

**Hacking Systems, Hacking Values:
Interactive Theories For An Interactive World**

by
Liam Kelly
Virginia Polytechnic Institute and State University

Submitted as partial fulfillment of the requirements for the degree
Master of Science

In the department of
Science and Technology Studies

Committee
Joseph Pitt, Professor of Philosophy
Richard Hirsh, Professor of History
Deborah Johnson, Professor of Applied Ethics, University of Virginia

Defense Date: December 12, 2003
Blacksburg, VA

Keywords: technology and values, information security, technological momentum,
philosophy of technology

Copyright 2003, Liam Kelly

Hacking Systems, Hacking Values: Interactive Theories For An Interactive World:

Liam Kelly

Abstract

Langdon Winner's article "Do Artifacts Have Politics?" (1986) has become a classic piece within Science and Technology Studies. While Winner was certainly not the first to consider the inherently political qualities of technology, his article has assumed the role of a touchstone for both supporters and critics of the idea that artifacts embody political and social relationships. In the chapters that follow, I shall try to answer Winner and his critics, by studying a particular technology that I believe to be capable of shedding some much-needed light on the issue. My aim is provide a restatement of Winner's question in the pages that follow, with the hope of getting past such problematic terms as "embodiment" and "encapsulation." My hope is to make the issue itself clearer, so that we can get to the heart of how technology, values, and human beings systematically interact.

I shall utilize in my discussion computer network scanning software. I shall first discuss the background to the question "Do Artifacts Have Politics?" and then describe some of the ethical and political forces alive in the computer security world. Next I shall closely examine two particular pieces of network scanning software and describe their interactions in terms of political and ethical motivations. Finally, I shall use this case study as a basis for a broader discussion of how values may be better conceived in terms of complex interactive systems of human beings and technologies.

Acknowledgements

I would like first of all to thank my committee for agreeing to take on what might seem to each of them a peculiar and hopefully adventurous project. In particular, this paper owes much to Dr. Joseph Pitt. He has been an oppressive advisor, a good sport, and a great friend. Additionally, I would like to thank my many colleagues and professors whose ears I have bent over last two years. While I prefer to write in solitude, without the collaborative thinking of which I've so often been the beneficiary that solitude would have nothing to say. Finally, I owe a debt to the many electronic bandits and lawmen who make the Internet such an exhilarating space in which to work and play. As long as there are interesting questions in the world, I am pleasantly confident that we shall always have women and men willing and able to craft interesting solutions.

Table of Contents

ABSTRACT	II
ACKNOWLEDGEMENTS	III
CHAPTER ONE: TECHNOLOGY AND VALUES	1
CHAPTER TWO: HACKING ETHICS	10
DOMINANT MORALITY	11
HACKER ETHICS.....	12
CRACKER ETHICS.....	15
COMPLEX MODELS FOR A COMPLEX WORLD.....	17
CHAPTER THREE: HACKING TECHNOLOGY	20
AN INTRODUCTION TO NETWORK SCANNERS	20
A DOUBLE-EDGED SWORD	23
THE BIRTH OF SATAN	24
SATAN IS DEAD; LONG LIVE SATAN.....	29
CHAPTER FOUR: HACKING VECTORS	39
THE HEADACHE OF COMPLEXITY	50
DO ARTIFACTS HAVE POLITICS?.....	53
BIBLIOGRAPHY	55
CURRICULUM VITAE	59

Table of Figures

FIGURE 1: JOHN DRAPER HARD AT WORK	18
FIGURE 2: THE SAINT™ SCANNER OUTPUT	22
FIGURE 3: NESSUS PREPARES FOR ATTACK	30
FIGURE 4: TROUBLE IN PARADISE. NESSUS HAS FOUND SOME SECURITY HOLES IN THE TARGET HOST. THE CVE NUMBERS IN THE OUTPUT CAN BE USED TO LOCATE CODE TO EXPLOIT THE HOLES.	32
FIGURE 5: NMAP'S "USE DECOYS" OPTION	33
FIGURE 6: TWO DIFFERING SYSTEMS (S1 AND S2) EXERT COMPETING FORCES (V1 AND V2) UPON A THIRD (S3), RESULTING IN A NEW MOMENTUM (V3) THAT IS THE SUM OF THE INPUTS.	43
FIGURE 7: IN AN ENGINEERING DESIGN TEAM, A HUGE NUMBER OF INFLUENCES FORM A COMPLEX NETWORK OF VALUES, ALL OF WHICH MAY IMPART SOME MOMENTUM TO THE FINAL DESIGN OF THE ARTIFACT.	44

Chapter One: Technology and Values

In his 1986 presidential address to the Society for the History of Technology (SHOT), Melvin Kranzberg asserted that "technology is neither good nor bad, nor is it neutral" (Kranzberg 1986). It is easy to make sense of the idea that technology is neither good nor bad – the artifacts themselves do not have intentions, nor do they call themselves into being or act autonomously. That type of moral autonomy is something that we usually reserve for human beings. However, Kranzberg also does not believe that technology can be neutral. This seems to have all of the makings of a fine contradiction. How can technology (or technologies) be neither good nor bad without instead being neutral? What other option is there? And what exactly is at stake in attempting to answer this question?

In "Do Artifacts Have Politics?" Langdon Winner (1986) argues that technologies carry with them the politics of their origins. In one of his main examples, he discusses the fact that Robert Moses specifically engineered the bridges on the Long Island Expressway to be too low for a city bus to pass beneath, so that the inner-city poor who relied on public transit would be unable to access public areas near the wealthier suburbs. Winner uses this story as a clear illustration of why he believes technologies can be inherently political. He goes further still, and states that the politics need not necessarily even be an intentional part of the design process in order to be significant forces within the object itself. Perhaps more often, value judgments creep into technological design without

conscious decision-making on the part of the designer. For instance, architectural design in the early twentieth century failed to make many buildings accessible to wheelchairs. In contrast to the case of the exclusion of the poor from the bridges on Long Island, it is unlikely that many architects intentionally sought to exclude handicapped individuals from public or private spaces. Requirements for handicapped access simply weren't part of the building code or of the design mentality of architects at the time. Nonetheless, their building designs had unquestionably political effects, and ultimately limited the ability of handicapped individuals to access certain places and thus participate in certain segments of society. To Winner, these designs "embody" a set of political values – even though those values were not a conscious part of the design process.

Winner also discusses a second means by which politics are embodied in artifacts. He asserts that particular technologies – such as nuclear reactors and the electric power grid – require that societies maintain certain social and political systems in order to ensure their proper operation. Without a centralized authority, Winner contends that one cannot have nuclear power. In contrast, he argues that decentralized power sources allow for greater decentralization of governance. For him, this serves as further evidence that these technologies are inherently political.

Joseph Pitt, on the other hand, argues in *Thinking About Technology* (2000) that it is not artifacts that have politics, but people. He believes that to attribute politics to the artifact itself is to assign responsibility to the wrong place. Pitt's "Model of Technology" (MT)

prefers to direct discussions of politics and values away from the artifact, and back to the human decision-making process. It is there, argues Pitt, that we find our politics, not in the artifacts themselves. This leads him to the conclusion that the artifact itself is "value-neutral" – the apparent flip side of Kranzberg's coin from Winner.

It is in many ways understandable that the question of the "value-laden-ness" of technology should receive so much attention within STS and the philosophy of technology, for it seems that much hinges upon the question. If technologies were in fact value-neutral (as Pitt claims), then it would seem to follow that a "proper" study of technology and values need not address the material artifact itself, but only the story of its development and deployment. This line of thinking commits us to examining the process, rather than the product, of that development. To be sure, there is much to be gained by such social history, particularly in its ability to guide our current and future technology policies. Additionally, Pitt's method has strength in that it avoids the traps of technological determinism. It makes clear that whatever the current material forms and uses of technologies may be, they didn't necessarily have to be that way, nor do they need to continue to be so in the future. Pitt emphasizes that humans make decisions – either for good or ill – throughout a technology's life cycle. It is here that Pitt finds politics.

However, I am made uncomfortable by the implication that doing social history necessarily leads us to assert that the artifacts themselves are neutral. I am concerned that this takes the material technology too lightly. If technology is in fact value-neutral, then

we ought to be able to do anything we want with it, and we should be able to take any technology and make it ambivalently serve any given political or value system with equal ease. If we can't do that, then we can't really maintain that the artifact is utterly neutral. However, due to the material constraints of artifacts, they are rarely so supple (very often by design.) If I happen to be in a murderous state of mind, a handgun enables me to do things that a scrub brush does not. While the intention to kill is obviously mine (and not the gun's), we can hardly say that the handgun is neutral in its ability to help me realize that intention. Furthermore, if we allow that objects are value-neutral, then there is no reason to include artifacts at all in a study of values or politics, and we can turn the entire enterprise over to the psychologists, sociologists, and political theorists. I, for one, am not willing to do so, and strongly suspect that Pitt would be even less willing than me. We *do* need to be accountable to the object itself, and do need to include it in our social histories. It does not seem to me that acceptance of the "neutrality thesis" provides us with an adequate basis to demand that sort of accountability to materials.

However, accepting that artifacts are "laden with politics" has its own set of problems. If the artifact itself "contains" the politics, as Winner suggests, then it would follow that we need look no further than the material object in order to make value judgments about it. In that world, we could safely ignore any sort of social history of technology, and divine political secrets from the technologies themselves. If a technology "embodies" a value system in any real way, then there would seem to be little need to study seriously the things on which Pitt dwells – namely, the development and usage of the technology. The

term “embodiment” seems to suggest that everything we could possibly need to know about the political dimensions of an artifact is right there in its material substance.

This, too, is not entirely satisfying. Any archaeologist worth her salt will tell you that politics and values are never entirely legible purely within material culture. Reading artifacts is simply not that straightforward. Archaeologists all too often make incorrect judgments about the political qualities of artifacts, and nearly always consult with external sources – such as written and oral records – in order to verify the correctness of their political interpretations from the artifacts themselves. This seems to me to weigh against the thesis that values are encapsulated within the material artifact. The very word “encapsulation” suggests a tidy, closed container, with all of the important information inside. The metaphor of “embodiment” has a similar effect. Neither term seems to stress the importance of context in understanding technologies. Additionally, it concerns me that accepting “embodiment” may lead much too easily to a form of technological determinism. If a technology “contains” the politics of its origin in a closed way, then how can it possibly be diverted for other purposes? How can we hope to get the politics “out” of the artifact, in order to put new politics “in”?

Obviously, I am artificially polarizing these two points of view in such a way that no clear-headed scholar of technology studies would accept either extreme as I have described them. Indeed, both Winner and Pitt would no doubt wail that I have sorely mischaracterized their respective positions, and both would agree that we ought to

consider *both* the material artifact and the social history of its development. In fact, I would strongly suspect that Winner and Pitt actually agree upon a great deal. Both are quite concerned with doing good social history of technology, and both consider “good social history” to be history which takes into account the roles of both the material technology and the social and political interactions that surround it. So what exactly is the disagreement within the philosophy of technology, and why has the seemingly innocent question "Do Artifacts Have Politics?" engendered so much debate?

It is my strong suspicion that the primary disagreement over the answer to the question "Do Artifacts Have Politics?" lies neither in methods nor even in emphasis, but rather in a disagreement over the meaning of the question itself. In short, I believe the argument is more about semantics than substance. What does it really mean to say that a technology is “value-laden” or “has politics”? At stake is the very attitude that we take toward technological development. The answer has the potential to affect our engineering decisions, our policy-making, our educating, and our continued historical and philosophical discourse.

In the pages that follow, I shall make an effort to clarify what it may mean to say that an artifact is "value-laden." In so doing, it is my hope that the answer to Winner's question will make itself obvious, and that I will have demonstrated that the two seemingly conflicting points of view caricatured above will prove to be quite a bit more similar than they portray themselves to be. Ultimately, it is my goal to demonstrate that terms like

“value-laden” and “embodiment” are red herrings that create apparent dissent where in fact there need be none. I shall suggest an alternate means of conceptualizing the problem, and in so doing, attempt to elucidate and vindicate Kranzberg's puzzling statement about technology and values.

However, it seems to me that any consideration of Technology as a monolithic entity is premature and perhaps ultimately impossible. In order to best consider the issues of technology and values, I find it useful to avoid beginning with claims about “Technology Writ Large”, and instead to focus upon one particular technology that seems especially well suited for the discussion at hand. It is my hope that by doing so, we can clarify very specifically the question of what it would mean for that particular technology to be value-laden, and then perhaps attempt to transfer that clarity to the broader discussion. The best technology for the job would be one about which it is difficult to make obvious and reflexive value judgments (i.e., we shall avoid involving nuclear missiles at this point). Rather, we need a technology that can easily be seen from either of the apparently conflicting points of view in our discussion – one for which we could make a clear case either for or against its “neutrality.”

Additionally, we ought to select a technology that has some history and literature surrounding it. Not only does this make research easier, but it also provides a richer and more complete understanding of the artifact itself. If we were to ignore the history and literature associated with the artifact (or select a technology without other source material

available), and instead attempt to “read” values solely from the artifact itself, we would already be selecting in favor of a methodology that assumes that the artifact “contains” everything that we need to know about it values. I propose that we do indeed need to take a close look at the technology, but also at the discussion surrounding it, the statements made by that technology's creator(s), and the uses to which people actually put the technology. In this manner, I hope to avoid reading too many of my own biases into the study of the technology, and to maintain some external touchstones to which we can hold our examination accountable.

I shall therefore utilize in my discussion a technology that I find to be particularly well suited to debates about values – computer network security scanning software. Such software is in many ways an ideal case. Developers and proponents of such software have extolled its virtues while the popular media has lamented its vices. Network administrators and hackers¹ alike can equally use the software for good or ill, for offensive or defensive purposes. Additionally, most network scanning software is well documented and widely discussed on the Internet, providing sources external to the software itself with which to make value assessments. This allows for the possibility of “reading” both the artifact itself and the rhetoric surround it, allowing for some consistency checking and mitigation of bias on my part. Finally, I have chosen to

¹ Throughout this paper, I shall primarily use the term “hacker” to refer to individuals who break into computer systems. This usage would no doubt offend many self-labeled “hackers” who do not engage in illegal activity (which they would call “cracking”), but only in recreational programming and creative problem-solving. The history of the hacking/cracking distinction is itself fascinating; for points of comparison, see GNU/LINUX guru Eric S. Raymond's paper “How To Become A Hacker” (Raymond 2001) and *Slashdot's* interview with the hacker/cracker collective Cult Of The Dead Cow (“Bizarre Answers...” 1999). I shall, however, observe the hacker/cracker distinction in my discussion of value systems, mostly for lack of a better pair of contrasting terms.

examine network scanning software for personal reasons. It is a technology that fascinates me socially, ethically, and technically. It is also a technology with which I am familiar enough to be able to weigh the finer points of its technical aspects along with its usefulness and effectiveness for accomplishing particular goals, both political and non-political. Such technical understanding is bound to contribute to our wider understanding of the technology and its relation to broader social contexts.

However, before embarking upon a technical explication of computer security software and an analysis of its relationships to the question of technology and human values, I believe it is necessary to first make an effort to articulate the human value systems involved. In the chapter that follows, I shall address the issues of computer security software and value systems, in an effort to uncover some of the interests and motivations of the participants on various sides of the computer security community. This should help us to understand why individuals choose to develop such software, why others choose to use it, why still others condemn it, and what (if anything) they all hope to gain from it. Next, I shall look at some of the technical and rhetorical aspects of the software and documentation themselves, with the intention of understanding and mapping their interactions with the described value systems. Finally, I shall use this mapping as a case study to revisit the question “Do Artifacts Have Politics?” with the hope of bringing it to a successful and satisfactory conclusion.

Chapter Two: Hacking Ethics

The political and moral systems at work in the computer security software world are extraordinarily complex, and in a constant state of flux. There are nearly as many manifestos about hacking as there are computer hackers, and scores of journalists, sociologists, and moral philosophers at work deciphering and analyzing those manifestos and the actions consequent from them. By the time their work gets to press, the field is likely to have already changed, as new technologies emerge to fix or exploit flaws in the old ones. The personalities involved frequently change sides themselves – today's hacker is often tomorrow's corporate security consultant. All of this makes generalizations about “Hacker Ethics” difficult to pin down. This, I think, is for the best. The world is a complex and changing place, and we ought to have complex and mutable theories in order to accommodate it.

However, we also need some starting point. I shall here offer the moral “lay of the land” as I see the computer security world today, with the intent of providing not a road map, but merely a point of departure. The following divisions are convenient constructions in order to expose some important differences. By my conclusion, I hope to demonstrate that these divisions are not in fact “zones” of demarcation containing people or tools, but something more like vector forces acting upon (and enacted by) individual agents to varying degrees.

Dominant Morality

When most of us hear the word “values,” we make immediate associations with right and wrong, or good and evil. Both our everyday language and the majority of our moral philosophy use the term “values” in this way. Whether we're speaking of “family values,” instilling a sense of “values” in our students, or the erosion of “traditional values” in our society, we mean roughly the same thing – namely, standards of conduct which determine moral or immoral attitudes and actions. Moral philosophers certainly take the issue to be quite a bit more complex than the layperson, but both operate under the common notion of “values” being closely linked to morality.

The overwhelming majority of the material published on the subject of computers and values has taken the approach of normative ethics. Volumes such as those produced by Parker (1979), Forrester and Morrison (1994), Johnson (1994), Bowyer (2001), Hester (2001), and Spinello (2001) primarily focus on human moral responsibility and its relation to the uses of computers and computer network technology. They tell us what we should and shouldn't do with technology, and provide us with systematic justifications for moral attitudes. Unquestionably, there is need for this type of work, and need to educate our computer scientists, engineers, and end users in the moral foundations and moral consequences of their decisions. However, as Pitt rightly points out in *Thinking About Technology*, moral judgments are just one of many ways of considering “values” in relation to technology, and it would be a mistake to paint all value judgments broadly as moral decisions. In fact, I would suggest that moral considerations are nowhere near the

primary type of value judgment made by most security software developers. There are other, more prominent value systems at work, as we shall see.

Hacker Ethics

Most computer enthusiasts are at least familiar with the notion of “The Hacker Ethic.” The attitude referenced by the term has been around since the first craftsman took pride in the first man-made artifact. Put generally, the “hacker ethic” simply places value on finding novel solutions to interesting problems. It does not imply “ethics” in the sense of normative moral philosophy, but uses the term more generally (e.g., the “Puritan work ethic”) to describe a different sort of value system. While the term “hacker ethics” has been in use since at least the 1960s (Levy 1984), it has most recently been popularized by Pekka Himanen in his book entitled *The Hacker Ethic and the Spirit of the Information Age* (2001). In it, Himanen discusses the so-called hacker ethic and its relation to what Manuel Castells has coined “The Spirit of Informationalism” (Castells 2000).

The value system described by Himanen and Castells is not without some normative force behind it, but it is not a normativity based entirely upon moral judgments. We witness a definite move away from the human decision-making process simpliciter, and toward some culpability to the materials and artifacts themselves. One of the key principles of the “spirit of informationalism” is that “information wants to be free” (Castells 2000). Notice the interesting verbal shift that takes place in the phrase: it does not tell us that “people want information to be free”, but that *information wants* to be

free. This seems to suggest that freedom of information is not simply a matter of human moral judgments, but somehow also a function of the type of thing that information *is*. It derives its normative force not from human ethics, but from accountability to what we might describe as the “metaphysics of information.” In fact, “information wants to be free” sounds downright Aristotelian, if we consider it within a philosophical context. Just as Aristotle contended that fire “wants” to go up and earth “wants” to go down – each seeking its natural place, independent of what we humans would wish for it to do – Castells would have us believe that information “wants” to be free.

This “metaphysics of information” manifests itself in the hacker world as a disregard or disrespect for policies and technologies designed to limit the flow of information. If the hacker subscribes to the attitude that information wants to be free, then she may see it as her right (or even her duty) to liberate it whenever possible from oppressive forces such as copy protection schemes, intellectual property laws, and network firewalls. For computer data is nothing if not information, and access control devices (including legislation) are explicitly designed to limit access to that data. The normative conclusion that the hacker can draw is that despite what traditional morality might have to say on the subject, there may be great value in committing what the legal system would describe as “computer crimes.” Like the animal rights activist who breaks into an experimental laboratory to free the test subjects, the hacker may see herself as liberator of an entity that deserves to enjoy an existence without restriction.

The second important piece of the hacker ethic is the value placed on problem-solving ability. The term “hacker” is frequently applied to any individual who solves some difficult problem – not necessarily computer-related – in an interesting way. Whether I modify my car's engine in order to increase its efficiency or find some legal loophole that allows me to avoid paying income tax, I might be said to have “hacked” one system or another. The key components, as described by Pekka Himanen, are characteristics such as curiosity, persistence, and ingenuity. The hacker ethic (as he understands it) advocates solving problems for their own sake, not because of some potential for personal gain. Just as a mountain climber might scale a peak “because it was there,” a hacker might probe for entrance into a computer network just for the intellectual exercise of seeing if it can be done. Most hackers never derive any sort of personal gain from their activities (and, in fact, many earn jail time because of them.) For the majority of them, personal gain is beside the point. The point is rather to expand one's knowledge and to satisfy curiosity (and often to gain social status among other hackers.)

Here again, the relationship between values and technology is interesting. While traditional moral values ask what we *should* do with technology, the hacker ethic is more interested in what we *can* do with technology, while the question of whether we “ought” to or not is more of an afterthought. An illustration may help to raise this distinction: Suppose someone has broken into a computerized banking system, and stolen a great deal of money. The value judgment the moral philosopher might make would be that the act was morally wrong. This question could probably be resolved without having to ask

questions about the technology utilized in the break-in. Whether the thief committed the crime with a computer or a crowbar would be of minimal relevance to the moral value of the outcome. The hacker, on the other hand, would instinctively make a different sort of value judgment, based on the elegance of the break-in. What tools did the thief use? Did he utilize commonly available equipment and methods to exploit known flaws in the compromised system, or did he craft his own tools to discover some novel means of attack? The answers to these questions will have a profound impact on the value that the hacker ethic would place on the perpetrator and his illegal act.

Note that I am not claiming that any individual is quite autistic enough to make value decisions based entirely upon the “hacker ethic,” nor entirely upon traditional moral distinctions. Indeed, in the pages that follow I intend to demonstrate that entirely the opposite is true – humans always make decisions about technology with a mix of motives and value systems that interact in complex and sometimes contradictory ways. What I am seeking to identify here are not dogmas to which any individual wholly adheres, but merely different attitudes and justifications that influence technological decisions and choices of technologies, usually simultaneously. How those interact within *real* people and *real* technologies remains to be seen.

Cracker Ethics

Finally, I need to address what I choose to call “cracker ethics.” Closely related to the “hacker ethic,” the cracker ethic explicitly denies traditional morality and authority in all

its forms, and seeks to subvert it whenever possible. Crackers regularly deface web pages, author computer viruses, take control of home computers, hold sensitive corporate data hostage, and generally make headaches for thousands of computer owners and systems administrators around the world. Motivations for such activities vary, but quite often it comes down to power and status within the computer underground.

Manifestos expressing the “cracker ethic” are prevalent, but the best-known example is certainly “The Hacker Manifesto,” (Mentor 1986) authored in 1986 by a hacker known as The Mentor shortly after his arrest. Unlike “old-school” hackers like Eric S. Raymond, who explicitly try to disassociate their version of “hacking” from any sort of criminal activity (Raymond 2001), The Mentor is quite willing to embrace his criminality, wearing it as a badge of honor against what he sees as an oppressive authoritarian system. Near the end of his manifesto, Mentor says, “Yes, I am a criminal... My crime is that of outsmarting you, something that you will never forgive me for.”²

The relation of crackers to computer technology is similar to that expressed by the “hacker ethic”, but with a subtle shift. While the hacker ethic expresses a certain accountability to the diffuse network community, the cracker ethic has much more of a “survival of the technically fittest” mentality. To the cracker ethic, technology is something to be appropriated towards one's own ends, without regard for moral

² For a more detailed description of The Mentor’s manifesto and the context from which it came, see Douglas Thomas’ *Hacker Culture* (2002). Thomas treats The Mentor’s manifesto through the eyes of a sociologist. While I find many of his conclusions dubious and overly extrapolated, his work is nonetheless an interesting attempt to understand hackers and crackers as social animals.

consequences. Cracker groups and cracker tools more often than not take names that are explicitly intended to offend traditional moral intuitions (e.g., “The Legion of Doom,” “The Cult of the Dead Cow,” l0pht's “Back Orifice” tool, etc.) Additionally, cracker “wars” are not unknown, as one group or individual attempts to make a name for themselves at the expense of others. The cracker world has little of the optimism of the “hacker ethic” – quite possibly because it so frequently finds itself on the wrong side of the law, and an enemy to “polite” society.

Complex Models for a Complex World

As stated previously, the values systems that I have depicted here are largely of my own construction. They do represent attitudes and actions taken by real people in the real world (and are distinctions used by others both inside and outside of computer security communities), but they are nonetheless representations. The real world is quite a bit messier than any tidy demarcation can possibly express. Even the most predictable of human beings do not fit neatly into one set of moral categories or another; this is especially true of most individuals who participate in computer security. The popular media has made a great deal of the distinction between “white hat” and “black hat” hackers, but the fact is that nearly everyone’s “hat” is some shade of gray. This, too, strikes me as wholly appropriate to our discussion. It forces us away from essentializing people, and forces us to look instead at the things they actually do.

A great many of today's professional computer security consultants are former crackers



Figure 1: John Draper Hard At Work

who have “gone straight.” John Draper (a.k.a. “Captain Crunch”) is a classic example of this (Fig. 1). Draper was a hacker in the classical sense – an early member of Berkeley's Homebrew Computer Club, a savant when it came to telephone networking equipment, and an early influence on Apple Computer founders Steve Jobs and Steve Wozniak (Levy 1984). Draper also eventually

served a prison sentence for using his skills to compromise the phone system and steal telephone calls. He now runs a computer security firm that sells the “Crunchbox”, a network security appliance designed to prevent similar sorts of activity to those for which Draper himself was convicted (“John T. Draper...” 2003). Would we want to say that one value system or another could exclusively characterize either Draper or the technologies that he created? I think not. People are simply more complex than that. We may, however, want to address particular motives for particular actions, the consequences of those actions, and the technologies that enabled or foiled them.

As a result, I would like to argue in the following chapters that the technologies that people create in order to engage with the world are more complex with regard to value

than the literature on the subject has traditionally granted. In order to do that complexity justice, we need to be willing to engage intimately with the artifacts themselves. In the chapter that follows, I shall briefly examine the origins of computer network analysis tools, and shall then focus on two particular tools with the intent of drawing some conclusions as to the roles of value systems in their design, implementation, and usage.

Chapter Three: Hacking Technology

It seems appropriate to me that if we truly wish to understand whether values are implicit in a technology, we would do well to make a serious effort to understand the technology itself. My concern is that if we fail to understand the technical and functional aspects of a technology, we may too easily attribute to politics things that may in fact have their origin in technical and material limitations. It would be flatly false to claim that *all* of the features of any technology can be attributed entirely to social and political factors. Even if we grant nothing else, we must grant that materials themselves are not infinitely pliable, and that material constraints often influence engineering design. Naturally, most other factors – often including our choice of inflexible raw materials – may be influenced by more “human” considerations. But if we fail to understand those very basic design considerations over which the individual engineer has little control (or fail to understand some of the more obscure technical aspects of a technology), we may be tempted to overstate the political influences in the design process. I shall therefore start with a (hopefully painless) technical exposition of network scanning software. Once we have a good grasp of the basic technical functions of the software, I believe we will find ourselves in a better position to judge its “political” merits.

An Introduction to Network Scanners

Put most simply, a network scanner (or “port scanner”) is a software tool designed to collect information about computer networks. Scanners vary broadly in complexity and capability, but all share at least one basic feature: they poll a computer (or network of

computers) in order to discover what services are available on the target machine(s). The basic means by which they do so is common to all network scanners, and understanding it requires some basic understanding of computer networking protocols (which I shall attempt to offer here).

Each computer connected to the Internet communicates with others by means of a protocol called TCP (Transmission Control Protocol.) In order to receive incoming TCP connections, a computer must listen on one or more “ports.” For the purposes of our discussion, a port can be thought of as numbered point of entry – it represents an “opening” in the computer, through which other computers may communicate. Each port has a number assigned to it, which typically indicates the type of service offered on that port. (For example, web servers usually “listen” on port 80, mail servers on port 25, etc.) So if I want to get a piece of e-mail to someone at EXAMPLE.ORG, my computer must open a connection to port 25 of the e-mail server for EXAMPLE.ORG, in order to send e-mail data into that system.³

³ Naturally, the actual process of doing so is considerably more complex than described here, but the simplified model offered here should be adequate for our discussion.

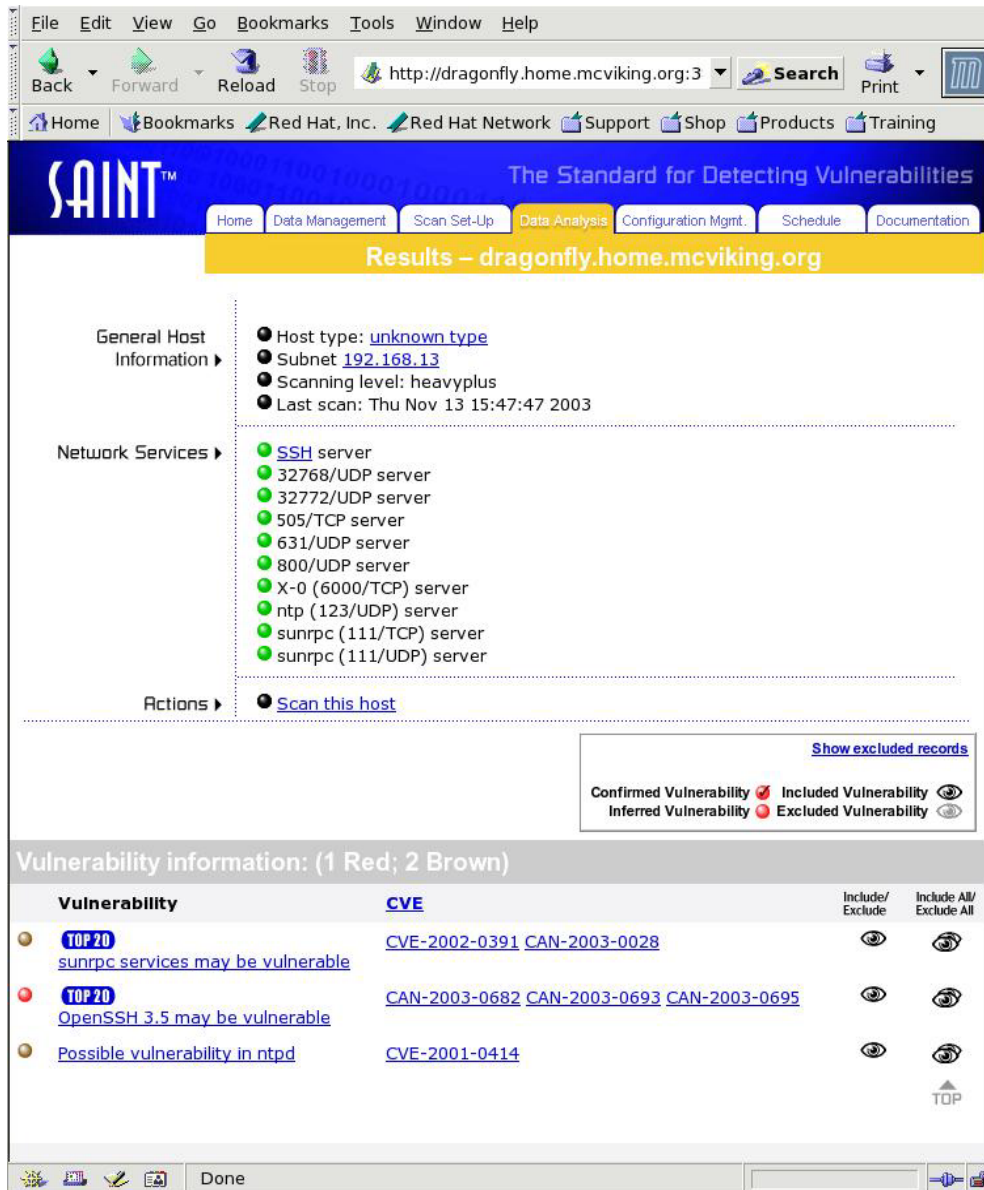


Figure 2: The SAINT™ Scanner Output

Network scanners utilize the open ports on the target system in order to gain information about that system. Rather than connecting to a particular port on a particular machine, a network scanner examines a wide range of ports on one or more computers, in order to discover which ones are open and listening (and thus possible avenues of attack). So if I

were to scan MAIL.EXAMPLE.ORG, I might discover that it not only listens on port 25 (indicating an e-mail server), but also has ports 21 and 23 open, indicating that it is probably running FTP and TELNET services, as well. Many scanners do more than simply list the available ports, and additionally provide more detailed information about the services running on those ports. For instance, scanners such as SAINT™ and Nessus may tell me the operating system of the target system, the versions of the services running on that machine, information about default passwords, network share names, “back doors” left by previous hackers etc.

A Double-Edged Sword

The capabilities provided by network scanners are of use to network administrators and hackers alike. For the network administrator, a scanner allows for a quick and easy audit of the network, in order to expose possible vulnerabilities that can then presumably be blocked or patched. However, that same scanning software can provide the same information to a potential hacker, which she can then presumably exploit. This seems to raise some interesting questions about the relationship between the technology and value systems. If looked at from the point of view of design, a port scanner developed by a hacker for the express purpose of breaking into remote systems would seem to be positively infested with politics: the technology was designed by an individual with a very specific (and perhaps insidious) purpose in mind. However, if we look instead at the tool's application by network administrators and hackers alike – as a means both for securing and penetrating networks – such a strong conclusion seems less justified. In that

case, it would seem that we could make a strong case for neutrality: clearly, the sword can be made to cut both ways. Can we claim that one reading is more appropriate than the other, or that one usage of the technology is “truer” to its purpose, whatever that may mean? A bit of history and a bit of technical exposition may help us along toward an answer to these questions.

The Birth of SATAN

It is impossible to say when the first network scanner was created, or by whom. This is in part due to the lack of record keeping by early hacking software authors, but it is primarily because of the difficulty in determining what should count as a network scanner. The capabilities offered by a comprehensive scanning tool are essentially no different than what any knowledgeable computer user could accomplish with a web browser, a TELNET client, and a small amount of computer scripting know-how. Prior to the release of the first “all-in-one” scanning tool, computer hackers and security administrators simply wrote their own scripts to gather the same information. As such, there is no clear single point of origin for automated scanning, nor is there any single agenda from which it was born. Both the “good guys” and the “bad guys” wrote their own tools, and whoever had the best people working on the smartest tools had the upper hand in the computer security wars.

But in 1993, a new breed of program was introduced to the Internet with the release of programs such as ISS (the Internet Security Scanner) and COPS (the Computer Oracle

and Password System.) ISS and COPS were basically collections of UNIX shell scripts that scanned for a well-known set of common security flaws in UNIX computer systems (“CERT Advisory...” 1997, “COPS README 1998”.) What distinguished the two was that COPS was primarily a local auditing tool that had to be run on the system being audited, while ISS was a *remote* auditing tool, meaning that one didn't need direct local access to a system in order to scan it for vulnerabilities – it could be done across the Internet. While ISS didn't invent or discover any security vulnerabilities that were previously unknown, and while remote Internet exploits of systems were certainly nothing new, ISS did make such remote scans much more convenient than they had ever been before. Users no longer had to collect or author their own scripts to perform the task; ISS had already done the work for them, and they only needed to run the program. While doing so was trivial to users already familiar with UNIX commands and shell scripts, the audience and appeal of ISS were still mostly limited to competent system administrators and hackers. As such, it went largely unnoticed by the media and the wider computing public.

Such were the rationale and setting for the public release of SATAN – the Security Analysis Tool for Auditing Networks. Authored by Dan Farmer, the creator of COPS, SATAN first hit USENET in 1995, and offered the general computing public a comprehensive, “user-friendly” tool to accomplish more simply and more comprehensively what ISS and all of the “home-brewed” scanning scripts already did. SATAN was a point-and-click graphical program that allowed a user to connect to a

network, to enumerate the services running on that network, and to gather information about a built-in list of common security vulnerabilities. It would then provide the output in an easy-to-read graphical format via a web browser. It was also available free of charge to anyone who cared to download it. In the release notes, the authors of SATAN were seemingly explicit about their intended audience: “SATAN is a tool to help systems administrators. It recognizes several common networking-related security problems, and reports the problems without actually exploiting them” (Farmer 1995.) Morally, the stated intention seems to fit well with our first model of values introduced in chapter two (“dominant morality”). SATAN was a tool to “help systems administrators.” Put in terms of values, it was billed as a program intended to do some moral good – to assist the “good guys” in keeping the “bad guys” out.

Unlike ISS, however, SATAN caused something of a stir in the mass media, who expressed grave concerns as to the actual intention and the inevitable result of SATAN. In the *LA Times* the chairman for FIRST (the Forum of Incident Response and Security Teams) was quoted as saying, “SATAN is like a gun, and this is like handing a gun to a 12 year old” (Harmon 1995.) While there is obviously a bit of journalistic hyperbole at work here, the value judgment is clear enough: when the media looked at SATAN, they saw a tool containing tremendous *negative* moral value. It was not a tool to help systems administrators; on the contrary, it was a tool that would bring enormous *harm* to systems administrators. And when the program's authors deliberately chose a name like SATAN, who could really fault the media's negative reaction? Naming one's program after the

“Prince of Darkness” is not exactly a positive first step toward establishing its moral credibility⁴. Neither were many of the comments Farmer made to the press. In the aforementioned *LA Times* article he said, “If we do this right, a great number of systems will get hammered by this thing. That's why we're writing it. System administrators will be racing to fix their systems because Satan is going to be out there and nothing can stop it.”⁵ Farmer's idea was to create a crisis of proportions significant enough that system administrators would be forced to take the security of their networks seriously. He argued that this would ultimately create a “safer” Internet.

If we look more closely at SATAN's documentation, it becomes quite clear that something other than (or in addition to) “traditional moral values” is at work. On a page on SATAN's web site entitled “Philosophical Musings,” the following statement appears:

Why did we create SATAN? Quite simply, we wanted to know more about network security, particularly with respect to large networks. There is an enormous amount of information out there, and it is definitely not clear by examining information and hosts by hand what the *real* overall security picture is. SATAN was an attempt to break new ground, to promote understanding, and to have fun writing such a program.

Elsewhere on the same page:

History has shown that attempts to limit distribution of most security information and tools has [*sic*] only made things worse. The "undesirable" elements of the computer world will obtain them no matter what you do, and people that have legitimate needs for the information are denied it because of the inherently arbitrary and unfair limitations that are set up when restricting access.

⁴ Interestingly, a few years after the release of SATAN, a separate computer security group released a new scanner based upon SATAN's code base. Its name: SAINT™ -- the Security Administrator's Integrated Network Tool (“Saint Introduction”).

⁵As an interesting side note, Farmer's vigilante attitude toward computer security and his comments to the press resulted in him being fired from his security administrator's position at Silicon Graphics shortly after SATAN's release (Bank 1995.)

And finally, in response to the question of why SATAN is designed to allow remote scans:

All the hosts scanned with SATAN are done so because it gives a clearer picture of what the network security of your site is, by examining the webs of trust and the possible avenues of approach or attack. Since there is no way that SATAN could, a priori, know where it is going to scan, we decided that instead of placing artificial constraints on the program, we would allow the system administrator to place their own constraints on where SATAN would run, via the configuration file.

It seems clear that values other than traditional moral judgments have entered the discussion at this point, and that their introduction has significant influence on the basis for the justification for SATAN's existence. SATAN was not simply a tool to help network administrators, but also an attempt to “break new ground” and simply to “have fun.” In short, it could be described as an attempt to solve an interesting problem in a novel way – an instantiation of the hacker ethic described in chapter two. This seems well confirmed by the statement of concern raised by the program's author as to what would happen if access to SATAN were “unfairly” and “arbitrarily” limited.

Also important to note is the justification as to why SATAN is allowed to scan network hosts other than one's own. Here, the appeal is neither to morality, nor to the “hacker ethic” of freedom of information. The explanation this time is purely technical: since SATAN has no idea in advance where it will need to scan, the authors have no way in advance to limit its scanning to particular networks. Because of the ways in which Internet Protocol works, and because SATAN needs to be portable to a wide variety of networks with a wide variety of architectures, there are certain limitations that simply

can't be placed on it without breaking its basic functionality. Those “material” limitations exist independent of the politics or moral leanings of the program's authors, and are instead bound up in the type of thing that SATAN *is*, and in the environments in which it must function.

The question that we might be tempted to ask at this point is “Which is the *real* SATAN? The SATAN with positive moral value that exists as an aid to network administrators? The SATAN with negative moral value that exists as a threat to network administrators? The SATAN that exists as an embodiment of the “hacker ethic”? Or the SATAN that is what it is because of material necessity, independent of politics and values?” I hope that by now, my answer is predictable: it is all and none of these. The technology and its life cycle are far too complex to be reduced to any single value system, or to be simply written off as “neutral.” The tale of the technology and its relation to values is quite a bit more interesting and elaborate than that.

SATAN Is Dead; Long Live SATAN

The SATAN software is no longer actively maintained, and is now years out of date, and no longer a serious threat nor a significant aid to network security. But dozens of other network scanners (both commercial and freeware) have taken its place. SAINT™, Nessus, Nmap, LANGuard, SARA, and many others now offer far more capability than SATAN ever did (“Quality Security Tools” 2003). In fact, many of them are founded upon SATAN's source code, and add improvements and capabilities to that code. This

type of “code-sharing” is common among open-source security software. Because the source code is available along with the software itself, competent network programmers can easily add (or remove) features of one piece of software to make it their own. This process of addition and subtraction is itself quite relevant to our discussion of values (particularly with regard to the discussion of technological momentum offered in chapter four).

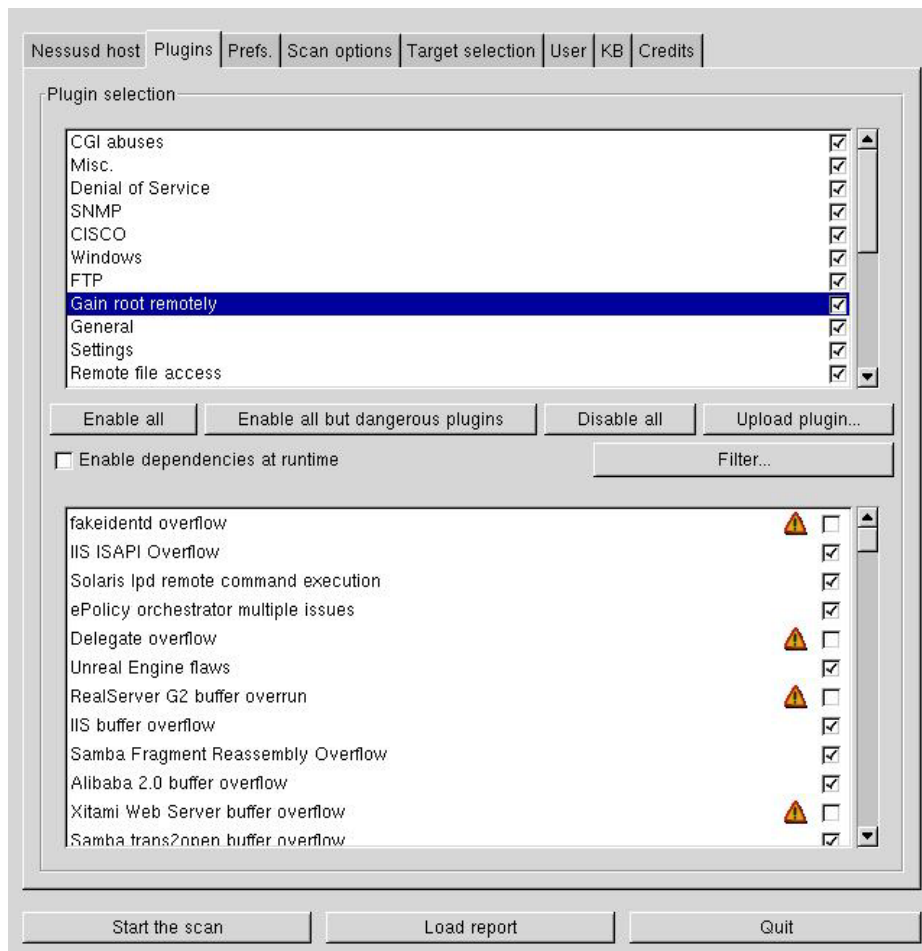


Figure 3: Nessus Prepares For Attack

Let us take, for example, the Nessus scanning engine. Nessus is one of the most popular

and powerful network scanners available today. It is also (arguably) one of the easiest to use. Nessus provides a graphical user interface (GUI) that offers the user a stunningly complete set of scanning options. It goes well beyond simply displaying open ports on the target, also displaying the specific versions of the services running on those ports, a list of common exploits to which those services may be vulnerable, a list of other hosts or networks considered “trusted” by the target host, and a series of graphical charts showing the complete list of possible vulnerabilities for a network, ranked by order of severity. The exploit output of Nessus also includes links to the Bugtraq⁶ web page disclosing details of the vulnerability, from which a user can then usually download programs to exploit it.

⁶ The BUGTRAQ mailing list, run by Securityfocus.org (“Security Focus...” 2003), is one of the most active computer security mailing lists on the Internet. The list centers upon public disclosure of software and operating system security flaws, often including “proof of concept” exploit code in order to demonstrate in detail how the vulnerability works.

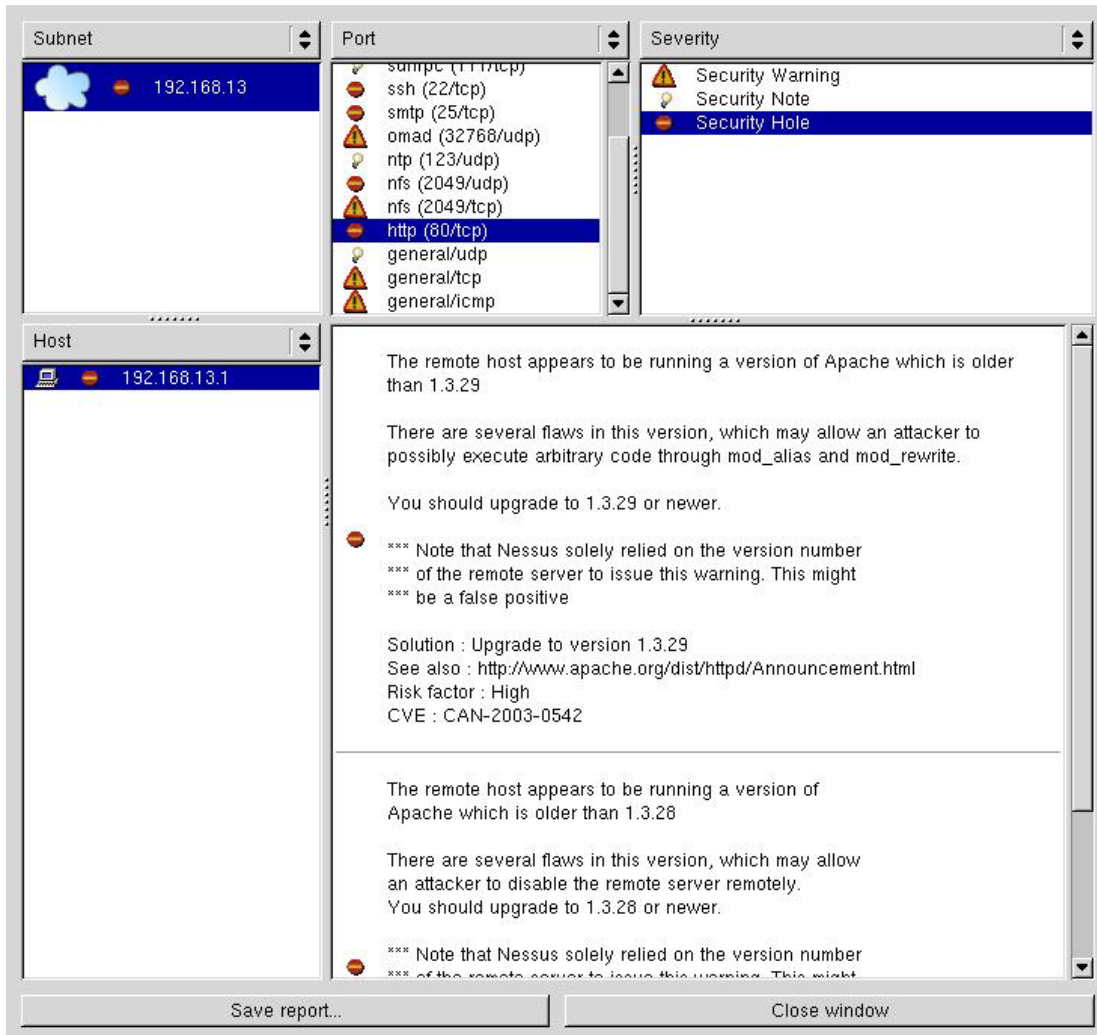


Figure 4: *Trouble In Paradise.* Nessus has found some security holes in the target host. The CVE numbers in the output can be used to locate code to exploit the holes.

But perhaps more interesting are the features that Nessus *doesn't* include. Nessus (an open-source program) uses Nmap (also open-source) for its scanning functions. While Nmap generally has far fewer features than Nessus, it does include at least one feature that Nessus does not: the ability to mask the identity of the computer performing the scan. Through the use of Nmap's "Use Decoys" option,

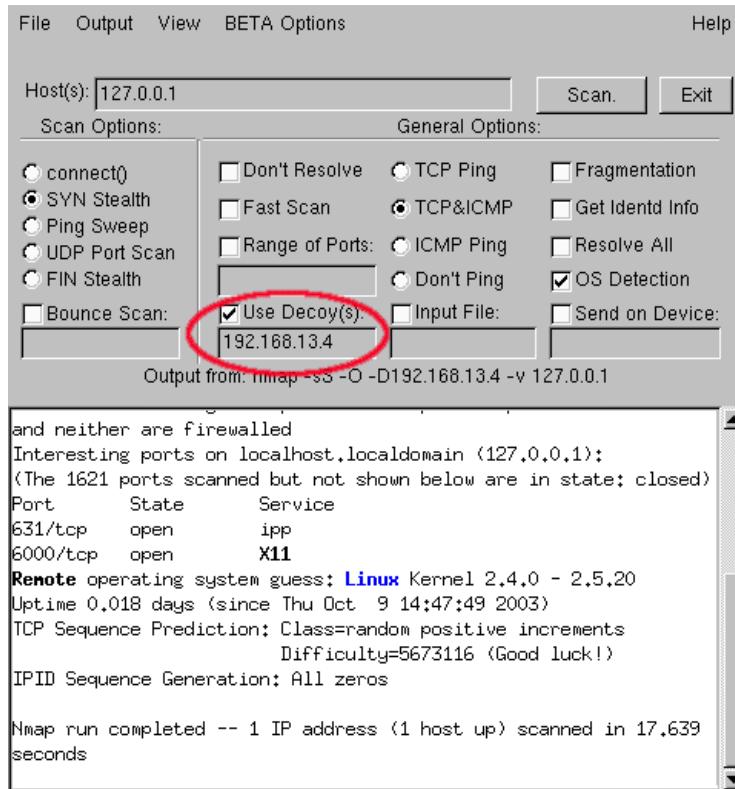


Figure 5: Nmap's "Use Decoys" Option

it is possible to set a number of decoy hosts, such that a systems administrator would be unable to tell whether it was the hacker's machine or one of the decoys conducting a scan against her system.

The "use decoys" feature of Nmap could have been included in the Nessus scanner, as well. (Although for technical reasons, it could only have been used for certain limited types of scans.) In fact, since Nessus actually uses Nmap for its scanning functions⁷, the

⁷ More recent versions of Nessus in fact move away from reliance on Nmap for the underlying technical architecture. While current versions of Nessus still allow the option to use Nmap to perform scans,

author of Nessus would have deliberately had to *omit* the decoy feature from his engine. This seems to suggest some difference in values between the designs of the two programs. Both Nessus and Nmap can be described to fit with what we've said about the “hacker ethic.” Both are interesting solutions to an interesting problem, and both seem to be built on the notion that access to information should be relatively unrestricted. However, on the decoy issue they seem to diverge. The ability to mask one's identity serves no *positive* purpose in terms of gathering information. If I were to scan my own network for flaws, there is no imaginable reason that I would want to hide my point of origin. Only if I am performing an unauthorized scan on someone else's system does the decoy feature become useful. The author of Nmap seems to condone such activity (consistent with the “cracker ethic” described in chapter two) by implication of choosing to include the feature in his software. The authors of Nessus, on the other hand, seem implicitly to condemn such activity by deliberately *removing* the option from their interface. Presumably, this was a decision made at least partly on moral grounds – i.e., influence exerted by “traditional morality” rather than the “cracker ethic.” (As we shall see shortly, this presumption seems to be supported by the language used in the documentation of both programs, as well as personal correspondence with the author of Nessus.) The case of decoy scanning is but one of several features of Nmap that did not make their way into Nessus. Nmap also includes things such as “idle scanning” (which “bounces” the scan off of some innocent “zombie” host en route to the target) and “bounce scanning” (which does a similar bouncing via File Transfer Protocol.) Each of these features contributes to the anonymity of the computer performing the scan – an

Nessus now contains its own separate scanning libraries, as an alternative to those provided by Nmap.

anonymity for which a “legitimate” user of the software would find little use.⁸

I should however mention that while Nessus does not allow for the use of “decoys,” it does include options for IDS (Intrusion Detection System) evasion. An IDS is a piece of software that runs on a computer server and warns the system administrator that a scan or attack is in progress against the system. The IDS evasion features of Nessus allow the user to modify the port scan such that an Intrusion Detection System will have a more difficult time identifying that a potential attacker is probing the system. In the documentation for the IDS evasion features, the author of Nessus admits some misgivings:

Adding such features in Nessus is something we've been hesitating to do. On the one hand, we did not want to see it turn into a script kiddies⁹ tool, on the other hand, we did not want to see it used as a NIDS stressing tool as it was. As it turns out a good NIDS will have even more red blinking alerts when these features are enabled, we felt confident in adding them and releasing them to the public (Nessus 2003.)

The author of Nessus justifies the IDS evasion features as a means of “stressing” or testing one’s detection system in order to find out if it is capable of correctly identifying

⁸ In personal correspondence with Renaud Deraison, the author of Nessus, I asked about his decision not to include Nmap’s “decoy scanning” in Nessus. Deraison responded that the decision was “both ethical and commonsense.” He then proceeded to explain the technical reason that it wouldn’t fit well with the architecture of Nessus. Whether these were meant to be two different answers or whether Deraison has a definition of “commonsense” with which only a computer scientist could empathize, I do not know. However, he did make it quite clear that both ethical and technical considerations played roles in the decision.

⁹ “Script kiddy” is a derogatory term usually applied to a breed of adolescent hackers who use other people’s scripts or programs to break into (and deface) web sites. People who perceive themselves as “true” hackers particularly malign the fact that most script kiddies have only a dim idea of how the tools that they employ actually work.

attacks. Consistent with the wider goals of the Nessus scanner, his expressed hope is that the inclusion of the IDS features will ultimately result in better intrusion detection and fewer break-ins in the long term.

The contrast between the documentation of both Nessus and Nmap seems to support the thesis that there is some difference in moral agendas between the two programs. Nmap's manual describes it as a program “designed to allow system administrators and *curious individuals* to scan large networks” (italics added) (“Nmap Network...” 2003.) The manual speaks of how to “own” systems (hacker jargon for obtaining system-level access on a computer) successfully and anonymously, and even includes a whimsical option to log the program's output in “s|<ipT kiDd|3 f0rM” (“script kiddy form” – the typewriter symbol pidgin used by many would-be crackers in Internet chat rooms and web site defacements.) The language used in the documentation for Nessus, on the other hand, is much more closely allied to the stated goals of SATAN as a tool for computer security administrators. The FAQ (frequently asked questions) page for Nessus even describes it as a replacement for the badly outdated SATAN and an alternative to the expensive commercial products on the market (“Nessus FAQ” 2003). While there is obviously considerable overlap in both the audience and technology of Nmap and Nessus, there also seem to be some clear differences between them.

As we look more closely at the technologies in question, it should become increasingly apparent to us that the alleged distinction between technologies being value-laden vs.

neutral doesn't begin to either address or explain the technology itself. Even in our extremely simplified model including only three value systems and two pieces of very similar computer technology, the relationships between technology and value systems get very complex very rapidly. Nessus and Nmap both seem to be influenced by the “hacker ethic,” but seem to be pulled in different directions by traditional morality vs. the “cracker ethic.” Yet despite the fact that the tools themselves may have been written with specific uses in mind, they are by no means deterministically bound to those uses. While it may (arguably) have been designed at least partly for crackers, Nmap is still an extremely useful informational tool for a system administrator trying to keep crackers out. Likewise, even if we accept the characterization of Nessus as a network analysis tool for system administrators, it could also be a very powerful information-collection tool for hackers and script kiddies. Nonetheless, Nessus clearly places some limitations on the flexibility that it allows its users with regard to access to Nmap's underlying capabilities, suggesting that the tools themselves are clearly not neutral as to the types of activities in which they encourage their users to engage. Put simply, Nmap facilitates activity that Nessus impedes. So: are the programs “value-laden” or not? The interactions between values (or “politics”) and technology now seem muddled at best. Or, if we prefer to state the situation more optimistically, we are beginning to discern some of the complexity that the value-laden/value-neutral discussion sadly obscured from view.

This brings us back around to our original question, and begins to expose some if its

inadequacies. We need an explanation for our software case study that is rich enough and capable enough to be able to do justice to the data. Neither Winner's "value-laden" artifacts nor Pitt's "value-neutral" artifacts seems up to the task. Here we hear echoes of Melvin Kranzberg's First Law, reflected back at us in the form of a question: if technology (or a particular network scanner) is neither good, bad, nor neutral, than what *is* it? Or, put more precisely (and unfortunately more verbosely): if an artifact is not reducible to "containing" a particular set of values, but also can't accurately be described as neutral with regard to how it influences human behavior, then how can we best describe the interactions between human values and material artifacts? It is this question to which I hope to provide a satisfactory answer in chapter four.

Chapter Four: Hacking Vectors

In “The Evolution of Large Scale Technological Systems” (Bijker 1987), Thomas Hughes discusses a tool that I believe has considerable utility in our current discussion. In addressing the question of technological determinism and whether or not large-scale technological systems ever gain autonomy, Hughes argues that what results from sustained large-scale technological development is not deterministic autonomy, but something more like momentum. That is, as a technological system grows in size, its component technologies become increasingly interdependent and its human participants become increasingly entrenched in habit and subject to the material limitations of the complex systems in which they live, work, and play. Because of the increased size and complexity of a mature technological system, making significant changes to either the purpose or composition of that system becomes quite difficult. Thus, Hughes describes the system as having obtained a large “technological momentum”: early on in a technology’s life cycle, it is more easily altered than it is once those dependencies and habits become more developed.

It seems to me that momentum is not only an excellent metaphor to help us solve the problem of technological determinism, but also a tremendously powerful tool for understanding the complex interactions between technologies and values. The metaphor allows us to think about technologies and values within the context of *systems* (rather than in isolation). It allows us to account for the dynamic and changing character of the

procession from a technology's design to its deployment, and it allows us to embrace complexity while avoiding the problematic reification of values as being “in” objects.

Thinking about technologies and values as components of larger systems is something that we ought to take quite seriously. What do we *mean* when we say that a technology “has politics”? What I think we mean is that because of the way that a technology has been designed, it enables or limits certain behaviors or power relationships. A technology can only do so through *interaction* with people and the world. An artifact (or a human being, for that matter) suspended in a vacuum, affected and being affected by nobody, cannot engage in politics or values. In fact, it does not seem remiss to understand politics or values to be rules or standards that govern interactions (between human beings, between humans and the material world, and perhaps even between components of the material world itself.) As such, we ought to look for politics or values *in those interactions*, not in either the people or the objects themselves.

This approach allows us to capitalize on the best parts of both sides of the “do artifacts have politics?” debate, while avoiding the shortcomings of each. It acknowledges the role of the social in the development of technologies (humans making choices about material technology), and forces us to look beyond the material objects themselves. It also acknowledges the inverse (material technology influencing human behavior), and forces us to look at more than just human decision-making, and to take technical/material limitations seriously. Additionally, this approach serves Hughes' purpose of explaining

the difficulty in changing established technological systems without having to resort to simplistic deterministic arguments.¹⁰

It is not enough merely to assert that systems have momentum; we ought also to attempt to understand the dynamics of that momentum. Here I shall treat the metaphor of momentum even more literally than Hughes himself does. As any high school physics student knows, momentum has two components: mass and velocity in some vector (direction). “Mass” we can take to be analogous to the “size” of the system. That is, the more constituent parts a technological system has (and the more complex the interactions between those parts), the more “massive” the system is. For instance, the electrical power grid in the United States is a more massive system than a crowbar. “Velocity in some vector” is a bit more complicated, but I think that here the metaphor holds particularly well. Technological development is usually not aimless but has some “direction” associated with it, and some rate of progress toward that direction. For instance, a technology is usually developed toward some particular goal or set of goals: in the case of Nmap, one of those goals might be to facilitate computer break-ins. Momentum can then be understood as the combination of mass and vector velocity: a

¹⁰ There may still be some deeper problems of technological determinism as it relates to human free will. Momentum, at least as understood by physicists on the macro scale, is an entirely deterministic force. While a sufficiently complex system with a sufficiently large number of interactions may be better treated through stochastic modeling and statistical mechanics than through traditional Newtonian mechanics, there may still be issues as to whether individual human beings truly choose which value systems they endorse and employ if we take the metaphor of momentum literally as I have suggested here. As addressing that particular question would require addressing the entire philosophical literature on human free will and determinism, it is my hope that such issues may be safely ignored for the moment, without substantively detracting from my overall purpose. For the sake of understanding large-scale systems, I think it is adequate to view only the interactions in which people engage, and to “black box” the free will question as it pertains to human decision-making. This may have implications as to the types of normative claims that we can make in such systems, but as I intend to offer only a descriptive model here, I am willing to leave moral normativity aside for the time being.

massively complex system moving quickly toward a clear set of goals is more difficult to divert than a less massive system with no clear direction. This is true not just for material systems, but for humans and organizations of humans as well. (For example, it would be much easier to change my mother's mind on some political issue than to sway the platform of the Republican Party.)

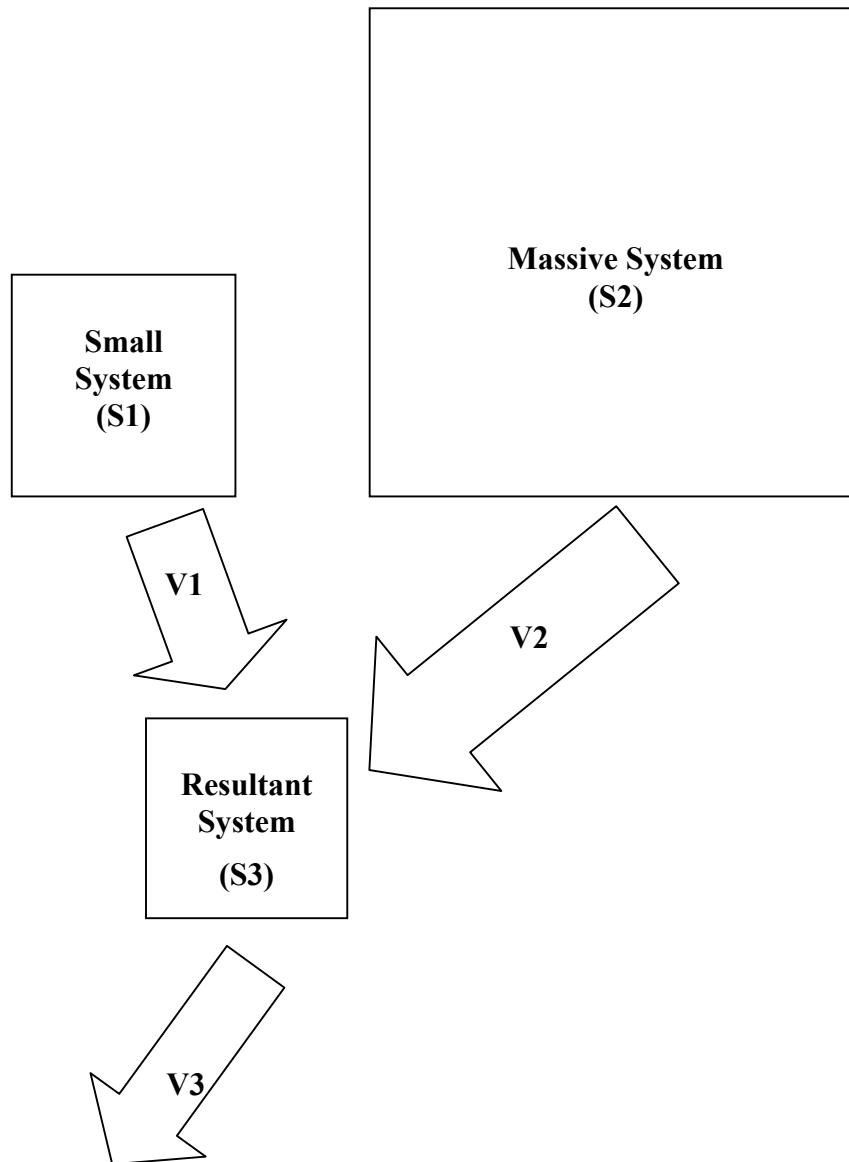


Figure 6: Two differing systems (S1 and S2) exert competing forces (V1 and V2) upon a third (S3), resulting in a new momentum (V3) that is the sum of the inputs.

Given this framework, we can understand values and politics in much richer ways than simply being intrinsic or extrinsic to technologies. We are now prepared to talk about

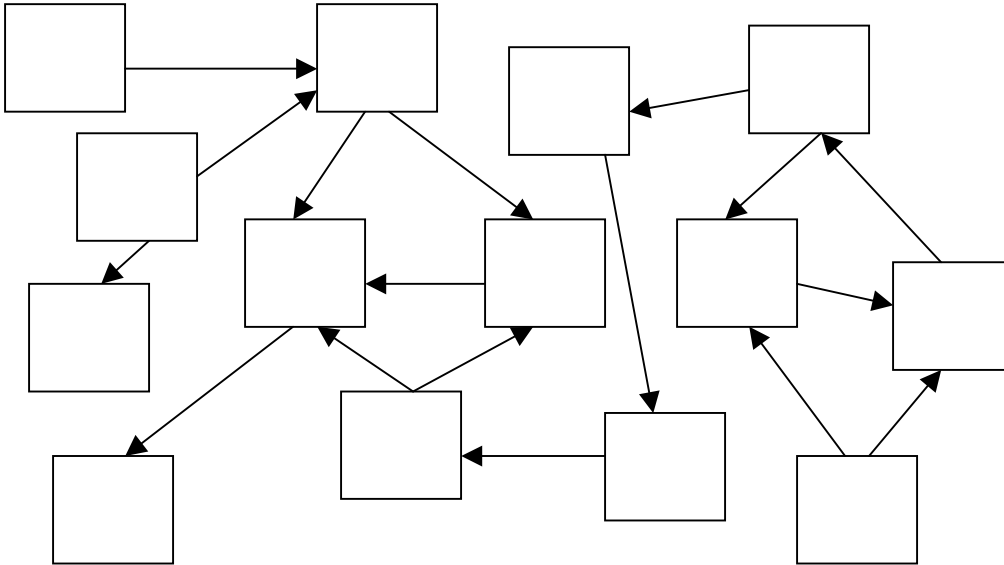


Figure 7: *In an engineering design team, a huge number of influences form a complex network of values, all of which may impart some momentum to the final design of the artifact.*

interactions – specifically, we can view politics as a force that is exerted by and upon a technology (or technological system) within the context of some larger system. When an engineer, acting as part of a design team, exerts her influence on the design of some technology, she imparts a certain momentum to that technology, and in so doing enacts some set of values or politics. The other members of her design team do likewise, and by the time the technology is produced and released to the public, it has had potentially hundreds of “value-forces” exerted upon it, each in slightly different directions and with varying degrees of force. The technology, as it enters the world, has some momentum already, the direction of which is the resultant vector from all of the various input forces.

Per Hughes' model, this resultant momentum is not something to which the technology is autonomously bound for all of time. On the contrary, after the artifact is sent forth into the world, the users of that artifact may appropriate it towards slightly different ends –

slightly different directions – than that which its designers had foreseen. The ease with which the user may exert their own values toward re-directing that artifact will be a product of the momentum of the artifact, the difference in direction between the original vector and the user's new desired direction, and the force which the user is able and willing to exert.

To return to our software example and thus hopefully to make the point more clear, suppose we assume that the author of Nmap designed the software primarily as a tool to assist crackers. We could probably chart the initial momentum of Nmap as a combination of forces enacted primarily through the “hacker ethic” and the “cracker ethic” described in chapter two. The resultant vector lies somewhere between the two with its specific direction and magnitude determined by the balance of momentum behind the original values. It is part hacker tool, part cracker tool. Nmap then takes on this resultant momentum, and as its capabilities “collide” with or become co-opted into other individuals and technologies, its momentum will influence theirs. So when Nmap falls into the hands of a fledgling cracker with some inclination towards mischief but little capability to realize that inclination, the tool’s momentum combines with the cracker’s and has an amplifying effect – effectively extending the human capability and simultaneously reinforcing a set of attitudes and behaviors. Similarly, if a curious systems administrator uses Nmap to explore the security of her own network, the “hacker” momentum imparted to Nmap provides facilitation for that activity. It also, however, provides a tempting set of options for other sorts of activity, in part because of

the “cloaking” features of the software. It provides at least a nudge in the direction of the “cracker ethic.” Whether or not our fictional systems administrator acts upon that nudge probably depends largely on the momentum with which “dominant morality” nudges back, acting as a balancing force.

If we assume that the Nessus scanner (in contrast to Nmap) is a tool primarily influenced by “dominant morality” and the “hacker ethic,” then when the developers of Nessus chose to use Nmap as their underlying scanning engine, they were partially able to divert it away from its original vector toward a slightly different purpose: i.e., to divert it away from facilitating the “cracker ethic.” Because Nmap was already “moving” in a direction at least partially compatible with the goals of the Nessus team, diverting it towards slightly different goals requires less force and effort than if Nmap had had a completely different suite of purposes in the first place. It required little more than the omission of certain buttons and switches from the interface, and not a radical redesign in the underlying software architecture. The change in momentum was more like a sideways nudge than a complete reversal of direction. Nessus, in turn, nudges its users toward a different set of behaviors than does Nmap.

I have spoken so far of how humans influence technological momentum through exertion of values, but it is also true that the momentum possessed by technologies affect “human momentum” as well. Once we create a tool that facilitates some behavior, we make it much easier for someone to enact certain values. E.g., once hacking tools exist, hacking

becomes easier. The (possibly small) momentum of an aspiring hacker, when combined with the (possibly large) momentum of a mature technological system (such as a hacking tool set), may produce a much more forceful resultant with much more dangerous consequences than that the individual could have enacted on his own. Likewise, the interactions between humans may be mutually reinforcing: when like-minded individuals with common goals collaborate, they not only reinforce each other's beliefs, but also have a tendency to facilitate behaviors mutually. We can see exactly that take place when hacker or crackers join together into collectives like The Legion of Doom or the Cult of the Dead Cow. This “amplification of values” is probably one of the very most basic principles behind the formation of human communities. On the other hand, when directly opposing value systems attempt to act on the same technological system, the outcome is likely to be of minor results in proportion to the energy expended. (We need look no further than political deadlock on technological policy-making for examples of this.)

We are now prepared to return to the original examples of power plants, nuclear weapons, and the bridges on the Long Island Expressway. The metaphor of interactive momentum allows for a dynamism and a vivacity that the “value-laden” metaphor does not. What forces motivated Robert Moses in his municipal designs? Certainly the racism and classism that Winner cites were active influences. However, we can be equally certain that they were not the *only* influences. Moses, as any other human being, did not exist in an ideological and material vacuum. He was at least as equally bound to state and city ordinances and fundamental principles of civil and material engineering as to his

own ideological goals. Those ordinances and material constraints placed limitations on the type of bridge that he was capable of designing. Within those limitations, Moses still found enough flexibility to express his ideological goals. On at least that much, the advocates of both “value-laden-ness” and “value-neutrality” will agree.

But here, the question of whether the bridge “contains” Moses’ racism can go no further. Is the racism “in” the bridge or not? What we end up with is an endless semantic dispute as to what we can possibly mean by “in.” But if we discard the question and employ Hughes’ metaphor of momentum instead, we get a much more interesting and profitable discussion. We’ve identified at least three forces that had input into the bridge design – Moses’ ideological goals, New York construction ordinances, and the material constraints of bridge design. Naturally, we could add hundreds of other factors to this list – public expectation, funding constraints, the review process for the project proposal, etc. Next we can start balancing those forces in terms of their influence. Did Moses compromise accepted standards of engineering in order to achieve his ideological ends? Was public pressure inadequate to alter the outcome of the review process? The answers to questions such as these will help us to form a clearer picture of the complex interactive systems of value judgments that went into the design of those bridges by forcing us to be accountable to both the material artifact and the social forces involved.

This approach also allows us to look at the *effects* of the material artifact, and of the forces that it exerts upon future technologies and decisions. Rather than asking what

values (if any) are contained in a bridge, we can ask questions that I think are closer to the heart of the matter and the spirit of our original question. What behaviors does the bridge facilitate? What activities does it impair? How did the material structure of the bridge affect the material forms of other technologies with which it interacts? What social, moral, and material consequences – both intended and unintended – resulted? Neither encapsulation nor neutrality is adequate for providing answers to these questions. The bridge may have been created in part to serve some set of ideological goals, and it may well have achieved those goals in the short term. But what happens when we simply design shorter buses, rendering the ideological goal ineffective? Would Winner be forced to say that the politics have somehow “gone out” of the bridge? The question barely makes sense. However, we can talk quite easily and naturally about a change in the way that the world and the bridge systematically interact in terms of diversion or diffusion of technological momentum.

One of the primary advantages to the metaphor that I have described is that it allows us to treat a technology as something dynamic and changing, in a way that calling an object “value-laden” does not. It allows us to talk sensibly about how a hammer can be a tool one day and a weapon the next. Also importantly, it does so in a way that doesn't neglect the material constraints of the artifact the way that the “neutrality thesis” does. Less complex technologies with fewer material constraints and dependencies attach to less “massive” systems, and are more easily diverted as a result. Stringent material constraints contribute to a more forceful momentum toward some goals at the expense of

others.

The Headache of Complexity

One of the other main advantages to the metaphor that I have advocated is that it allows us to deal with enormous complexity in a way that “value-laden-ness” and “value neutrality” do not. It goes beyond simply categorizing values as something “inside” or “outside” the object, and instead insists that we place people and objects in much-needed contexts with respect to one another. However, complexity comes at a high cost. To understand the politics of any technological system *completely* – or even to understand one small component of that system – requires that we map out a dizzying maze of interactions. Even if we consider one small piece of technology in isolation, to have a complete understanding of the values enacted in its interactions we would need to look at each “input” and “output” force in order to get a sense of the system as a whole. This keeps us from making quick judgments like “nuclear reactors are bad,” and instead forces us to articulate what behaviors or systems nuclear reactors facilitate or limit, and to sort out which systems contributed to the formation and/or reinforcement of those reactor systems. Here again, I think that once problematic terms like “embodiment” are removed from the debate, Pitt and Winner ultimately would find themselves in agreement as to the ultimate goals of doing social histories of technology. Similarly, just as the momentum metaphor forces us to complicate our view of technology and values, it also keeps us from being able to make the judgment that “decision-maker (X) is bad.” We need to look at what factors influenced the decision-making individual in the context in which the

decision was made, and what outputs the decision is likely to obtain. Only in looking at those interactions can we form an accurate picture of the politics of the decision.¹¹

Many philosophers – particularly those in the field of ethics – may be made uncomfortable by the suggestion that we can treat human beings and technologies similarly in terms of their contribution to values within systems of momentum. The main objection is likely to be one of moral culpability. Isn't it the case that we want to hold human beings morally accountable for what transpires from the interactions in which they engage, and isn't it the case that we don't want to assign such moral accountability to objects? Isn't there some fundamental difference between the two? Indeed, the idea of putting a bridge or a piece of software on trial for moral crimes does seem absurd. It is this very issue that makes the “value-neutrality” approach to technology so appealing. If people and technologies are all together in the same interactive moral soup, then how can we hold humans to moral standards, and from where would those standards derive?

Here, however, I would argue that my proposed system does at least no worse than its predecessors. In Winner's case, saying that artifacts have politics is by itself insufficient

¹¹ David Bella gives a terrific account of complexity theory and emergent phenomena as they relate to decision theory in his paper “Organized Complexity In Human Affairs: The Tobacco Industry” (1997). In it, he argues that treating organizations as individuals with goals, intentions, and morals is misguided, and that the outputs of the organization cannot be reduced to the sum of the organization's parts. Rather, each individual within an organization may act on what seems like the organization's best interest from the perspective of his or her position with the larger structure, while the net result of each individual doing so may in fact be negative for the organization as a whole. In other words, a group of truly well meaning individuals may comprise an organization that, when taken as a whole, has the appearance of bad intentions. Bella makes a compelling case for mapping out the interactions within the organization and understanding them within the context of the whole, rather than pinning blame on particular individuals or attributing malice to the organization itself. At least on the organizational level, Bella argues quite convincingly that the “moral status” of an organization is an emergent (non-reducible) phenomenon resultant from very complex interactions between its constituent parts.

to make any normative moral claims. If artifacts contain moral agendas, we still need some ethical framework to determine which moral agendas are good (or desirable) and which are bad (or undesirable.) The same is true of Pitt's value-neutrality. If people are the only things that embody value systems, then we still need an ethical tool kit by which to make normative claims about which value systems we ought to encourage in people. Likewise, what I am attempting to offer here is a metaphorical description of where I believe values lie – in interactions. Like the other two systems, the question of what sorts of interactions we ought to facilitate or impair will require some separate ethical framework for determining value. It does, however, turn the issue of moral culpability at an admittedly odd angle. The system of momentum that I have described has very little affinity for passing blame. As mentioned earlier, it doesn't provide us with a world in which we can make the context-free assessment that some particular person is a bad person. What it can do, however, is to talk about the interactions in which that person engages. So instead of asking whose fault a particular failure is, we have to ask what systemic conditions allowed the fault to take place, and how we might alter the system in order to impair such behaviors or interactions in the future. As far as human moral culpability goes, this still allows for an ethical system with a rich basis for reward and punishment. Once we've determined what sorts of interactions are desirable, we can begin orchestrating the participants (both human and non-human) within any system towards facilitating those interactions. So putting people (or objects) on trial only makes sense in so far as it encourages or discourages certain types of interactions with the world and with each other. This seems to me perfectly reasonable. And, should the day arrive

when we need to put an artificially intelligent robot on trial for first-degree murder, it spares us much needless agonizing over the moral status of robots. We can instead assess the interactions of that robot with its world, and make a judgment based on those interactions. As such, it may be that the system that I have proposed is in fact *more* robust in terms of moral culpability.

Do Artifacts Have Politics?

To return to our original question: Do artifacts have politics? The problem with finding an answer, it seems, is that the question itself is poorly posed. If we answer “yes,” then we seem forced to explain how we can get the politics “out” of the artifact, in order to put it toward other uses. If we answer “no,” then we seem forced to explain why some artifacts are better suited for enacting some value systems than others (or, indeed, why we ought to study artifacts at all in understanding political structures.) More sensible, I think, is to throw out the question as a dead-end linguistic dispute. Artifacts do not “have” politics any more than a falling rock is “laden” with momentum. To suppose otherwise is to take an outdated Aristotelian view on the metaphysics of values, and to damn ourselves to semantic quibbling which has outlived its usefulness.

Instead, I think we can find hope and a chance to move forward in the metaphor of technological momentum, and in Kranzberg’s first law, which we are now in a good position to explain. “Technology is neither good nor bad, nor is it neutral.” The “goodness” or “badness” of a technology is a function of the context in which it dwells,

and what behaviors it is capable of facilitating or impeding. But not all technologies function equally well in all contexts. Momentum, both in the forms of material limitation and human habit, plays a huge role how people and technologies interact with their contexts. Understanding those contexts, technologies, and forces of momentum (and embracing the complexity that comes along with them) can ultimately only improve our understanding of the politics of artifacts.

BIBLIOGRAPHY

- “@stake Research Labs Overview.” @stake, Inc. <<http://www.atstake.com/research>>.
- Bank, David. “Satan Costs Software Creator His Job.” *The Orange County Register*. 23 March 1995.
- Bella, David. “Organized Complexity in Human Affairs: The Tobacco Industry.” *Journal of Business Ethics*, Vol. 16, 1997, pp. 977-999.
- Berleur, Jacque, and Klaus Brunnstein. *Ethics of Computing: Codes, Spaces for Discussion and Law*. London: Chapman and Hall, 1996.
- Bijker, Wiebe, Thomas Hughes, and Trevor Pinch, eds. *The Social Construction of Technological Systems*. Cambridge: The MIT Press, 1987.
- “Bizarre Answers from Cult of the Dead Cow.” *Slashdot* 22 Oct. 1999
<<http://interviews.slashdot.org/article.pl?sid=99/10/22/1157259&mode=thread>>.
- Bowyer, Kevin, ed. *Ethics and Computing: Living Responsibly in a Computerized World*. Los Alamitos: IEEE Press, 1996.
- Bowyer, Kevin, ed. *Ethics and Computing: Living Responsibly in a Computerized World*. 2nd ed. New York: IEEE Press, 2001.
- Bynum, Terrell, Walter Manner, and John Fodor, eds. *Computing Security*. New Haven: Research Center on Computing and Security, 1992.
- Castells, Manuel. *The Rise of the Network Society*. Oxford: Blackwell Publishers, 1996.
- “CERT Advisory CA-1993-14 Internet Security Scanner (ISS).”
<<http://www.cert.org/advisories/CA-1993-14.html>>, 1997.
- “COPS README File.” <<ftp://ftp.jaring.my/pub/cert/tools/cops/cops.1.02.README>>, 1998.
- Denning, Dorothy, and Herbert Lin, eds. *Rights and Responsibilities of Participants in Networked Communities*. Washington: National Academy Press, 1994.
- Edgar, Stacey. *Morality and Machines: Perspectives on Computer Ethics*. Sudbury: Jones and Bartlett Publishers, 1997.

- Farmer, Dan. "Improving the Security of Your Site By Breaking Into It."
<<http://www.alw.nih.gov/Security/Docs/admin-guide-to-cracking.101.html>>, 1993.
- _____. "SATAN." <<http://www.fish.com/satan/>>, 1995.
- Forester, Tom, and Perry Morrison. *Computer Ethics: Cautionary Tales and Ethical Dilemmas in Computing*. Cambridge: The MIT Press, 1990.
- Forester, Tom, and Perry Morrison. *Computer Ethics: Cautionary Tales and Ethical Dilemmas in Computing*. 2nd ed. Cambridge: The MIT Press, 1994.
- "The Hacker's Ethics." *The Cyberpunk Project*.
<<http://project.cyberpunk.ru/idb/hacker%5Fethics.html>>.
- Harmon, Amy. "Computer World Expects Devil of a Time with Satan Program." *LA Times*. March 01, 1995. D1.
- Hester, D. Micah, and Paul Ford, eds. *Computers and Ethics in the Cyberage*. Upper Saddle River: Prentice-Hall, 2001.
- Himanen, Pekka. *The Hacker Ethic and the Spirit of the Information Age*. New York: Random House, 2001.
- Hughes, Thomas. "Technological Momentum In History: Hydrogenation In Germany 1898-1933." *Past and Present*, No. 44 (Aug., 1969), 106-132.
- _____. *Networks of Power*. Baltimore: Johns Hopkins University Press, reprinted 1999.
- "John T. Draper: Biography." *ShopIP Information Security Solutions*.
<http://www.shopip.com/about/john_draper_bio.html> 2003.
- Johnson, Deborah. *Computer Ethics*. 2nd ed. Englewood Cliffs: Prentice-Hall, 1994.
- Kaempf, Michel. "Re: Ethics." *Archives de lorraine*. 27 Nov 2000
<<http://www.via.ecp.fr/via/ml/lorraine/200012/msg00004.html>>.
- Kranzberg, Melvin. "Kranzberg's Laws." *Technology and Culture* 27 (1986): 544-560.
- Latour, Bruno. *Pandora's Hope: Essays on the Reality of Science Studies*. Cambridge: Harvard University Press, 1999.
- Levy, Steven. *Hackers: Heroes of the Computer Revolution*. Garden City: Anchor Press, 1984.

MacKenzie, Donald, and Judy Wajcman, eds. *The Social Shaping of Technology*. 2nd ed. Buckingham: Open University Press, 1999.

Mentor, The. "The Hacker Manifesto."
<<http://www.via.ecp.fr/via/ml/lorraine/200012/msg00004.html>>

McClure, Stuart, Joel Scambray, and George Kurtz. *Hacking Exposed: Network Security Secrets and Solutions*. Berkeley: McGraw-Hill, 1999.

"Nmap Network Security Scanner Man Page." *Insecure.com LLC*.
<http://www.insecure.org/nmap/data/nmap_manpage.html> 2003.

"Nessus." <<http://www.nessus.org/>> 2003.

"Nessus FAQ." <<http://www.nessus.org/doc/faq.html>> 2003.

Parker, Donn. *Ethical Conflicts In Computer Science and Technology*. Arlington: AFIPS Press, 1979.

Pitt, Joesph. *Thinking About Technology: Foundations on the Philosophy of Technology*. New York: Seven Bridges Press, 2000.

"Quality Security Tools." *Insecure.org*. <<http://www.insecure.org/tools.html>> 2003.

Raymond, Eric Steven. "How To Become A Hacker."
<<http://www.catb.org/~esr/faqs/hacker-howto.html>>.

"SAINT Introduction." *SAINT Corporation*. <<http://www.saintcorporation.com/cgi-bin/doc.pl?document=intro#what-is-saint>>.

"Security Focus BUGTRAQ Mailing List." *Security Focus*.
<<http://www.securityfocus.com/archive/1>> 2003.

Spinello, Richard, and Herman Tavani, eds. *Readings in Cyberethics*. Sudbury: Jones and Bartlett, 2001.

Teich, Albert, and Mark Frankel, eds. *The Use and Abuse of Computer Networks: Ethical, Legal, and Technological Aspects*. Washington: American Association for the Advancement of Science, 1994.

Thomas, Douglas. *Hacker Culture*. Minneapolis: University of Minnesota Press, 2002.

Vidstrom, Arne. "Ethics and My Tools." *NTSecurity.nu*.
<<http://ntsecurity.nu/toolbox/ethics2.php>>.

Wark, McKenzie. "A Hacker Manifesto [version 4.0]."
<http://subsol.c3.hu/subsol_2/contributors0/warktext.html>

Winner, Langdon. *The Whale and the Reactor*. Chicago: University of Chicago Press, 1986.

"Worst Case Scenario." *Cult of the Dead Cow*. <<http://www.cultdeadcow.com/tools/>>.

CURRICULUM VITAE

Liam Kelly
205 Bennett St.
Blacksburg, VA 24060
540-951-0808
lkelly@vt.edu

Education

- Current PhD student in the department of Science and Technology Studies, Virginia Polytechnic Institute and State University, Blacksburg, VA. Expected graduation date: December 2006.
- MS in Science and Technology Studies (2003), Virginia Polytechnic Institute and State University, Blacksburg, VA.
- BA in English/Art (1997), College of Wooster, Wooster, OH.

Research Interests

- Master's Thesis – *Hacking Systems, Hacking Values: Interactive Theories For An Interactive World*. Considers the relationship between moral value systems and computer software development, particularly with regard to the question of whether computer security technologies should be considered “value laden,” and how that might affect their development and deployment
- Artificial biology and computer modeling
- Epistemology and technology

Employment

2002-present – **Graduate Teaching Assistant**, Department of Philosophy, Virginia Polytechnic Institute and State University, Blacksburg, VA

- Acted as GTA for introductory philosophy courses on ethics, epistemology, and metaphysics
- Led Friday recitation sections
- Met with students individually to discuss assignments and course material
- Graded examinations and papers

2000-2002 – **Senior Consultant**, L-Soft international, Landover, MD

- Prepared technical requirements documentation for L-Soft software development
- Provided training and consulting services on L-Soft products to external customers
- Assisted customers in planning and implementing their electronic mailing list operations
- Designed and taught training classes for LISTSERV and LSMTP administrators

1998-2000 – **Support Engineer**, L-Soft international, Landover, MD

- Provided customer support for LISTSERV and LSMTP on Windows, UNIX, VMS, and VM platforms
- Provided technical assistance to L-Soft sales and marketing staff
- Acted as Oracle database administrator for Oracle 8.1.6 on Tru64 UNIX
- Assisted with maintaining the L-Soft network of servers and workstations

1997-1998 – **Technical Support Representative**, Mindspring Enterprises, New Cumberland, PA

- Provided telephone and e-mail support for commercial and private dial-up Internet accounts

Conference Organization and Papers

- Assisted in the organization of the 2002 Mephistos graduate student conference at Virginia Tech
- Assisted in the organization of the 2003 Technology and Morality workshop at Virginia Tech
- Presented a paper (“Millions of Monkeys with Smarter Typewriters: The Epistemology of Hacking Software”) at the 2003 Computing and Philosophy conference at Oregon State University