computeren og netværket, må det tage sig ud som den velkendte ”havn”, der skaber den fornemmelse af kontrol og sikkerhed, der er forudsætningen for, at brugeren vil involvere sig i computerbrugen.

Betragter man f.eks. *Windows* operativsystemet, kan man sætte spørgsmålstegn ved, hvad logikken bag at lade standard skrivebordsbilledet⁴⁷ være et åbent landskab med grønt græs og åben horisont. Svaret kan være, at det er en iscenesættelse af systemet, der dels underbygger forestillingen om mediet, som et vindue til verden og dels giver fornemmelsen af en immanent naturlighed i systemet. I tråd med sidstnævnte forklaring bliver landskabet del af en selvscenesættelse, som underbygger forestillingen om computeren, som en neutral og harmløs anordning. At standard skrivebordsbilledet gengiver et stykke natur, er i dette perspektiv ikke udelukkende et rent æstetisk valg fra producentens side, men ligeledes udtryk for en bevidst designmetodologi, der benyttes til at influere brugerens reception af operativsystemet. Landskabet er en af ingredienserne i et overordnet design, der søger at give brugeren en fornemmelse af stabilitet og sikkerhed, således at interaktionen med computeren føles, ligeså tryk og forudsigelig, som at gå en tur ud over heden.

Operativsystemets naturlighed er da også blevet parodieret i en hel række vira. Her kan nævnes *Muhamor* virussen, der placerer en lille fluesvamp i hjørnet af himlen på *Windows 95*'s nedlukningsskærmbillede, for på denne måde - via en naturlig metafor - at understrege den "giftige" indtrængen i systemet.



Muhamor af XEOM

Et andet eksempel er *Nople* virussen, som lader græs vokse hen over operativsystemets grafiske brugergrænseflade og efterfølgende skriver teksten: ” *Es hora de formatea tu disco* ” (det er tid til at formatere din harddisk), for hermed at aftegne et

⁴⁷ Her refereres til Windows XP/NT. Windows 95/98 benyttede lignende visuel retorik med deres billeder af åben himmel i opstarts/nedluknings-skærmbillederne.

rationale, der tilsyneladende indebærer at offeret nu må formatere sin harddisk, for at fordrive den vildtvoksende natur fra sit system. Sidstnævnte virus, som i øvrigt ikke sletter harddisken eller indeholder nogen direkte destruktiv payload, parodierer naturmetaforikken, samtidig med at den med sin tekstbesked på sarkastisk vis kommenterer det ultimative middel mange computerbrugere tyr til for at fordrive en virus (og ofte anden uforståelig computeropførsel). At formatere sin harddisk og geninstallere dens software har traditionelt været en udbredt løsning, når antivirusprogrammerne ikke kan få computeren til at opføre sig som forventet.

Om virulent metaforik

Omdrejningspunktet for dette afsnit har været vira, der formidler deres tilstedeværelse og effekt, i overensstemmelse med de forståelsesrammer de metaforiske brugergrænseflader danner. De producerer visuelle symptomer på den digitale infektion via organiske og naturlige metaforer, som kan afkodes direkte i den grafiske kontekst, operativsystemet udgør. Samtidig fylder de, på kættersk vis, disse udsagn med indhold og intentioner, som brugeren normalt ikke får præsenteret indenfor operativsystemets formelle rammer. Mens idéen om softwarens neutralitet vanligtvis er metaforisk frembudt i den grafiske brugergrænseflades facader, repræsenterer disse viras metaforik tværtimod virulente karaktertræk, som modvillighed, forfald og sammenbrud, hvilket fremstår ukendt og rystende i softwarens naturlige og uskyldrene overflader. Med disse grafiske indgreb, som kiler sig ind og bliver en del af operativsystemets arkitektur og struktur, slås skår i den - nærmest billeddyrkende - tro på softwarens fornuftsbestemte og harmløse natur. Dette gøres ikke uden ironi, idet de destruktive foretagender og intentioner de metaforiske skildringer varsler som regel ikke iværksættes i realiteten (ud af de eksempler der indtil videre er behandlet i denne tekst, er det kun *Magistr.a* der indeholder en direkte destruktiv payload). De udtrykker, med andre ord, en udpræget sarkasme, idet de på paradoksal vis frembyder det *brugerfjendtlige* via brugergrænsefladens *brugervenligt* formidlende metaforik. Man kan af den årsag opfatte disse symbolfyldte grafiske indgreb, som kommunikative udsagn, der på subversiv vis søger at sætte spørgsmålstegn ved brugergrænsefladens æstetik og troværdighed.

2.3 Det fremmede i maskinen

Dette afsnit vil - endnu engang - beskrive en payload-strategi, som er i modsætning til det foregående afsnits. Hvor det passerede afsnit beskrev vira, der gav sig til kende indenfor den grafiske brugergrænseflades velkendte forståelsesrammer, vil dette afsnit omhandle vira, der kommunikerer via det fremmedartede og uforståelige. De forsøger med sine effekter at iscenesætte en faretruende tilstedeværelse i systemet, for herigennem at aktiverer paranoide forestillinger om den digitale infektions natur. De kommunikerer i kraft af kulturelt forankrede forestillinger om computervirusset, hvilket vil sige, at den frygt og massehysteri, fænomenet er omgærdet af, udnyttes som virkemiddel i offerets møde med virussen.



Virussen FEAR af NecroSoft Technologies

Frygt og massehysteri

Året 1992 bød på virussen *FEAR*, som bragte følgende tekstbesked med sig:

" I'm afraid you have been bitten by the FEAR bug. Just think, you may have passed it on to all of your friends! What fun it will be when it goes off in all of their systems!"

Med begrebet *fear bug*, som er et slangudtryk for at blive ramt af angst, paranoia og de vrangforestillinger dette måtte indebære, antydes det dobbelttydigt, at det ikke udelukkende er viruskoden, der er selvreplicerende, men ligeledes virusfrygten, der som en parasit inficerer brugernes bevidsthed. *FEAR* virussen kommenterer hermed et vigtigt karaktertræk ved computervirussen, nemlig at den ikke udelukkende er

kontagios i teknisk forstand, men også i kulturelle henseender. De digitale vira udgør på mange måder, hvad *Richard Dawkins*, med sit *meme* begreb, formulerer som kulturelt selvreplicerende idéer⁴⁸. De formerer sig bl.a. i kraft af deres negative omdømme, effektivt i medierne og den offentlige bevidsthed. I denne egenskab demonstrerer de et karaktertræk, som ifølge *Elaine Showalter* er symptomatisk for den informationsteknologiske tidsalder:

*“Hysteria not only survives in the 1990s, it is more contagious than in the past. (...) Infectious epidemics of hysteria spread by stories circulated through self-help books, articles in newspapers and magazines, TV talk shows and series, films, the Internet, and even literary criticism. The cultural narratives of hysteria, which I call histories, multiply rapidly and uncontrollably in the era of mass media, telecommunications and e-mail”*⁴⁹

De digitale vira kan nærmest betragtes som en inkarnationen af denne mediebarne massehysteri, *Showalter* her beskriver. Dette indebærer, at underholdningsindustrien, nyhedsmedierne og sikkerhedsindustrien ikke blot formidler eller repræsenterer virusfænomenet, men indtræder i et mere komplekst forhold, hvor frygten for den digitale infektion ligeledes indtager en epidemisk karakter. I denne tilstand af massehysteri, som moderne trusselsbilleder har en tendens til at afføde, trives myter og vrangforestillinger i bedste velgående. Dette har gennem tiderne haft en betydelig indflydelse på den gængse opfattelse af de digitale viras gøre og være.

Den biologiske virus

Lad os i forlængelse af dette konstatere, at frygten for computervirus ikke udelukkende stammer fra den digitale kultur. Hvor analogien mellem den digitale og den biologiske virus oprindeligt blev skitseret i forskningsmæssig sammenhæng som et praktisk hjælpemiddel i beskrivelsen af den digitale virus' karakter og egenskaber, så blev virusbetegnelsen også et prædikat, som ladede computervirussen med associationer til det biologiske. Den metaforiske genbrug af virusbetegnelsen har haft en tilbøjelighed til at sløre forskellen mellem det naturlige og artificielle, hvilket nærmest har plantet frygten for den digitale virus under huden på computerbrugerne. Dette kan ses i lyset af, at det brede kendskab til de digitale vira brød frem i cirka

⁴⁸ David Bennahum beskriver på forsiden af sit “Meme” magasin, en meme på denne måde: ”A contagious idea that replicates like a virus, passed on from mind to mind. Memes function the same way genes and viruses do, propagating through communication networks and face-to-face contact between people.” Teorien er oprindeligt formuleret af Richard Dawkins i ”The Selfish Gene” fra 1990.

⁴⁹ Showalter, 1998, s. 5

samme tidsperiode, som HIV/AIDS dukkede op. Man kan antage, at den kulturelle angst, dette udbrud medførte, blev projiceret på den digitale infektion.⁵⁰

Både sikkerhedsindustrien, massemedierne og viruskrivere har da også flittigt brugt referencen til den biologiske virus til at fremstille den digitale virus som en uforklarlig og faretruende entitet. Tænk f.eks. på antivirusindustriens brug af medicinske referencer i deres verbale og visuelle retorik, som f.eks. brugen af stetoskoper, organiske celler og røde kors i billedsproget, eller sproglige udtryk som *diskdoktor, infektion, epidemi, vaccination, karantæne* osv.⁵¹ Denne brug af sygdomsmetaforik har i høj grad været med til at fremavle ideen om de digitale vira, som levende programmer. Den sproglige genbrug af virusbetegnelsen har på denne måde fostret en mystificering af computervirusen.

Viruskriverne har ligeledes trukket på angsten for den biologiske virus. Gennem tiderne har brugernes computere været inficeret med både *DOS.AIDS*, *CyberAIDS*, *Disk-herpes*, *Hantavirus Pulmonary Syndrome* og en lang række andre viruskoder med reference til biologiske sygdomme.⁵² Iscenesættelsen sker f.eks. via tekstbeskeder og navngivning, som f.eks. *NEW AIDS* virussen, der melder sin ankomst med følgende meddelelse: "*I am the NEW AIDS Virus and your PC is InFeCtEd. I'm sorry...*", men referencen kan også trækkes på mere subtile måder, f.eks. har den *social engineering*, der ligger bag en virus som *Loveletter*, konnotationer til venerisk sygdom, idet virussens e-mail udbredte kærligheds-erklæring medfører en digital infektion. Hvor disse eksempler ved brug af tekstbeskeder skaber en direkte angstfremkaldende forståelsesramme for den digitale infektion, så findes der også en række vira, der kommunikerer med mere spidsfindige visuelle virkemidler. Man kan f.eks. sige, at de vira, jeg i afsnittet *Grafiske infektioner* behandlede, med deres brug af organisk metaforik, implicit bærer referencen til den biologiske infektion og dens symptomer.

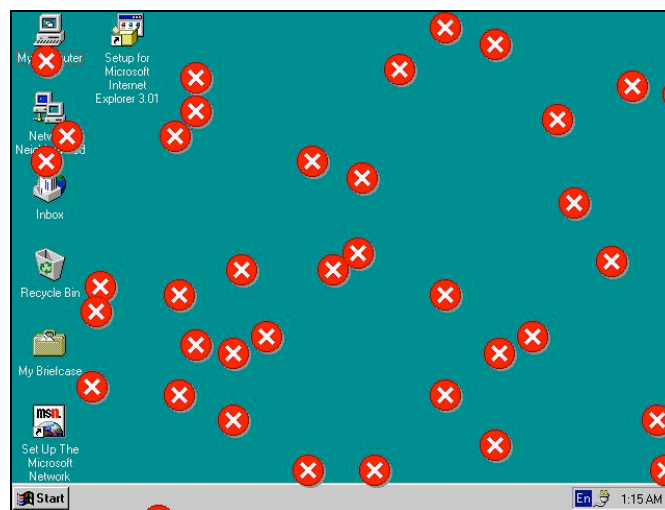
Endvidere skal det nævnes, at der findes en række computervira, som med deres grafiske effekter helt eksplicit trækker veksler på sikkerhedsindustriens visuelle retorik. Denne strategi benyttes f.eks. af *Marburg* virussen, som på en tilfældig dato lader advarselsikoner (et hvidt kryds i en rød cirkel) regne ned over skærmladen, som havde den fået røde hunde. Et andet eksempel er *Sulpex.a* virussen, som skaber en

⁵⁰ For en analyse af dette fænomen se: "Virus Culture" af Jeffrey Weinstock, 1997

⁵¹ Antivirusprogrammet Norton bruger f.eks. stetoskopet som logo på softwarens emballage o. lign.

⁵² *CyberAIDS*, *DOS.AIDS*, *CyberAIDS*, *Disk-herpes* er alle forholdsvis tidlige firser kreationer. Den navnemæssige brug af biologiske virus-referencer er primært et ældre fænomen i virussens historie.

tekstfil med navnet "Warning!!!" på skrivebordet, hver gang en ny fil inficeres på computeren - dette indebærer at skrivebordet fyldes med advarselsdokumenter i et heftigt tempo. Disse vira benytter sikkerhedsindustriens eget symbolfyldte billede-sprog, men forvrænger samtidig dets budskab via en påfaldende overeksponering af de faresignaler, det repræsenterer. De kan opfattes som parodierende kommentarer til den digitale sikkerhedsindustri's visuelle retorik og den endeløse strøm af advarsler, antivirussoftware ofte iværksætter.



Marburg af GriYo

Sulfnbk.exe: Tekst-virus

Det næste analyseobjekt – den såkaldte *hoax-virus* - falder lidt udenfor den overordnede analytiske tematik, idet det ikke er en computervirus i traditionel forstand og ikke umiddelbart indeholder nogen payload. Dog er den relevant, fordi den tager sig ud, som en nærmest tragikomisk karikatur af den massehysteri, som den digitale virus traditionelt har været omgivet af. *Hoax-virussen* har form, som en virus-advarsel, der spredes via e-mail. Det, der kendetegner denne type virus, er, at den ikke indeholder nogen eksekverbar maskinkode, hvormed dens selvspredende egenskaber udelukkende er baseret på e-mailens tekst og *social engineering*. Et af de mere udspekulerede eksempler på en *hoax-virus* er *Sulfnbk.exe*⁵³, som dukkede op i 2001. E-mailen findes på mange sprog og er ligeledes blevet oversat til dansk -

⁵³ I den forbindelse kan man nævne, at der er mistanke om, at *sulfnbk.exe* opstod som en misforståelse i kølvandet på den virus-paranoia *Magistr.a* skabte. Det vil sige, at en person tolkede *sulfnbk.exe* filen, som en *Magistr.a* inficeret fil og skrev virus-advarslen med velmente intentioner. Windows systemfilen *sulfnbk.exe* et internt program, systemet benytter til at gendanne lange filnavne. Se evt. Deseriis, 2008

formentlig af en velmenende person. Teksten tager sig ud som en advarsel og en opfordring til at videresende denne, samt en punktformet instruktion til, hvordan man fjerner den opdigtede virus fra sit system. Denne desinfektions-vejledning, som består i at brugeren skal lokalisere og slette en fil på computeren, viser sig imidlertid at være misvisende, idet offeret ved udførelsen af denne ender med at slette en systemfil (*sulfnbk.exe*) i *Windows*. Det vil sige, at brugeren selv udøver skaden på sit system. Desinfektions-instrukserne tager sig, i sin punktform, ud som en algoritme i *pseudokode*⁵⁴, der direkte kunne oversættes til maskinkode, men som i dette tilfælde er almensprogligt rettet mod offeret. Eller sagt på en anden måde; *hoax-virussen* indeholder, som den traditionelle virus, både en *bærer* og en *payload* mekanisme. Forstået på den måde, at den bliver båret frem af modtagerens umiddelbare - nærmest automatiserede – reaktion på frygten for den digitale infektion, og den udøver en *payload* effekt på systemet, som i offerets manglende evne til at afkode *pseudokodens* tekniske karakter og konsekvens, eksekveres manuelt af offeret selv. Der er tale om, at virussens tekniske og sociale niveauer er repræsenteret i en og samme almensproglige syntaks. Dette formelle aspekt indebærer endvidere, at dens udbredelse er baseret på sociale og kulturelle aspekter, snarere end tekniske.

I forlængelse af dette kan man pege på en af *Jean Baudrillard's* betragtninger vedrørende computervirusfænomenet:

*”Within the computer web, the negative effect of viruses is propagated much faster than the positive effect of information. That is why a virus is an information itself. It proliferates itself better than others, biologically speaking, because it is at the same time both medium and message. It creates the ultra-modern form of communication which does not distinguish, according to McLuhan, between the information itself and its carrier.”*⁵⁵

Medieteoretikeren *Marshall McLuhans* kendte *The Medium is the Message*⁵⁶ tankegang, som *Baudrillard* i ovenstående citat refererer til, betyder i den forbindelse, at et givent medies *budskab* er den indvirkning mediets *form* har på samfundet, og den måde vi opfatter og agerer i virkeligheden. Det vil sige, at

⁵⁴ Pseudokode er algoritmer skrevet i almindeligt sprog, men med strukturelle konventioner fra computerbaserede programmeringssprog.

⁵⁵ Citeret fra kunstgruppen 0100101110101101.org's hjemmeside.

⁵⁶ McLuhan, 1964, s. 7

fremkomsten af nye informationsteknologiske medier ikke udelukkende ændrer vores adgang til og forbrug af medier og informationer, men ligeledes skaber nye kommunikative paradigmer og informationelle entiteter. Følger man *McLuhans* tankegang, kan man ræsonnere, at grunden, til at nyheden om virusudbrud spredes så effektivt i det netværksbaserede samfund, findes i, at den digitale virus, som teknisk og informationel entitet, er en integreret del af det informationsteknologiske medie. De digitale virus evne til nærmest eksplosivt at brede sig i massemedierne og den offentlige bevidsthed opstår i denne optik, i kraft af deres egenskab af at være fuldbyrdede hybrider mellem medie og budskab. ”Sygdommen” bæres med andre ord af det samme medie, som bringer nyheden om den. *Hoax-virussen* udnytter eksplicit denne egenskab.

Den elektroniske graffiti

Nogle viruskrivere synes direkte at have spekuleret i viraenes kulturelt kontagiøse egenskaber. At kunne læse om deres viruskreationer i nyhedsmedierne, *in the wild* lister eller på antivirusindustriens hjemmesider, kunne give disse skrivere ”*a real boost to the ego*”, som antivirusaktør og tidligere virusprogramør *D. D. Shelby* formulerer det⁵⁷. Det lader da også til, at viruskriverne nærmest har betragtet antivirusindustriens analyser, beskrivelser og virusvarsler som anmeldelser af deres arbejde. Dette kan ses i forlængelse af, at en af motivationerne - for nogle af virusscenens aktører - var ideen om at bruge computervirussen til at sprede deres digitale aliaser verden rundt. Man kan f.eks. nævne at viruskriveren *Hellraiser* under et interview til *Wired Magazine* i 1995, proklamerede: ”*Viruses are the electronic form of graffiti.*”⁵⁸ Disse digitale graffiti-skrivere var meget bevidste om den opmærksomhed, de kunne opnå via deres virusproduktion, både ved udbredelsen af selve virussen, men ligeledes gennem den omtale, de kunne opnå i massemedierne. Der blev op gennem halvfemserne bragt mange interviews og artikler om og med virusscenens aktører. For læserne fremstod virusscenen som en hemmelighedsfuld undergrundskultur, befolket med lyssky computergenier, der gemte sig bag kryptiske aliaser og havde uforståelige ondsindede intentioner – alt dette gjorde emnet til spændende læsestof. Den opmærksomhed, dette kunne medføre, har formentlig tiltalt mange viruskrivere.

⁵⁷ Shelby, 2002 (ingen sidetal)

⁵⁸ Dibbell, 1995 (ingen sidetal)

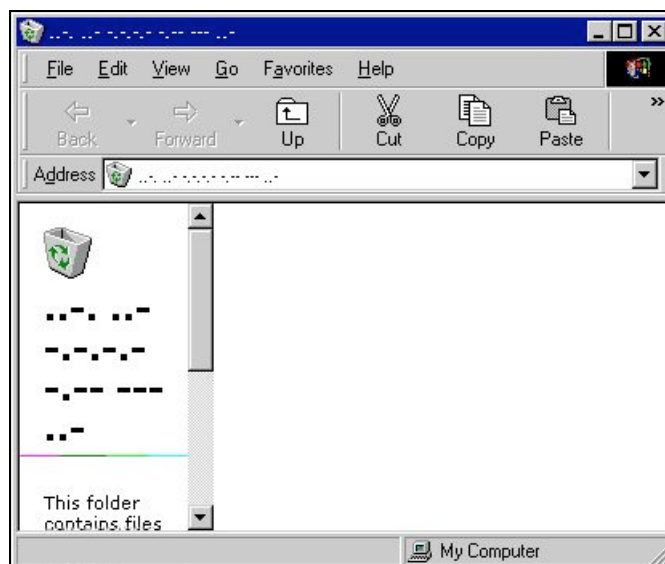
Ligheden med den analoge graffiti kan da også give en forklaring på opblomstringen af de mere opfindsomme payloads. I lighed med den analoge graffiti, som udviklede sig fra at være simple *tags* til at blive mere og mere farvefulde og omfangsrige *pieces*, gennemgik viruskreationerne en lignende transformation i relation til den opmærksomhedsskabende kreativitet. De mere opfindsomme payloads, jeg i denne tekst behandler, kan med andre ord betragtes, som produkter af denne udvikling. De er i dette perspektiv designet til at vække opsigt og til at give offeret en oplevelse, som vedkommende ikke glemmer foreløbigt. Visse virusprogrammørers appetit efter medieopmærksomhed kan samtidig give en mulig forklaring på nogle af de mere ondsindede virusaktiviteter. For en virusprogrammør var og er den mest effektive måde at få omtale i nyhedsmediernes overskrifter at skabe en destruktiv virus.

De fleste af Vx-scenens viruskreationer baserer dog udelukkende sine opsigtvækkende træk på uskadelige virkemidler, der dog sjældent holder sig tilbage for at give offeret en chokerende oplevelse. *SilverRat* virussen er et af de mere idérige tilfælde af denne slags.

SilverRat: En suspekt gæst

SilverRat virussen er en blanding af en orm og en e-mailvirus, der spreder sig via lokale netværk, internettet og IRC kanaler⁵⁹. Det særegne ved *SilverRat* er, at den inkluderer et af-installationsikon i *Windows-systemets* kontrolpanel under 'Tilføj/Fjern programmer' delen. Virussen giver sig altså til kende og tilbyder tilsyneladende brugeren at fjerne virussen via ikonet mærket med "De-install *SilverRat virus*". Denne gestus viser sig dog at være blændværk, idet eksekveringen af af-installationsinstruksen ikke fjerner virussen fra systemet, men i stedet resulterer i, at en beskedboks med teksten "I have to return some videos" dukker frem på skærmen. Ved samme lejlighed overskrives papirkurvens ikon- og adresselinjetekst med en kryptisk sekvens af punktummer og bindestreger, der ved nærmere eftersyn viser sig at være morsekode, som kan oversættes til "Fuck You".

⁵⁹ "Internet Relay Chat" vil sige et chatforum. Hvis computeren har et IRC-chatprogram installeret, inficeres dette med et script, der sender en kopi af virussen til de brugere der er logget på samme chatkanal.



SilverRat

Misledende grafisk repræsentation

Virussens udløser altså ikke selv sin payload via en *logical bomb* eller en tidsindstillet *trigger-mekanisme*, men overlader denne gerning til brugeren selv. Eksekveringen af payloaden er hermed baseret på den godtroende brugers tillid til, at den grafiske repræsentation af funktionalitet, som operativsystemet viser, er pålidelig. Brugeren bliver, med af-installationsikonet, sat i en paradoksal situation, hvor der skal vælges mellem tilliden til operativsystemets rationalitet og mistilliden til den uindbudte gæst. Tilliden til systemet, der så at sige sejrer, hvis brugeren vælger at klikke på ikonet, sættes følgende på prøve med det inficerede systems uforudsigelige opførsel.

Med tekstbeskeden "*I have to return some videos*" som er et citat fra den filmatiserede bog *American Psycho*⁶⁰, fremhæves den fremmede tilstedeværelse, der har indlejret sig i systemet via viruskoden. Den citerede sætning benyttes af fortællingens hovedperson, seriemorderen *Patrick Bateman*, til at bortforklare og redegøre for den tid, han bruger på sine skumle bedrifter, samt som en bekvem undskyldning for at undgå social omgang med andre mennesker. På samme måde fungerer beskeden, når den benyttes af *SilverRat* virussen som en undskyldning for ikke at efterkomme brugerens instruks, for i stedet at hengive sig til de mere tvivlsomme algoritmiske gøremål. Samtidig bliver computeren, med citatets sproglige udsagn i første person, iscenesat som en skikkelse med egen vilje. Virussen tager herigennem form som et selvstændigt væsen, der modarbejder brugerens interesser og

⁶⁰ Ellis, 1991

træffer sine egne valg. Systemet bliver ladet med en subjektivitet, som står i skærende kontrast til forestillingen om computeren som en objektiv maskine, der per automatik tjener brugerens behov og vilje. Virussens kryptiske visuelle udsagn, uforudsigelige misledende adfærd og åbenlyse egenlyst kiler sig - som en parasit - ind og bryder operativsystemets harmoniske udstråling og logik.

Man kan udlede, at denne type vira, via dens misvisende repræsentationer af funktionalitet, samt dens ulogiske og ofte foruroligende adfærdsmønstre, udfordrer eller måske ligefrem underminerer brugerens tillid til computeren. Det paradoks, der ligger i den generelle tillid til teknologien, på trods af den almene brugers manglende indsigt i, hvordan en computer grundlæggende fungerer, synes at interessere mange viruskrivere. F.eks. erklærede viruskriveren *Ellison* i et interview i *Wired Magazine*:

*"People trust computers and technology too much, when so few really understand them. Viruses, by that view provide a destabilizing force to remedy such unfounded trust."*⁶¹

Ifølge *Ellison* er viruscreation altså et middel til at kultivere brugerens kritiske bevidsthed i forhold til computerteknologien. Antager man, at denne tillid til teknologien er avlet af den måde, software fremstiller sig selv, kan man konkludere, at virusfænomenet indtager en modsatrettet kulturel funktion i forhold til softwareindustriens produkter. Hvor den kommercielle software, gennem dets rationelle logik, fundamentalt set er baseret på at fostre fornemmelsen af tryghed og sikkerhed, fungerer virussen, som en destabiliserende faktor i disse forestillinger. Den bliver en modvægt til den almene brugers antagelse af, at software er neutrale indretninger, og viser i stedet, at den kode, softwaren er baseret på, ikke af natur er objektiv eller ufarlig.

I forhold til dette rationale af kritisk potentiale er det dog ikke svært at argumentere for, at virusfænomenet dybest set har haft præcis den modsatte effekt. Udbredelsen af de digitale vira, og følgelig den store opmærksomhed virusfænomenet har haft i massemedierne og den offentlige bevidsthed igennem tiderne, har medført, at den almene bruger ofte har en tendens til at mistænke enhver fejl eller uigennemskuelig opførsel i relation til computerbrugen som resultatet af en virus infektion. I dette perspektiv indtager virusfænomenet altså ikke en destabiliserende funktion i

⁶¹ Dibbell, 1995 (ingen sidetal)

forhold til brugerens tillid til computeren, men bliver snarere en forklaring på, hvorfor computeren ikke fungerer optimalt eller tilregneligt. I den forbindelse har virusfænomenet ligeledes hyppigt indtaget rollen som sydebuk for den kommercielle software/hardwares fejl og mangler. Virusparanoiaen vejer så tungt i brugernes bevidsthed, at virussen – og ikke computerteknologien – bliver genstand for brugerens mistillid, uanset om den er tilstede eller ej.

Digitale monstre og binære aliens

Computervira bliver i høj grad betragtet som uønskede digitale immigranter, der overskrider den personlige computers grænser. De fremmaner følelsen af en tilstedeværelse af noget fremmed i operativsystemet og brugerens digitale privatsfære. På trods af, at en computervirus kan sprede sig ukontrolleret i digitale netværk, så er virussens selvreplicerende egenskaber og payload-effekter fuldstændigt fast defineret i dens kildekode. Uanset brugerens fornemmelse af, at en virus opfører sig uforudsigeligt og ukontrolleret, så er dens indvirkning på systemet konkret algoritmisk specificeret fra ende til anden. Men, hvor det for den viruskyndige er fast afgrænset, hvad viruskode kan gøre, er det tværtimod komplet uigennemskueligt for lægmand; virussen kan fra dennes synsvinkel gøre hvad som helst. *Julian Dibbell* beskriver mødet således:

“What scares you most about getting that virus? Is it the prospect of witnessing your system's gradual decay, one nagging symptom following another until one day the whole thing comes to a halt? Or is it not, in fact, something deeper? Could it be that what scares you most about the virus is not any particular effect it might have, but simply its assertive, alien presence, its intrusive otherness”⁶²

I denne foruroligende uvished om virussens intentioner, egenskaber og konsekvenser, indtager den karakter af at være en abstrakt og obskur entitet – et *digitalt monster* eller en *binær alien*⁶³. Virussen bliver en uforståelig og uhåndgribelig fremmed tilstedeværelse, som fostrer skræmmende forestillinger om dens ophav og væsen. Der åbnes op for paranoide associationer, hvor fiktive forestillinger om virussens beskaffenhed kan blive langt mere intimiderende end dens reelle påvirkning af systemet.

⁶² Ibid. (ingen sidetal)

⁶³ Som formuleret af Parikka, 2005 (ingen sidetal)

I den forbindelse er det værd at nævne, at det netop er denne karakteregenskab, som en mystiske og fremmed entitet, der gør virusfænomenet uhyre egnet som nyhedsstof og materiale i filmiske og litterære fremstillinger. De digitale vira har, som alle andre velkomponerede monstre, evnen til at vække frygt, forundring og fascination på en og samme tid, hvilket gør dem til medrivende stof i massemediernes og underholdningsindustriens mere eller mindre virkelighedsnære skildringer af fænomenet.

SilverRat iscenesætter, via *American Psycho* referencen, meget eksplicit sig selv som en ondsindet skikkelse, med sin egen suspekta agenda. Den personificerer via *Bateman* referencen virussen som en figur med uforståelige og utilregnelige giftige intentioner. Virusprogrammøren er med andre ord bevidst om modtagernes paranoide forestillinger og adresserer dem direkte via den grafiske payload. Iscenesættelsen af virussen - via filmcitatet - benyttes på denne måde som et virkemiddel, hvis hensigt er at give os en hårrejsende oplevelse, på præcist samme måde, som når vi møder det fremmede i en gyserfilm.

Den kryptiske sekvens af punktummer og bindestreger, som for de fleste sikkert ikke umiddelbart åbenbarer sig som morsekode, fungerer i sin binære natur efter samme princip, som computerens elementære mekanismer. Sekvensen kan med andre ord let forveksles med computerens egen binære notation, og den repræsenterer her igennem den for slutbrugeren uforståelige kode, som driver maskinen. I dette lys fremstår *SilverRats* modifikation af papirkurvens tekststrenger, som om computerens underliggende sprog og logik trænger frem i interfacets velkendte overflade, for her igennem at gøre den kryptisk og gådefuld. Den måde, det binære tager sig ud, som noget fremmedartet i den grafiske brugergrænseflades velkendte rammer, viser i hvilken grad computerens grundliggende natur er fortrængt fra systemets overflade. Det fremmede i maskinen bliver på denne måde ikke udelukkende repræsenteret i form af virussen, men ligeledes i form af computerens egen maskinelle – binære – natur. Brugerens angst for det ukendte knyttes hermed op på computerens egne underliggende mekanismer; det fremmede viser sig at være noget, der er dybt indlejret i computerens indre. Det er hermed oplagt at læse *SilverRat* virussen som et billede på, hvordan udbredelsen af grafiske brugergrænseflader, og med dem den stadig stigende tendens til at skjule maskinkoden og mediets underliggende mekanismer, har

fremmedgjort slutbrugeren fra computerens grundlæggende sprog og maskinelle beskaffenhed.

SilverRat sammenfletter paranoia, film referencer, kryptisk kode og provokerende algoritmisk adfærd i et komplekst udsagn, hvor referencerne peger både ind i og ud af computeren. Dens gådefulde visuelle kommunikation divergerer fra den logik, der gennemsyrrer operativsystemets lettilgængelige flader, idet den gør krav på en besværlig afkodning af sine kryptiske udsagn. *SilverRat* hengiver sig, med andre ord til en irrationel og tvetydig kommunikation, hvor funktionalitet og budskab ikke er letforståelig eller direkte tilgængelig, som det er tilfældet i operativsystemets vanlige tilstand.

Om parodi og paranoia

Mens selve det misledende, ubrugervenlige og dysfunktionelle tager sig ud som fremmedartet indenfor det moderne operativsystems rationelle rammer, overdriver dette afsnits vira denne fremmedhed. Ved at aktivere og overdramatisere gængse forestillinger om den digitale virus' ondsindede natur, karikeres brugerens virusfrygt. Effekternes overspillede iscenesættelser af en fremmed tilstedeværelse i systemet fremstår i dette lys, som en parodi på de paranoide forestillinger og den massehysteri virusfænomenet er omgærdet af. En tidligere viruskriver kommenterer den aura af myter og mystik, de digitale vira gennem tiderne har været indhyllet i, med følgende:

*"The truth behind it – one that was rarely noticed – was that a computer-virus is a program – nothing more, nothing less"*⁶⁴

For viruskriverne er mytedannelsen omkring virusfænomenet og den frygt, denne har medført, absurd. Den almene opfattelse af computervirussen er blevet et paranoidt fantasifoster, fremavlet i en vekselvirkning mellem antivirusindustriens kommercielle interesser, nyhedsmediernes svælg i trusselsbilleder samt computerbrugernes skortende indsigt i, hvad en virus egentlig er og kan gøre. Den tørre konstatering af, at en virus er et computerprogram, hverken mere eller mindre, anviser den ironi, der er i de grafiske payloads, dette afsnit har behandlet. Om mødet med disse munder ud i parodi eller paranoia, afhænger i høj grad af modtagerens forestilling og viden om, hvad en virus grundlæggende er. De mere kulrede vrangforestillinger om virussets

⁶⁴ Shelby, 2002 (ingen sidetal)

væren og gøren opstår næppe, hvis offeret har en basal indsigt i, hvad et computerprogram er.

Med viruskrivernes tilbøjelighed til at spille på den mystik virusfænomenet er omgivet af, ligger der på samme tid en sarkasme, som potentielt er afmystificerende. Offeret får, med disse overdramatiserede dæmoner i maskinen, sine vrangbilleder skåret ud i pap. Det bliver her igennem iøjnefaldende, at de digitale vira ikke er uforklarlige eller overnaturlige skabninger, men konstruerede entiteter; et produkt af kultur - ikke (over)natur. Selv om virustruslen i dette lys udfolder sig i et mere jordnært perspektiv, tager de digitale vira sig dog ikke ud som ufarlige, idet de tilkendegiver en chokerende kontrol over systemet. De demonstrerer, med deres fiflen med funktionalitet og misledende grafisk repræsentation, den manipulation, der potentielt kan foregå bag skærmen.

3. Computervirus-æstetik og -kultur

Æstetiske virkemidler

Det burde efterhånden være klart, at viraenes effekter og virkemidler, gennemgående rummer en stor del ironi og sarkasme. Der er, hvis offeret kan overkomme sin frygt for, at systemet aldrig vender tilbage til normal tilstand, ligeledes en god portion humor i viraenes indgreb. Det er en humor, som baserer sig på at forvirre den intetanende computerbruger gennem fornuftstridig computeropførsel og fjollede effekter. Payloadens indgreb kan i mange tilfælde betragtes som en slags kodede *practical jokes*, som peger fingre af lægmandsbrugerens (sparsomme) teknologi-forståelse og parodierer softwareindustriens designskikke og interfaceæstetik. På den anden side er det nok de færreste af virusinfektionernes ofre, der finder disse effekter specielt morsomme. I bevidstheden om at man nu har fået sin computer inficeret med en potentielt destruktiv virus, blegner de mere humoristiske aspekter af infektionen.

Alligevel formår disse digitale vira, ved brug af deres uforudsigelig *trigger-rutiner*, ofte at eksekvere sine effekter, uden - eller inden - offeret er bevidst om, at de er forårsaget af en virusinfektion. I disse tilfælde, hvor modtageren ikke får mødet indhyllet i frygt og paranoia, ligger der potentielt både overrumplende og forvirrende oplevelser i vente. Dette øjeblik, hvor viruseffekten uventet indtræffer i systemets velkendte rammer – som når operativsystemet smelter eller tekstbehandlingsprogrammet pludselig er rundt - er beskueren åben for en oplevelsesorienteret

kommunikation. Det er dette greb mange vira iværksætter med deres *trigger-mekanismer*. Forskydningerne af interfacets form og logik er interfaceæstetiske virkemidler, der henvender sig umiddelbart til modtageren af viruskoden. Hvor denne kommunikation, som jeg i analysen har uddybet, kan afkodes i flere retninger, så er et fælles træk, at den ikke holder sig til softwarens fastlagte rammer for kommunikation – den forsøger snarere at bryde dem. Viraene udfolder gennem deres manipulation af softwarens repræsentationelle lag - dets udformning og metaforer - højst kontekstuelle udsagn, som nægter at skelne mellem software-arkitekturen og det indhold, der formidles via denne. Disse payload-effekter er baseret på at overskride softwarens konventioner og dens fastdefinerede grænser mellem form og indhold. I den forbindelse er det centrale virkemiddel det fornuftstridige og tvetydige, og dette synes ikke at være nogen tilfældighed - viruskriveren *Opic* udtaler f.eks. i et interview:

*"I'm into making points through ambiguity, rather than the tried and true in-your-face method,"*⁶⁵

Payload-effekterne iscenesætter irrationelle og uforudsigelige situationer, der forvirrer eller forundrer computerbrugeren, og det er i disse tilstande, hvor det velkendte pludselig fremstår tvetydigt og fremmed, at der potentielt sker, hvad man kan kalde en æstetisk erfaringsdannelse. Introduktionen af det tvetydige indenfor operativsystemets rationelle rammer, provokerer perceptuelle og kognitive rammer for forståelse, for på denne måde potentielt at åbne op for en genforhandling af den måde vi ser og forstår computeren. Man kan sige, at denne flertydighed, de grafiske payloads iscenesætter, er et æstetisk virkemiddel, som appellerer til refleksion og kritisk revurdering. De fastgroede forestillinger om systemets natur undermineres for her igennem at bane vejen for en revurdering af dets præmisser. Det vil sige, at den grafiske payload leverer en sanseforankret betydningsdannelse, som folder sig ud i kollisionen mellem dets æstetiske udsagn og computerbrugers kognitive erfaring. Dette sanssemæssige skisma, som potentielt indfinder sig i offerets møde med virussens visuelle effekter, opstår i kraft af den måde, modtageren opfatter og forstår computeren, og dette omfatter såvel bevidste, som ubevidste erfaringslag. Effekternes perceptuelt konflikt-skabende udsagn er, med andre ord, knyttet op på computerbrugers opfattelse af, hvad en computer er, samt formodningerne om, hvilke muligheder og begrænsninger

⁶⁵ Neely, 1999 (ingen sidetal)

den repræsenterer. Man kan på denne måde konkludere, at disse vira ikke udelukkende engagerer sig i at bryde ind i systemet i teknisk forstand, de bryder ligeledes ind i computerbrugerens konception af systemet.

Computermediets effekt

Brugerens opfattelse af computeren formes grundlæggende af den måde computeren fremstiller sig selv til via dens interfaces. I forlængelse af denne konstatering er det væsentligt at bemærke, at brugeren i den rutinemæssige brug af computeren nærmer sig den og begynder at tilegne sig verden i dens perspektiv. At medier konstruerer sanse- og perceptionsorganer, var en af *McLuhans* vigtigste pointer, da han skrev "*Understanding Media*" i 1964:

*"The effects of technology do not occur at the level of opinions or concepts, but alter our sense ratios or patterns of perception steadily and without any resistance"*⁶⁶

McLuhans teori baserer sig på grundtanken om, at fremkomsten af nye teknologier og medier har en afgørende indvirkning på samfundet. Det centrale i den sammenhæng er, at mediernes effekt i højere grad er bestemt af teknologiens form, end ved det indhold den formidler. Mediets form danner rammen for muligheder og begrænsninger i forhold til det indhold, der formidles, hvilket har stor betydning i forhold til, hvad der bliver formidlet og hvordan dette sker. Dette indebærer, at med hver ny informationsteknologisk opfindelse, opstår der ligeledes nye perceptuelle paradigmer. Der er tale om, at der implicit i ethvert givent medie fungerer en række specifikke repræsentationelle mekanismer, som har indflydelse på vores måde at recipere og afkode den information, vi bliver præsenteret for. Det vil sige, at fremkomsten af nye informationsteknologiske medier ikke udelukkende ændrer vores adgang til - og forbrug af - medier og informationer, men ligeledes i bredere forstand har indflydelse på vores måde at percipere, afkode og forstå den virkelighed, vi lever i, og som *Mcluhan* påpeger, er det en indflydelse, som foregår fortløbende og uden modstand.

Denne teori er skrevet, før computeren for alvor blev udbredt i samfundet, men er - på trods af sin temmelig deterministiske karakter - stadig gangbar, man må dog tage hensyn til de ændringer, mediet har bragt. Computeren er interaktiv og

⁶⁶ McLuhan, 1964, s. 19

eksekverbar, hvilket vil sige, at det er et medie, som sammenblender repræsentation og handling. Hvor *McLuhan* fremhæver, at medierne har indflydelse på den måde, vi agerer i verden, kan dette i relation til computeren forstås i en mere direkte forstand, idet den helt konkret giver brugeren mulighed for at interagere med verden via sine brugergrænseflader. De interaktive mediers samfundsmæssige effekt er med andre ord ikke udelukkende defineret af den måde, de repræsenterer virkeligheden, men ligeledes gennem den måde, den lader os interagere med den.

Interfacet er den formelle anordning computerteknologien benytter til repræsentation og interaktion. *Lev Manovich* taler, i den forbindelse, om begrebet *det kulturelle interface*⁶⁷, der beskriver måden, hvorpå computeren præsenterer og lader brugeren interagere med kulturelle data. I den forbindelse påpeger han, at interfacet ikke spiller nogen neutral rolle i formidlingen af informationer:

*”In short, far from being a transparent window into the data inside a computer, the interface brings with it strong messages of its own.”*⁶⁸

For at kunne gøre mediet transparent fordrer det, at man kan lave en klar adskillelse af form og indhold. Dette er ifølge *Manovich* ikke muligt, fordi mediet er ladet med betydning og presser sin egen logik ned over indholdet. Man kan sige, at kommunikationen er afhængig af interfacet, hvormed interfacet ikke kun formidler et indhold, men er en del af det. Computerens interfaces er altså ikke bare neutrale anordninger, som lader os interagere med de data eller den virkelighed, computeren repræsenterer, de har ligeledes en indflydelse på, hvordan vi forstår og behandler denne information. Man kan med andre ord sige, at den måde computermediet giver sig til kende for os, via dets interfaces’ designmæssige struktur og fremtoning, har en stor indflydelse på, hvordan vi forstår og bruger computeren. Der er da også blevet argumenteret for, at softwarens implementering af konkrete receptions- og arbejds-metodologier direkte leder brugerne til at tænke og handle på bestemte måder.

Matthew Fuller skriver f.eks.:

*“This is the fatal endpoint of the standard mode of HCI. It empowers users by modelling them, and in doing so effects their disappearance, their incorporation into its models.”*⁶⁹

⁶⁷ Manovich, 2001, s. 69

⁶⁸ Ibid. s. 65

⁶⁹ Fuller, 2003, s.13

Computeren former brugeren i dens billede gennem inkorporeringen af denne i brugergrænsefladernes modeller for interaktion og erkendelse. I den forstand udøver computeren en magt over brugeren, og denne kan udøves bevidst eller ubevidst for såvel bruger som softwareproducent. Samtidig er dette magtforhold, ikke altid let at gennemskue, fordi de moderne brugergrænseflader er baseret på idéen om at give brugeren fornemmelsen af fuldstændig gennemskuelighed og kontrol i interaktionen med computeren. HCI-orienteret design baserer sig, som *Fuller* formulerer det, i udstrakt grad på forestillinger om *empowerment* og suveræniteten hos den bruger, hvis "personlighed" former og går i dialog med maskinen.⁷⁰

Kritiske potentialer

Computeren er altså ikke en neutral maskine, den influerer menneskets måde at se, vide og handle på i verden, og efterhånden som de digitale medier er blevet allestedsnærværende og har indtaget et utal af funktioner i samfundet, er det en væsentlig indflydelse eller magt den besidder. I dette perspektiv udgør de digitale virus brud med softwarens rammer og logik et kritisk potentiale, idet indgrebene - via deres irrationelle, tvetydige og fremmede karaktertræk - forstyrrer de sanseorganer, handlemodeller og magtrelationer systemet implementerer gennem dets brugergrænseflader. Payload-effekterne kolliderer med de perceptions- og orienteringsmønstre vi tilegner os i den vanemæssige brug af computeren, for i denne proces at åbne op for en reflektiv ombrydning af etablerede forståelsesrammer. Hvor forestillingen om softwarens neutrale og formålsbestemte karaktertræk, er metaforisk manifesteret i det grafiske interfaces brugervenlige overflader, blotlægger disse interventioner, den subjektivitet der er indlejret i softwarens algoritmer; de påpeger at den måde vi ser og forstår softwaren er en konstruktion.

Det er altså ikke konkrete budskaber payload-effekterne overbringer, det er snarere konceptuelle udsagn, som foldes ud i eksperimentet i at sprænge brugergrænsefladens konventioner. De repræsenterer en undersøgelse af det materiale, vi kalder software, og dette foregår i en vekselvirkning med de kulturelle sammenhænge softwaren er blandet ind i - og betinget af. Dette leder os til det næste spor, der løber igennem min analyse af de grafiske payloads; deres udsagn kan aflæses på flere

⁷⁰ Ibid. s.12

(bruger)niveauer. De kan afkodes umiddelbart, via disse sansebaserede virkemidler jeg netop har beskrevet, men de kan ligeledes afkodes med refleksion i koderelaterede eller kulturelle kontekster. De kommenterer med andre ord, på mere subtil vis, forskellige karaktertræk og problematikker i relation til den digitale kultur. Man kan sige, at viraenes udsagn er dobbeltkodede i samme forstand, som man forstår det i relation til det postmoderne tekstbegreb.⁷¹ Denne dobbeltkodede egenskab tillader både en afkodning på interfacekulturens egne ”overfladiske” præmisser, men holder ligeledes nogle dybere mediebevidste og selvreflekterende afkodningsmuligheder. Den ironiske kommentar der f.eks. ligger i at vende vrangen ud på *det fremmede i maskinen* - ved at lade det binært kodede træde frem i interfacets overflade, eller ved at få computerbrugeren til manuelt at eksekvere en destruktiv algoritme, som offeret ikke kan afkode, fordi den er almensprogligt formuleret - taler ind i en meget specifik problemstilling, nemlig en der omhandler det fundamentale skel, der er mellem ekspert- og lægmandsbruger i den nyere computerkultur. Dette skel forekommer mig at være helt centralt i læsningen af virusfænomenet. Lad os derfor se nærmere på, hvad det er udgjort af.

Instrumentalisme og determinisme

Historisk set kan man sige, at med udviklingen af de GUI-baserede operativsystemer, var vejen banet for den kommercielle masseudbredelse af den personlige computer. Den visuelt appellerende brugervenlighed disse systemer bragte, gjorde det muligt at markedsføre computeren til hvermand, men samtidig ændrede disse nye systemer også den måde, brugerne forstod og brugte computerteknologien. Computeren blev ikke længere betragtet som en slags avanceret lommeregner, men som et medie, der rummede et utal af anvendelses og udfoldelses-muligheder for det enkelte menneske. Samtidig kan man, i et kritisk perspektiv, sige, at der undervejs i denne kommercialiseringsproces – frem til i dag – har fæstnet sig nogle ganske stereotype forestillinger om computerens natur i den digitale kultur og computerbrugernes bevidsthed. De to væsentligste stereotyper kan kategoriseres under *instrumentalisme* og *determinisme*. Disse begreber repræsenterer en række forestillinger og

⁷¹ I den forbindelse er det ligeledes værd at nævne, at disse grafiske payloads æstetisk i det hele taget har et ganske postmodernistisk islæt. Man siger, at nogle af postmodernismens stilmæssige markører er det dobbeltkodede, selv-reflekterende, tvetydige, paradoksale og ikke mindst det ironiske. (Barker, 2008, s. 204) Disse karaktertræk er ganske beskrivende for de grafiske payloads.

konventioner, der efterhånden er så indgroede i den almene computerkultur, at det er svært at forestille sig, at den kunne være anderledes.

Instrumentalisme er den simple og meget udbredte forestilling om, at computerteknologi og software udvikles for at løse bestemte mål. Dette bygger på en antagelse af, at de opgaver, teknologien skal varetage, er fast defineret på forhånd. I dette perspektiv tager software-design og produktion sig primært ud som et felt, der handler om at udvikle redskaber til at løse konkrete opgaver. Dette fostrer ligeledes en formodning om, at software er neutrale anordninger, som udelukkende opererer i brugerens interesse; som hjælpemidler til at opfylde brugernes behov og ønsker. I kølvandet på denne instrumentalisme følger nærmest naturligt en øget teknologisk determinisme. Denne udarter sig i en optimistisk tro på, at teknologien uundgåeligt, og i en lige linje, udvikler sig til at blive bedre og bedre. Den er bygget på en ufravigelig tro på, at det nyeste altid er en forbedring af det eksisterende.

Konventionelt softwaredesign er i høj grad fanget mellem disse to gængse karikaturer af computerteknologien. De repræsenterer en række designkonventioner og teknologi-forståelser, som har stabiliseret sig i en vekselvirkning mellem den kommercielle softwareproduktion og dens brugere. I et økonomisk perspektiv kan man samtidig sige, at de på mange måder er fundamentet for den software-teknologiske forbrugskultur, idet de forudsætter, at digitale pakkedesigner konstant må opdateres og udskiftes til nyere, hurtigere og bedre produkter. De markeds-økonomiske hjul holdes i gang, men man kan også argumentere for, at tendensen bliver en spændetrøje for softwareindustrien, som af kommercielle hensyn ikke kan tillade sig at foretage gennemgribende ændringer i softwareproduktens arkitektur og design. Dette skyldes ligeledes at forbrugerne er reaktionære - de bryder sig generelt ikke om at tilegne sig nye grundlæggende strukturer, der bryder med det, de kender i forvejen. Softwareindustriens produkter bliver en endeløs føljeton af nye versioner, som bygger oven på eksisterende designkoncepter og sjældent skuer tilbage og gentænker softwarens grundlæggende struktur. Den kommercielle softwareudvikling bliver på denne måde styret af konventioner, som industrien for længst har vedtaget, og som fastfryses i forbrugermassens tilegnelse af dem. Fra en kritisk vinkel er det disse mekanismer, softwareproduktionen er underlagt i dag.

Fra utopi til massekommercialisering

Skruer man tiden tilbage til den personlige computers spæde udviklingsmæssige begyndelse, vil man opdage, at computerteknologien var forankret i nogle radikalt anderledes idéer om, hvad en computer er - og kunne være blevet. Tænk f.eks. på *Alan Kay* og *Adele Goldbergs* visionære arbejde under udviklingen af *DynaBook*⁷² computeren i midten af halvfjerdserne. Denne forskning var baseret på idéen om at udvikle et *personligt dynamisk medie (PDM)*; en personlig computer, hvis dynamiske aspekt bl.a. lå i at den enkelte bruger kunne forme computerens funktion og muligheder efter egne behov. Denne egenskab var implementeret i systemet, via det modulære programmeringsmiljø *Smalltalk*, hvis formål var at lade brugeren udvikle egne redskaber og anvendelsesmuligheder, vel at mærke via delvist grafiske brugergrænseflader. *PDM* visionen var med andre ord kendetegnet ved, at mediets fleksibilitet ikke udelukkende angik, de data brugeren bearbejdede, men ligeledes var møntet på selve systemets udformning og funktionalitet. *Smalltalk* var samtidig baseret på idéen om at skabe et lettilgængeligt system, som selv børn kunne bruge, men som sideløbende fostrede en kognitiv læringsproces, der kultiverede brugerens forståelse og muligheder i relation til computerbrugen. Selvom *PDM* visionen kan forekomme utopisk i dag, så repræsenterer den alligevel en fundamentalt anderledes forestilling om, hvad computermediet kunne være blevet. Hvor *DynaBook* projektet er kendt for at forudsige og grundlægge mange af idéerne bag den moderne Pc, så fremhæver tilbageblikket på *PDM* visionen på mange måder et uforløst teknologisk potentiale i nutidens softwareprodukter. De grundliggende idéer om et dynamisk medie, som *Goldberg* og *Kay* tænkte det, er ikke realiseret. Med udgangspunkt i selv samme idéer - udviklet på *Xerox PARC* - har computeren i stedet taget form som et medie, der nok giver den almene bruger et utal af anvendelsesmuligheder, men det er samtidig muligheder, der langt hen ad vejen er låst fast i softwareindustrielle pakkelsesløsninger.

Dette skift er tæt forbundet med den brede markedsføring af den personlige computer, der blev grundlagt i firserne og eksploderede i halvfemserne med udbredelsen af internettet. Massekommercialiseringen bragte et krav om brugervenlighed, og med dette krav forduftede idéen om brugeren, som én der ”programmerede” computeren. De mere utopiske forestillinger om et medie, der var

⁷² Kay & Goldberg, 1977, s. 393-404

fleksibelt - eller demokratisk - i brugerens mulighed for at modellere både form og indhold, bristede under fascinationen af letfordøjelige pakkelsesløsninger. Dette resulterede i en markant adskillelse mellem kodebruger og GUI-bruger, eller som *Florian Cramer* konkluderer om udviklingen: "*Usage became graphical, programming textual*".⁷³ Undervejs i massekommercialiseringen af computermediet blev der, med andre ord, etableret et veldefineret skel mellem software-producent og slutbruger. Softwareproducenterne programmerede redskaberne, og slutbrugerne benyttede sig af de muligheder de frembød.

Man kan opfatte viraenes "indbrud" i den kommercielle software, som forsøg på at nedbryde dette skel. Viruskoden, som kiler sig ind og modificerer computerens operativsystem og programmer, rummer en insisteren på ikke udelukkende at udtrykke sig via softwarens præformaterede indholdskanaler, men ligeledes gennem dets struktur og form. De tilkendegiver på denne måde en modstand mod at blive reduceret til passiv forbruger af færdige softwarepakkelsesløsninger.

Interfacekritik

Det er ikke nogen hemmelighed, at virusscenen generelt har et anstrengt forhold til den massekommercielle softwareindustri og i særdeleshed *Microsoft*. Dette er gennem tiderne kommet tydeligt til udtryk gennem en lang række virusbårne smædebeskeder og *DDoS-angreb* rettet mod *Microsoft* og andre softwaregiganter.⁷⁴ *Clive Thompson* beskriver i sin artikel "*The Virus Underground*" viruskrivernes syn på *Microsoft*:

*"Virus writers are especially hostile toward Microsoft (...) Microsoft markets its products to less expert computer users, cultivating precisely the sort of gullible victims who click on disguised virus attachments. (...) By relying so exclusively on Microsoft products, virus authors say, we have created a digital monoculture, a dangerous thinning of the Internet's gene pool."*⁷⁵

Blandt viruskriverne er der, som i *hacker*, *free-software* og *open source* miljøerne, generelt en foragt for *Microsofts* forsøg på at monopolisere markedet, samt deres indkapsling af softwarens kildekode i intellektuel ejendomsret. Det er dog, som

⁷³ Cramer, 2003, s. 100

⁷⁴ Distributed Denial of Service.

⁷⁵ Artiklen indeholder en række interviews med aktører på virusscenen og fra antivirus-industrien. Clive Thompson "The virus underground" 2004 (ingen sidetal)

Thompson påpeger, ikke udelukkende disse aspekter, viruskriverne finder problematiske, det er ligeledes *Microsofts* designmæssige leflen for uerfarne brugere via lettilgængelige softwarepakkeløsninger. *Microsofts* brede markedsføring af ét universelt system til alle brugere og formål indebærer, ifølge viruskriverne, uundgåeligt, at man sigter efter laveste fællesnævner, hvad angår computerbrugernes teknisk formåen og indsigt. Markedsføringens mantra om brugervenlighed for *én og alle* får her igennem dyb indflydelse på den digitale kultur. I den forbindelse kan man ligeledes pege på en udtalelse, viruskriveren *Opic* kom med i et interview i 1999:

*"Virus writers would not be such a threat to the computing public if it weren't for the constant dumbing down of pc technology – a result of the ever increasing consumer demand for more automated, user friendly software and the eagerness of companies like Microsoft to meet it."*⁷⁶

I forlængelse af *Opic's* udtalelse må man huske, at målet i HCI-orienteret softwaredesign først og fremmest har været at lære computeren at forstå mennesket og ikke omvendt. Med GUI-systemerne behøvede brugerne ikke længere forstå tekstbaserede kommandoer, kode eller grundlæggende processer for at kunne benytte en computer. Samtidig har jagten på brugervenligheden medført en tendens til, at stadig flere af computerens aktiviteter er blevet automatiseret og skjult under interfaces overflade. Hvor det at afkode og forstå computerens arbejdsprocesser var en hverdags beskæftigelse for tidlige computerbrugere, blev det undervejs i GUI-systemernes evolution primært en aktivitet for softwareproducenter. Dette kan, som *Opic* påpeger, opfattes som en fordummelse af teknologien, idet massen af computerbrugere i stadig højere grad har mistet fornemmelsen af, hvad der foregår under den grafiske brugergrænseflades niveau. Softwaredesign efter disse modeller findes betænkelig, fordi den sjældent kultiverer nogen erkendelse af, hvordan en computer grundlæggende fungerer. Den fremavler i stedet en overfladisk teknologiforståelse, som samtidig binder brugerne til den kommercielle softwareindustri produkter. Med softwareindustriens nærmest ubetingede fokus på brugervenlighed kan man, med en bagvendt genbrug af *Donald Normans* slogan, sige at *interfacet er en forhindring*, i tilegnelsen af en helt grundlæggende computerteknisk viden og forståelse. Den almene computerbrugers ringe forudsætninger for at forstå, hvor og hvordan en digital

⁷⁶ Neely, 1999 (ingen sidetal)

virus angriber eller opererer - for slet ikke at tale om dennes mulighed for at opdage, lokalisere eller fjerne en virus på egen hånd - kan opfattes som et symptom på denne udvikling. I brugerens manglende viden om computerens grundlæggende opbygning og mekanismer, kan offeret udelukkende respondere på virusangreb ved at investere i antivirus-industriens kommercielle ydelser.

Ovenstående udtalelser fokuserer kort sagt på, at software ikke er neutrale anordninger. Virusskriverne mener, at softwarens udformning har indflydelse på computerbrug og teknologiforståelse, samt at de færdigpakkede softwareredskaber har en tendens til at forme brugerne som forbrugere af kommercielle softwareprodukter. I lyset af dette tager viraenes omdefineringer af softwarens form og logik sig ud som ganske interfacekritiske udsagn. Med virusskriverens aversion mod det brugervenlige, fremstår iværksættelsen af de forstyrrende, irrationelle, modvillige og misledende effekter som en direkte kritik af den grafiske brugervenlighed. Samtidig kan man ræsonnere, at denne kritik er bundet op på ideologiske og politiske problemstillinger, idet systemernes udformning anses for at være nøje tilrettelagt af kommercielle interesser, vel at mærke håndhævet af magtfulde softwareoligopolier.

Digital modkultur

I forlængelse af ovenstående er det relevant at fremhæve, at hacker-kulturen, som virusscenen jo udspringer af, netop er rodfæstet i forestillinger om teknologiens frigørelsespotentialer bl.a. i forhold til den kommercielle kultur. Disse frihedsideal, som de f.eks. er beskrevet i *Steven Levy's* bog "*Hackers: Heroes of the Computer Revolution*" fra 1984, bestod op igennem halvfemserne i form af forskellige modkulturelle strømninger – måske karest manifesteret i teksten "*A Declaration of Independence of Cyberspace*" (1996) af *John Perry Barlow* fra *Electronic Frontier Foundation*. Tekstens kendte indledning lyder som følger:

*"Governments of the Industrial World, you weary giants of flesh and steel, I come from Cyberspace, the new home of Mind. On behalf of the future, I ask you of the past to leave us alone. You are not welcome among us. You have no sovereignty where we gather"*⁷⁷

Halvfemserne emmede i visse kredse af en tiltro til, at computermediet ville bringe en teknologisk revolution, som ville frigøre mennesket fra undertrykkende kræfter og

⁷⁷ Barlow, 1996

bane vejen for social forandring i kraft af de demokratiske potentialer teknologien bragte. Mens den massive udbredelse af internettet i midthalvfemserne pusede liv i disse frihedsidealer, demokrati-utopier og hacker-ideologier, så var det samtidig en periode, hvor computerkulturen for alvor stabiliserede sig i stramme kommercielle rammer. Mens bl.a. freeware-, open source- og hackerkulturerne var forankret i overbevisningen om, at information måtte være fri, åben og tilgængelig, så arbejdede økonomiske kræfter på en kommercialisering og privatisering af den digitale sfære. For at den industrielle magtbalance kunne bevares i den digitale æra, måtte informationsstrømmen kontrolleres igennem kommercielle kanaler, og det blev den i heftigt ekspanderende grad i denne periode. Det var med andre ord en tid, hvor subkulturelle ideologier var på kollisionskurs med den markedsøkonomiske realitet. De modkulturelle frihedsidealer var en trussel mod den markedsøkonomiske orden og vice versa. I den forbindelse er det påfaldende, hvordan *Microsoft* beskriver open source licenserne på deres hjemmesider:

”These licenses are described as viral because they ’infect’ derivative programs, but vary in terms of how they define which programs are derivative works. However, the dominant open source license – the General Public License (GPL) – is considered one of the most infectious OSS licenses”⁷⁸

Mens *Microsofts* retorik mere end antyder, at open-source kulturens frihedsidealer er en virus i det kommercielle system, så må de mere idealistiske modkulturelle softwarebevægelser have betragtet kapitalismen som en virus, der spredte sig i den digitale kultur op igennem firserne og halvfemserne. Bemærkelsesværdigt nok var det med en lignende “*virus of liberty*” retorik, *John Perry Barlow* beskrev teknologiens frihedskræfter i tidligere omtalte manifest. Det viste sig dog snart at være de mere profitorienterede idéer, der spredte sig mest effektivt, efterhånden som informationsøkonomien tog form i netværkssamfundet.

Modstand og aktivisme

Den nærmest eksplosive stigning af virusforekomster, halvfemserne bød på, kan i dette lys opfattes som en reaktion på disse ideologiske friktioner. Virusaktiviteten kan

⁷⁸ “Microsoft’s Perspective on Open Source and the Commercial Software Industry”: http://download.microsoft.com/documents/australia/about/microsoft_commercial_software_in_australia.doc [02-08-2008]

læses som en modstand mod softwareindustriens indsnævring af computerens anvendelses muligheder i fastlagte kommercielle rammer. Viruskodens selvspredende algoritmer fungerede som en sidevej, der navigerede uden om de officielle kanaler, for på denne måde at gennemtvunge informationens frie bevægelighed. Den implementerede så at sige idéen om en netværksbaseret udveksling, som ikke var medieret af profitorienterede mekanismer. Viruskreationen tog sig på denne måde ud som en slags elektronisk civil ulydighed, der eksekveredes i en digital kultur, der i stadig stigende grad kontrolleredes af økonomiske interesser. Med brug af *Deleuze og Guattari's* retorik kan man beskrive virus- og hackeraktiviteten, som en *rhizomatisk-nomadisk modstand* rettet mod kapitalens herredømme over den digitale kultur.⁷⁹ Det vil sige en dynamisk, decentraliseret og ikke-hierarkisk aktivistisk modstand, der bestræbte sig på at opretholde computerteknologien og internettet som en platform for horisontal kommunikation, bl.a. gennem forstyrrende interventioner i netværkskulturens etablerede magtstrukturer.

Hvor denne modstand kan udformes ganske kontant, med *DDoS-angreb* eller andre mere eller mindre terrorlignende foretagender, så er der noget andet på spil i den gren af digitale vira, denne tekst har behandlet. De grafiske payloads udfører snarere en slags symbolske eller semiotiske angreb, som er rettet mod den måde, computerbrugeren ser og forstår computeren via dens grænseflader. Disse interventioners subversive ombrydninger af den grafiske brugergrænseflades metaforer og funktionalitet synes i sjældnen grad at ville omstyrte selve systemet, men snarere de forestillingsregimer, det implementerer. Mens interfacets polerede facader har en tilbøjelighed til at afstive dets egen sandhedsværdi, så forstyrres disse forestillinger med viraenes interventioner i den grafiske brugergrænseflades form og logik. Antager vi, at de grafiske payloads netop er udtryk for en sådan semiotisk modvæрге, så har denne virusaktivitet en slående lighed med den såkaldte *culture jamming* aktivisme.

Culture jamming, som blev bredt kendt med *Marc Derys* pamflet "*Culture Jamming: Hacking, Slashing and Sniping in the Empire of Signs*" fra 1993, omfatter en lang række aktiviteter, der har det tilfælles at de søger at transformere massemediernes og populærkulturens billeder, udsagn og artefakter, med henblik på at undergrave deres betydning eller sandhedsværdi. Dette gøres gerne gennem ironiske

⁷⁹ Deleuze & Guattari, 1980, s. 3-25

eller satiriske modifikationer, hvis mål er på subversiv vis at afdække den kommer-
cielle kulturs visuelt forførende kræfter. *Culture jamming* kendetegnes, med andre
ord, som en modstand mod forbrugerismens hegemoni med den kommercielle kulturs
egne midler. *Marc Dery* formulerer modstanden således:

“(Culture jamming) is directed against an ever more intrusive,
instrumental technoculture whose operant mode is the manufacture of
consent through the manipulation of symbols”⁸⁰

Som ovenstående formuleringer gerne skulle gøre klart, så er denne praksis nært
beslægtet med de strategier, jeg har argumenteret for, at de grafiske payloads dækker
over. Hvor *culture jamming* aktivisterne typisk intervenserer i det offentlige rum, f.eks.
med billboard modifikationer o. lign., så intervenserer viraene i det digitale rum.
Begge fænomener rummer en bevidsthed om billedernes manipulerende magt og en
modstand mod denne, udført med deres egne midler. Med dette trækker begge
praksisformer ligeledes en stærk reference til *situationist-avantgardisternes*
detournement strategi. *Detournement* handlede om at fravriste massemediernes
billeder deres betydning, ved at afmontere dem fra deres oprindelige kontekst og
remontere dem i en fragmenteret meningsløs sammenhæng. Målet var at afsløre
massemediernes altid medierede repræsentationer af verden, hvilket f.eks. ses i *Guy*
Debords film *La société du spectacle* fra 1973. Situationisternes mediekritik bundede
i forestillingen om, at verden er fragmenteret og derfor ikke lader sig gengive i
narrative fremstillinger, som sammenkæder virkelighedens adskilte data.⁸¹ De mente
derimod, at medierne passiviserer og fremmedgør mennesket fra virkeligheden
gennem deres falske repræsentationer af en opsplittet verden.

Sidestiller man viruskriveriet med disse aktivistiske og avantgardistiske strategier, er
det rimeligt at betænke, hvorvidt aktiviteten faktisk rummer en ideologisk motivation,
eller om den snarere skal kategoriseres som simple drengestreger. Hvor valget af
citater fra viruskrivere i denne tekst peger i en politisk bevidst retning, så vil man
dog ligeledes finde ganske ureflekterede udtalelser i interviews med Vx-scenens
aktører. Det er da også blevet betvivlet, hvorvidt hacker- og virusaktivitet overordnet

⁸⁰ Dery, 1993 (ingen sidetal)

⁸¹ Bolt (ingen årstal eller sidetal)

set skulle være ideologisk motiveret - f.eks. af sociologen *Sarah Gordon*⁸², som via sine interviewbaserede feltstudier konkluderer, at en stor del af scenens aktører i halvfemserne var amerikanske teenage-middelklassedrenge uden synderligt veludviklede politisk standpunkter og indsigter.⁸³ Med dette kategoriseredes virusprogrammeringen, som en aktivitet uden andet incitament end den skadefromorskab, der ligger i den vandalistiske handling. Det forekommer mig dog lidt for forsimplet at infantilisere virus- og hackeraktivitet på denne måde. Man kan stille spørgsmålet om, hvorvidt aktiviteten skulle være mindre politisk, fordi den ikke er ideologisk udsagt eller (vel)formuleret. Dette argument griber kulturteoretikeren *Andrew Ross* ligeledes fat i, i essayet "*Hacking away at the Counterculture*", hvor han argumenterer for hackerne som den elektroniske tidsalders "*rebels without a cause*".⁸⁴ Det vil sige, at de repræsenterer en ungdomskultur, der nærmest instinktivt gør oprør mod den digitale kulturs autoriteter, normer og regelsæt.

Hacker-mentalitet og computervirus-æstetik

Min pointe er, at den interfaceæstetik, de grafiske payloads aftegner, netop afspejler en sådan modreaktion. Payload-effekterne er forankret i selve eksperimentet i at få systemet og dets brugergrænseflader til at gøre noget, det ikke er designet til, for på denne måde at bryde de rammer og begrænsninger, systemet udstikker. Overføres *Ross'* anskuelsesvinkel til virusscenens aktiviteter, kan man udlægge dem som en spontan modreaktion i en instrumentaliseret computerkultur, der i højere og højere grad fikses i bestemte modeller for teknologi-anvendelse og forståelse. Ser man på den oprindelige betydning af hackerbetegnelsen, da den opstod blandt studerende på *MIT* i starten af tresserne, havde den på dette tidspunkt, hverken nogen kriminelle eller decideret ideologiske konnotationer.⁸⁵ En hacker var på dette tidspunkt en person, som hengav sig til opfindsom og ukonventionel brug af teknologi.⁸⁶ Hacker-mentaliteten var kendetegnet ved den legende, nysgerrige og mere eller mindre formålsløse eksperimenteren med computerteknologien og var sikkert nærmere knyttet til begreber som innovation og forskning end de betydninger, den senere er

⁸² Sarah Gordon er ansat som forsker i virusskriver-kultur ved IBM.

⁸³ Gordon, 1994 (ingen sidetal)

⁸⁴ "Rebel without a cause" fænomenet er personificeret med drop-out karakteren Jim Stark spillet af James Dean i filmen af samme navn fra 1955.

⁸⁵ Den gængse opfattelse af hackeren som en kriminel person, der bryder ind i digitale netværk osv., opstod først rigtigt i løbet af firserne, hvor der forekom en lang række tilfælde af denne slags.

⁸⁶ Som beskrevet af Ross, 1990 (ingen sidetal)

blevet pålagt.⁸⁷ Hvad der er interessant i forhold til de grafiske payloads, er at denne hacker-mentalitet ligeledes udfolder sig i en ukonventionel brug af den grafiske brugergrænseflades formsprog og metaforik. De repræsenterer en uortodoks leg med interfacets betydningsdannelse og beskuerens reception, vel at mærke i forhold til de forståelsesrammer computerens software udstikker. På denne måde tager de sig snarere ud, som en undersøgelse af den måde vi ser og forstår computeren, end som en decideret undersøgelse af teknologiens anvendelsesmuligheder. De grafiske payloads iværksætter skæve perceptuelle vinkler via fornuftsstridige softwaremutationer, som bringer computerbrugeren i et konfliktforhold med etablerede forståelsesrammer, men som ikke i sig selv peger på nye opbyggelige alternativer. Det er altså en praksis, hvis primære formål synes at være at bryde perceptuelle mønstre og vedtægter, for her igennem at åbne op for en genforhandling af den måde beskueren ser og forstår det givne system. Det interessante er i den forbindelse at dette greb, hvor det æstetiske udsagn søger at lave en perceptuel eller kognitiv åbning, ligeledes er vidt udbredt indenfor de mere konceptuelt orienterede kunstarter. Hackerens tilgang til sit *materiale* minder altså til forveksling om en kunstnerisk tilgangsvinkel. Denne parallel har net-kunstneren og hacker-teoretikeren *Cornelia Sollfrank*, som selv arbejder med hacker-inspireret kunst, ligeledes iagttaget. Hun opfatter hacking som en strategi, der bærer stærke referencer til nyere kunstneriske praksisformer:

*” if you take a closer look at the term ‘hack,’ you very quickly discover that hacking is an artistic way of dealing with a computer. So, actually, hackers are artists ”*⁸⁸

Om hackere er kunstnere, som *Sollfrank* her foreslår, skal være usagt. Der er ikke noget, der tyder på, at hackerne har betragtet sig selv som kunstnere eller haft den mindste interesse i at få påhæftet sig dette prædikat, hvormed man kan stille spørgsmålstejn ved, om aktiviteten kan kategoriseres som sådan. Alligevel er parallellen mellem disse praksisformer helt klar, og det er denne, der kommer meget tydeligt til udtryk med viraenes grafiske payloads. Når den eksperimenterende hacker-

⁸⁷ Nogle vil endog holde på, at det er denne hacker-mentalitet, der har drevet den tidlige udvikling af computerteknologien frem. F.eks. skriver Eric Steven Raymond: “Hackers built the Internet. Hackers made the Unix operating system what it is today. Hackers run Usenet. Hackers make the World Wide Web work.” (Raymond 2001)

⁸⁸ Telemann (ingen årstal eller sidetal). Cornelia Sollfrank har bl.a. arbejdet med socialrelational hacking og kunstinstitutionel hacking. Se evt. Tilla Telemann interview med Sollfrank.

mentalitet manifesterer sig i den grafiske brugergrænseflade, som den gør med disse sansbare effekter, konstituerer den en interfaceæstetik, som har iøjefaldende referencer til de nyere billedkunstneriske udtryksformer – specielt de mere postmoderne af slagsen. Denne parallel er selvfølgelig blevet ekstra distinkt efter fremkomsten af den softwarekunstneriske genre, idet den benytter sig af det samme teknologiske materiale som de digitale vira. Med dette kan man ræsonnere, at computervirussen markerer en interfacebaseret æstetik, som siden hen er blevet videreført indenfor den softwarekunstneriske diskurs.

Kapitel 4: Net- og softwarekunsten

Software-kunst og kritik

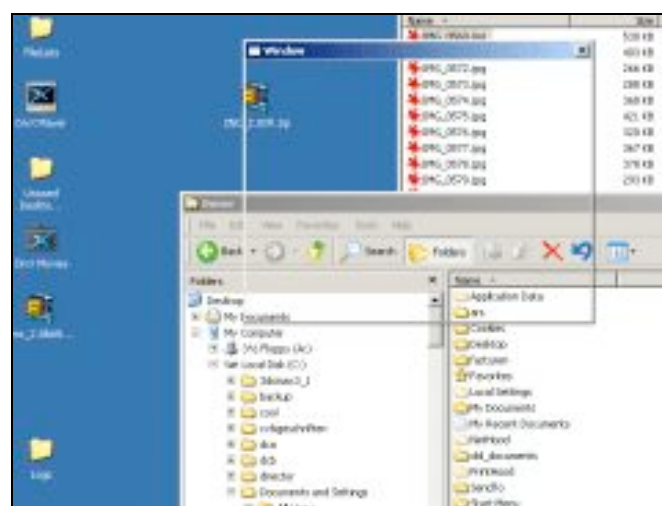
Forrige afsnits konklusion vil jeg afslutningsvis underbygge ved at kigge lidt mere konkret på analogien mellem softwarekunsten og computervirussen. Et helt centralt slægtskab findes i den kritik, der ligger i disse praksisformers manifestationer. Der ligger implicit en kritik af det bestående i den måde viraene forsøger at omdefinere eller udvide softwarens konventioner og ”offerets” rammer for forståelse, denne type kritik er ligeledes kendetegnende for net- og softwarekunsten. I *Matthew Fuller's "Behind the Blip: Software as Culture (Some Routes into 'Software Criticism', More Ways Out)"* efterlyser han en softwarekritik, som ikke er baseret på det sædvanlige snævre fokus på *brugertests og bithastigheder*, dvs. en kritik, som er bevidst om den kulturelle og samfundsmæssige sammenhæng, softwaren er en del af og har indflydelse på. I den forbindelse peger han på, hvad han kalder de kritiske praksisformer indenfor softwareproduktion. Disse kendetegnes som eksperimenterende, ikke-funktionalistiske og spekulative produktioner, der forholder sig reflekterende til softwarekulturens æstetik og politik. Det er software, der søger at mutere eksisterende tekniske, designmæssige, perceptuelle og konceptuelle aspekter af software, for her igennem at udsætte dem for kritisk analyse. De søger at udfordre eksisterende softwarekonventioner eller bestræber sig på at hive tæppet væk under gængse forståelser af software. Dette gøres ofte via ironiserende eller sarkastiske modifikationer af eksisterende software. Et greb som *Fuller*, med en reference til *Roland Barthes' frase: "Sarcasm is the condition of truth"*, formulerer således:

”Critical software is a voyage into that truth by means of its own devices.”⁸⁹

Hvor *Fullers* definitioner her er møntet på net- og softwarekunsten, er det dog ikke svært at argumentere for, at den gren af computervirusfænomenet denne opgave har behandlet, kan opfattes som en sådan softwarekritisk praksisform. De grafiske payloads bærer, med deres ikke-funktionalistiske, irrationelle og sarkastiske software modifikationer, de kendetegn, *Fuller* opridser. Softwarekunstens og computer-virusens praktisk formulerede softwarekritik har med andre ord en slående lighed.

Softwareværker: Window, Giant Cursor og WIMP

Ser vi på softwareværker, som i mine øjne bærer åbenbare referencer til den grafiske payloads æstetik, kan f.eks. nævnes *Peter Luinings* operativsystem-manipulerende værker. Med værket *Window*⁹⁰ reflekteres over vindueskonceptets udformning, funktion og betydning i operativsystemets kontekst. *Luining* kreerer, når man eksekverer hans program, et vindue på skærmen, som fungerer efter det virkelige vindues principper. Det vil sige, at det er ”gennemsigtigt”, hvormed man kan se, hvad der ligger ”bag” vinduet; ikoner, mapper, skrivebordsbilleder og andre vinduer osv. Værket spiller på den betydningsmæssige ambivalens mellem det fysiske og digitale vindue, samtidig med at det peger på, at den grafiske brugergrænseflade, på trods af de transparensforestillinger der flourerer omkring det, grundlæggende betjener sig af medieringer; at ”gennemsigtigheden” er et resultat af kodede bearbejdnings.



Window af Peter Luining (2003)

⁸⁹ Fuller, 2003, s. 22-23

⁹⁰ Peter Luinings værker findes på: <http://works.ctrlaltdel.org/> [02-08-2008]

Med værket *Giant Cursor* forstørres *Luining* musepilen i *Windows* operativsystemet, således at den optager en god portion af skærbilledet og på denne måde bliver både u håndterlig og ganske upraktisk. Med dette indgreb i systemets visuelle arkitektur forstyrres det repræsentationelle lag det funktionelle lag i en sådan grad, at systemet praktisk talt bliver ubrugeligt.

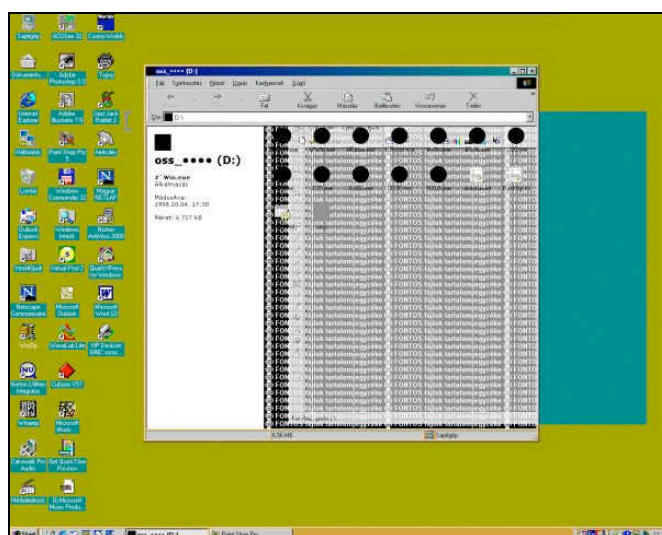
Et andet kodet kunstværk, der tager afsæt i operativsystemets grafiske opbygning, er *DX-Labs* (*Alexei Shulgin* og *Victor Laskin*) *WIMP* projekt. *WIMP* er efter skabernes eget udsagn et VJ-redskab beregnet til at skabe *real-time visuals* til koncerter og DJ-performances.⁹¹ *WIMP* udmærker sig ved, at disse visuelle effekter udgøres af *Windows-systemets* egne grafiske elementer. *WIMP* lader ved hjælp af tastatur-genveje brugeren manipulere med operativsystems grafiske arkitektur, for på denne måde at tilvejebringe spidsfindige animerede effekter; ikoner kravler hen over skrivebordet, og system-, mappe- og programvinduer hopper, ryster, flyver, falder og kolliderer rundt på skærmen, alt efter hvilke taster brugeren vælger at trykke på. *WIMP* transformerer operativsystemets form og formål, samtidig med at værkets iscenesættelse som et VJ-redskab, rummer en humoristisk kommentar, idet det synes at påpege, at *Windows* nu er blevet så meget overflade, at vi kan danse til det på diskotekerne.

Ovenstående eksempler kan betegnes som formelle interventioner i operativsystemets form og funktionalitet. Kunstneren er én, der manipulerer systemets repræsentationelle lag og dets metaforer, for her igennem at dekonstruere dets betydning og påpege interfacets altid medierede karakter. Værkerne omdefinierer grænsefladens fasttømrede logik og åbner på denne måde op for en refleksion over systemets konstruerede natur. De er forankret i at få operativsystemet til at gøre ting eller indtage former, det ikke er designet til, og med dette holder de en klar æstetisk og strategisk parallel til den gren af vira, denne tekst har fremdraget. Endelig kan man sige, at ovenstående værker på mange måder rummer den samme slags humor, som de grafiske payloads benytter.

⁹¹ I forbindelse med *WIMP* softwaren skal det nævnes, at den senere er blevet omprogrammeret til et mere traditionelt VJ-redskab, der kan afspille videofiler o. lign. Softwaren havde dog oprindeligt form som beskrevet i denne tekst. Dette kunne bl.a. opleves på read me 2004 festivalen i Århus, hvor Shulgin og Laskin lavede en VJ/DJ performance med *WIMP*. <http://wimp.ru/about.php> [02-08-2008]

Jodi: OSS

En anden værkpraksis, som er værd at fremhæve i forbindelse med computervirussen, er den, der har taget form i kunstnerduoen *Jodis* omfattende produktion. *Jodi* dyrker, langt hen ad vejen, hvad man kan kalde en systemsammenbruddets æstetik. De elsker at vende op og ned på den grafiske brugervenlighed ved at gøre brugergrænsefladens form og funktionalitet irrationel, ustyrlig, kaotisk og fragmenteret. De har en svaghed for at *vende vrangen ud på systemet* ved at lade dets kode og underliggende mekanismer trænge frem i interfacets overflade, for hermed at gøre det fremmedartet og uforståeligt. Et værk jeg specielt vil fremhæve er *OSS*. Indgangen til dette værk findes på hjemmesiden oss.jodi.org. En side, som i et heftigt tempo lader små browser-vinduer poppe op og forsvinde tilfældige steder på skærmen. Under forsøget på at interagere med disse vinduer begynder siden at *downloade* en række små programmer til den besøgendes computer. Hvert af disse programmer udfører, hvis modtageren vælger at eksekvere dem, grafiske og funktionelle modifikationer af *Windows* operativsystemet. Et af programmerne medfører f.eks. at skrivebordet ”ryster”, et andet at skærbilledet ”ruller” som på et defekt tv, et tredje skriver uforståelig kode og udfører form/farve modulationer på operativsystemets grafiske arkitektur, og et fjerde omdanner musepilen til en ”blyant”, der tegner ovenpå interfacets elementer.



En af Windows-modifikationerne i værket OSS af Jodi.

Jodis brug af systemsammenbruds- og fejlsimulation, deres iscenesættelser af kunstbeskuerens – eller måske snarere *kunst-offerets* - kontroltab og deres brug af

kode som et fremmedartet element i den grafiske brugergrænseflades velkendte rammer, trækker stærke referencer til den computervirusæstetik, denne opgave har beskrevet. I den forbindelse kan man, som en lille anekdote, nævne, at den hjemmeside, der indeholdt *OSS* projektet, på et tidspunkt blev midlertidigt lukket ned, fordi netværksudbyderen troede at *OSS* var en computervirus.⁹²

GWEI - Google Will Eat Itself

Endelig vil jeg nævne, at der er en gren af nyere net- og softwarekunstværker, der trækker referencer til computervirusfænomenet på mere taktisk vis. Hvor jeg tidligere har argumenteret for, at man kan læse de digitale vira som interventioner i den digitale kulturs etablerede magtstrukturer, så har lignende strategier manifesteret sig i de net- og softwarekunstneriske diskurser. Tænk f.eks. på *GWEI - Google Will Eat Itself* projektet, der søger at opkøbe aktieselskabet bag søgemaskinen *Google* via dets eget annoncesystem *AdSense*. Strategien beskrives således på projektets hjemmeside:

“We generate money by serving Google text advertisements on a network of hidden Websites. With this money we automatically buy Google shares. We buy Google via their own advertisement! Google eats itself - but in the end ‘we’ own it!”⁹³

Med denne mekanisme vil *Googles* penge-genererende museklikks-reklamesystem i princippet opkøbe sig selv. Endemålet er, at selskabet *GTTP Ltd. (Google To The People Public Company)*, der opsamler de opkøbte aktier, skal distribuere ejerskabet tilbage til brugerne – dvs. offentligheden. *GWEI* kiler sig som en parasit ind og udnytter *AdSense* systemets egne mekanismer mod systemet selv og kan på denne måde opfattes som en virus i *Googles* annoncesystem.

Afslutning

Mens det er udbredt udelukkende at opfatte computervirussen som en teknisk og økonomisk trussel, har jeg med dette speciale søgt at tilbyde en anden og måske mere nuanceret læsning af fænomenet. Jeg har påpeget, at ikke alle virusforekomster entydigt kan kategoriseres som udtryk for cyber-terror, økonomisk kriminalitet eller vandalisme, og i stedet reflekteret disse som kulturelle konstruktioner, modstands-

⁹² Ifølge Cramer, 2005, s. 97

⁹³ *GWEI* er skabt af Hans Bernhard (Ubermorgen.com), Alessandro Ludovico og Paolo Cirio.
<http://gwei.org/index.php> [02-08-2008]

handlinger, softwarekritik og æstetiske kreationer. Disse computervira er ikke kun tekniske entiteter, konstrueret for at genere vores digitale liv og færden, de afspejler ligeledes strømninger og modforestillinger i den digitale kultur. Modforestillinger som er forankret i en kritisk stillingtagen til det etablerede, en modstand mod at blive reduceret til passiv forbruger og en eksperimenteren med hjemmestrikkede teknologiske utopier. Med dette udgør de et kritisk potentiale i en kultur, hvor vores teknologiske muligheder, forestillinger og utopier er nøje tilrettelagte i stramme kommercielle rammer - rammer som er behændigt indkapslet i intellektuel ejendomsret. Computervirussen er med andre ord en vigtig del af computermediets kulturelle historie og kan ikke bare afskrives som et uheldigt kapitel i denne.

Endvidere har jeg ræsonneret, at en gren af de digitale vira danner en ganske veldefineret interfaceæstetisk diskurs, som fra en kontemporær kunstrelateret synsvinkel forekommer yderst moderne. I forbindelse med softwarekunsten er det blevet diskuteret, om man skal finde denne genrens kunstneriske forgængere indenfor bl.a. tressernes video- og konceptkunst eller den tidlige - maskinkodede - formalistiske computerkunst, som den f.eks. tog form i computerkunstpionererne *Frieder Nakes*, *Georg Nees* eller *A. Michael Nolls* værkproduktioner.⁹⁴ I mine øjne har den kategori af computervira, denne opgave har behandlet, dog en langt mere markant lighed med den softwarekunstneriske værk- og praksisform - både formelt, indholdsmæssigt og strategisk set - end nogen af disse kunstneriske forgængere. Slægtskabet er kendetegnet ved den samme eksperimenterende, mediebevidste og reflekterende tilgang til computerteknologien og kommer ydermere til udtryk i dé interfacebaserede greb og virkemidler, disse kodede kreationer benytter.

Om dette æstetiske slægtskab er udsprunget af inspiration eller tilfældighed er svært at afgøre, men med blik på de softwarekunstneriske kreationer forekommer det mig dog ikke usandsynligt, at nogle af disse kunstnere kan have tilbragt deres unge år med at eksperimentere med viruskode og grafiske payloads.

Men hvad blev der egentlig af disse grafiske payloads, denne tekst har beskrevet?

Antivirus-producenten *Kaspersky Labs* erklærede 2003 for *ormens år*⁹⁵ begrundet i, at alle de virusforekomster, der nu fandtes på *in the wild* listerne, var højest professionelt

⁹⁴ Spørgsmålet om net- og softwarekunstens æstetiske forgængere er bl.a. blevet behandlet af Florian Cramer(2002), Jacob Lillemose(2004), Erkki Huhtamo(2003)

⁹⁵ Findes her: <http://www.viruslist.com/en/trends>

programmerede computer-orme med fuldstændigt veldefinerede gøremål og intentioner. De var designet til at stjæle penge og informationer, udføre angreb på hjemmesider eller som redskaber til at sprede spam og lignende. Denne udmelding markerede - efter alt at dømme - definitivt de mere eksperimenterende virusaktiviteters forsvinden. Læser man i Vx-scenens publikationer fra denne periode, vil man finde en frustration over scenens tilstand og aktørernes engagement – de bærer præg af en scene i opløsning. F.eks. kunne man i en af de mest aktive virusskrivergrupper 29A's virusmagasin fra 2002 finde selvransagende tekster med titler som *"The Virmaking is Dying"* og *"Situation in VX scene"*⁹⁶, hvor skribenterne reflekterede over, hvad der var gået galt med Vx-scenen. Der findes formentlig en hel række grunde til, at Vx-scenen stort set gik i opløsning på dette tidspunkt.⁹⁷ Jeg vil dog nøjes med at pege på et par stykker, som er relevante i forhold til denne teksts ramme. Opfatter man virusaktiviteten som en slags aktivisme rettet mod de kommercielle kræfter, der dominerer den digitale kultur, så har det udviklet sig til at være en temmelig problematisk en af slagsen. De digitale vira er gennem antivirus-industrien blevet inkorporeret i den digitale kulturs økonomiske orden, hvormed det at skabe og sprede virus snarere får karakter af at understøtte denne industri end at bekæmpe den. Et andet ganske selvmodsigende aspekt i relation til virus- og hackerkulturerne er, at hvor de er forankret i overbevisningen om at information må være fri og tilgængelig, så har aktiviteten på mange måder resulteret i det stik modsatte. Den digitale færden skal bestandigt godkendes og kontrolleres af computerbrugeren selv eller antivirus-softwaren, hvilket vil sige, at computerbrugeren strammes op i sikkerhedsforanstaltninger, der i bund og grund modarbejder informationens frie bevægelighed. Endvidere gøres stadig større dele af f.eks. Windows API'en (Application Programming Interface) utilgængelig for at forhindre potentielle sikkerhedshuller og lignende. Virus- og hackeraktiviteten har med andre ord nærmere resulteret i en indsnævring af muligheder og tilgængelighed end en åbning. Alt dette kan have været medvirkende til, at flere og flere virusskrivere valgte at skrinlægge deres aktiviteter.

⁹⁶ Skrevet af henholdsvis ZOMBIE og Benny fra gruppen 29A, som siden er blevet opløst:

⁹⁷ En af dem kan være at Microsoft i 2003 oprettede en fondspulje på fem millioner dollars, beregnet til dusører for informationer der kunne lede til pågribelsen af hackere og virusskrivere.

http://www.treasury.gov/usss/press/ms_110503.pdf. [04-02-2008]

Litteraturliste

Barker, Chris: "Cultural Studies", SAGE Publications 2008

Barlow, John Perry: "A Declaration of Independence of Cyberspace" 1996:
<http://homes.eff.org/~barlow/Declaration-Final.html> [02-08-2008]

Bolt, Mikkel: "Situationisternes antifilm" (ingen årstal forefindes.)
<http://www.copenhagenfreeuniversity.dk/antifilm.html> [02-08-2008]

Cohen, Fred: "Computer viruses – Theory and Experiments" 1984:
<http://all.net/books/virus/index.html> [02-08-2008]

Cramer, Florian: "Concepts, Notations, Software, Art" In: read_me 1.2 kataloget
2002: http://cramer.plaintext.cc:70/essays/concept_notations_software_art [02-08-2008]

Cramer, Florian: "Executable statements: The Insistence of Code" - In: Ars Electronica kataloget "CODE - The Language of Our Time" Hatje/Cantz (ed.) 2003
http://cramer.plaintext.cc:70/essays/executable_statements [02-08-2008]

Cramer, Florian: "Words Made Flesh" 2005: http://cramer.plaintext.cc:70/00-recent/words_made_flesh [02-08-2008]

Dawkins, Richard: "The Selfish Gene" Oxford University Press 1990.

Deleuze, Gilles & Guattari, Félix: "A Thousand Plateaus" Continuum International Publishing, 2004 (Minuit 1980)

Dery, Marc: "Culture Jamming: Hacking, Slashing and Sniping in the Empire of Signs" 1993: http://www.markdery.com/archives/books/culture_jamming/ [02-08-2008]

Deseriis, Marco: "Text Virus" - In: Fuller, Matthew (ed.) "Software Studies" The MIT Press, 2008

Dibbel, Julian: "Viruses are good for you" in: Wired Magazine 1995:
<http://www.wired.com/wired/archive/3.02/viruses.html> [02-08-2008]

Ellis, Brett Easton: "American Psycho", Picador, 1991

Engelbart, Douglas & English W.: "A Research Center for Augmenting Human Intellect" 1968
<http://sloan.stanford.edu/mousesite/Archive/ResearchCenter1968/ResearchCenter1968.html>

Erkki Huhtamo: "WEB STALKER SEEK AARON - Reflections on Digital Arts, Codes and Coders" In: Ars Electronica (kataloget), 2003:
http://www.aec.at/en/archives/festival_archive/festival_catalogs/festival_artikel.asp?iProjectID=12320 [02-08-2008]

Finneman, Niels Ole: "Computeren – Et medie for en ny skriftteknologisk revolution" In: Jens F Jensen (ed.) "Multimedier, Hypermedier, Interaktive medier" FISK-serien nr. 3, Aalborg Universitetsforlag, 1998

Fuller, Matthew: "Behind the Blip: Software as Culture (Some Routes into 'Software Criticism', More Ways Out)" 2003:
http://www.multimedialab.be/doc/citations/matthew_fuller_blip.pdf [02-08-2008]

Fuller, Matthew: "Grænsefladens umulighed" In: "Interfacekulturens æstetik" - Passepartout – Skrifter for kunsthistorie Nr. 14. Årgang 2007

Gordon, Sarah: "Inside the Mind of The Dark Avenger" In: Virus News International, 1993: <http://www.research.ibm.com/antivirus/SciPapers/Gordon/Avenger.html> [02-08-2008]

Gordon, Sarah: "The Generic Virus Writer" 1994:
<http://www.research.ibm.com/antivirus/SciPapers/Gordon/GenericVirusWriter.html> [02-08-2008]

Johnson, Steven: "Interface Culture – How New Technology Transforms the Way We Create and Communicate" Perseus Books Group, 1997

Kay, Alan & Goldberg, Adele: "Personal Dynamic Media" (1977) – In: The New Media Reader ed. Noah Wardrip-Fruin and Nick Montfort, MIT Press 2003

Kay, Alan: "User Interface" in Laurel, Brenda (ed.): "The Art of Human-Computer Interface Design" Addison Wesley, 1990

Laurel, Brenda (ed.): "The Art of Human-Computer Interface Design" Addison Wesley, 1990

Levy, Steven: "Hackers – Heroes of The Computer Revolution", Penguin Books 1994 (Doubleplay 1984)

Lillemose, Jacob: "A Re-Declaration of Dependence – Software Art in a Cultural Context it Can't Get out of" in "Read_Me – Software Art & Cultures", ed. Olga Goriunova & Alexei Shulgin, Center for Digital Æstetik Forskning, Århus Universitet, 2004

Ludwig, Mark: "The Giant Black Book of Computer Viruses" Amer Eagle Publications, 1995

Manovich, Lev: "The Language of New Media" , MIT Press, 2001

McLuhan, Marshall: "Understanding Media" Routledge Classics, 2001 (Routledge 1964)

Neely, Kim: "Notes from the virus Underground..." - In Rolling Stone magazine September 1999: <http://vx.netlux.org/lib/p0021.html> [02-08-2008]

- Neumann, John Von: "First Draft of a Report on the EDVAC." 1945.
<http://www.virtualtravelog.net/entries/2003-08-TheFirstDraft.pdf> [03-03-2008]
- Parikka, Jussi: "Digital Contagions – A Media Archaeology of Computer Viruses",
 Peter Lang Publishing, 2007
- Parikka, Jussi: "Digital Monsters, Binary Aliens - Computer Viruses, Capitalism and
 the Flow of Information" in Fibreculture Journal vol. 4: "Contagion and The Diseases
 of Information" 2005: http://journal.fibreculture.org/issue4/issue4_parikka.html [02-
 08-2008]
- Pold, Søren & Bertelsen, Lars Kiel: "Hvad viser skærmen? - Interfacet mellem
 referentiel og medial transparens" – In: "Kunsthistoriske kontakter" ed. Anne-Louise
 Sommer og Hans Jørgen Frederiksen, Aarhus Universitetsforlag, 2003
- Pold, Søren: "Preferences" In: Fuller, Matthew (ed.) "Software Studies" The MIT
 Press, 2008
- Raymond, Eric Steven: "How To Become A Hacker" 2001:
<http://www.catb.org/~esr/faqs/hacker-howto.html> [02-08-2008]
- Ross, Andrew: "Hacking Away at The Counterculture" In: "Postmodern Culture vol.
 1.1, 1990: <http://vx.netlux.org/lib/mar00.html> [02-08-2008]
- Shelby, D.D.: "The Viral Mind: Understanding the Motives of Malicious Coders"
 2002: <http://www.securityfocus.com/print/infocus/1583> [02-08-2008]
- Showalter, Elaine: "Hystories – Hysterical Epidemics and Modern Media" -
 Columbia University Press 1998
- Shulgin, Alexei & Goriunova, Olga (ed.): "Quickview on Software Art" – In:
 Read_Me 2.3 Reader, 2003
- Sollfrank, Cornelia: "Biennale.py – The Return of The Media Hype" 2001:
<http://www.heise.de/tp/r4/artikel/3/3642/1.html> [02-08-2008]
- Stojakovic-Celustka, Suzana (ed.): "The Legend - Fred Cohen Interview." In: Alive
 E-mag. vol.1.1, 1994 - <http://www.etext.org/CuD/Alive/alive01.01> [02-08-2008]
- Sutherland, Ivan: "SketchPad: A Man-Machine Graphical Communications System"
 (1963) – In: The New Media Reader ed. Noah Wardrip-Fruin and Nick Montfort,
 MIT Press 2003
- Telemann, Tilla: "Hackers are Artists - and Some Artists are Hackers" (Cornelia
 Sollfrank interview, intet årstal): <http://artwarez.org/femext/content/interviewEN.html>
 [02-08-2008]

Thompson, Clive: "The Virus Underground" – In: New York Times, February 8, 2004:
<http://query.nytimes.com/gst/fullpage.html?res=9A02E7D9143BF93BA35751C0A9629C8B63> [02-08-2008]

Turing, Alan: "Computing Machinery and Intelligence" (1950) – In: The New Media Reader ed. Noah Wardrip-Fruin and Nick Montford, MIT Press, 2003

Weinstock, Jeffrey: "Virus Culture" In "Studies in popular Culture" vol. 20.1, April 1997: <http://vx.netlux.org/lib/mjw01.html> [02-08-2008]

Tekster fra VX-magasiner:

Asmodeus: "Who creates viruses?" – In: Xine#5, 2000:
<http://vx.netlux.org/lib/vas00.html> [02-08-2008]

Benny: "Situation in VX scene" – In: 29A#6, 2002:
<http://vx.netlux.org/29a/29a-6/29a-6.110> [02-08-2008]

DecimatoR: "VX: What the Hell's happened?" – In: 40Hex vol. 2, 1992:
<http://www.etext.org/Zines/ASCII/40hex/40hex-9.001> [02-08-2008]

Lord Jules: "Persuing The Virus Author Mentality" – In: Vxtasy#1, 1999:
<http://vx.netlux.org/lib/vlj06.html> [02-08-2008]

MidNyte: "Why I write viruses (and how not to stop me)" 1999:
<http://vx.netlux.org/lib/vmn01.html> [02-08-2008]

n0ph: "Why writing viruses?" – In: Xine#4, 1999:
<http://vx.netlux.org/lib/vno00.html> [02-08-2008]

Spanska: "Some politically incorrect words about the so-called 'scene'" – In: Coderz#1, 2000: <http://vx.netlux.org/lib/vsa00.html> [02-08-2008]

ZOMBiE: "The Virmaking is Dying" – In: 29A#6, 2002:
<http://vx.netlux.org/29a/29a-6/29a-6.111> [02-08-2008]

Links til virusbeskrivelser:

Australian Parasite:	http://amigaviruses.wikidot.com/australian-parasite-the
Chernobyl:	http://en.wikipedia.org/wiki/CIH_virus
CyberAIDS:	http://en.wikipedia.org/wiki/Festering_Hate
Disk-herpes:	http://amigaviruses.wikidot.com/disk-herpes
DOS.AIDS:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=245
EGABTR:	http://en.wikipedia.org/wiki/EGABTR
Elk Cloner:	http://www.securityfocus.com/infocus/1286
FEAR:	http://www.viruslist.com/en/viruslist.html?id=768&key=000010000100754&f_page=25
Fumble:	Teknisk beskrivelse i Ludwig, 1995, s. 551.
Green Caterpillar:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=1930 http://www.anti-malware.info/payloads/caterpi.gif
Gullible:	http://www.f-secure.com/v-descs/gullible.shtml
HPS:	http://www.peterszor.com/hps.pdf
Loveletter:	http://en.wikipedia.org/wiki/ILOVEYOU
Magistr.a:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=22574
Marburg:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=20255
Mawanella:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=26380
Melissa:	http://en.wikipedia.org/wiki/Melissa_virus
Melting Screen:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=22619
Muhamor:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=7627
NEW AIDS:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=7866
Nople:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=24803
SilverRat:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=23053
Staple-a:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=26495
Sulfnbk.exe:	http://www.f-secure.com/hoaxes/sulfnbk.shtml
Sulpex.a:	http://www.viruslist.com/en/viruslist.html?id=51380
TechnoRat:	http://www.viruslist.com/en/viruslist.html?id=3168&key=00001000050000400017
Zafi:	http://www.viruslist.com/en/viruses/encyclopedia?virusid=57484