AI & Soc (2006)
DOI 10.1007/s00146-006-0034-9

ORIGINAL ARTICLE

Alexander R. Galloway

"Carnivore personal edition": exploring distributed data surveillance

Received: 1 December 2004 / Accepted: 30 August 2005

Abstract The goal of this paper is to offer, in straight forward terms, some practical insight into distributed data surveillance. I will use the software project Carnivore as a case study. Carnivore is a public domain riff on the U.S. Federal Bureau of Investigation's software "Carnivore," which was developed to perform electronic wiretaps of email. As founder of the Radical Software Group (RSG), and lead developer on the Carnivore project, I will describe the technological, philosophical, and political reasons for launching the project. I will also offer an account of the development cycle of the core engine, identify trends in "client" interface designs, and present a series of design challenges that still remain.

Keywords Surveillance · Data · Network · Software

1 Introduction

On October 1, 2001, 3 weeks after the terrorist attacks in the U.S., the Radical Software Group (RSG) announced the release of CarnivorePE, a public domain riff on the notorious FBI software "Carnivore." The September 11 attacks brought on a crush of new surveillance activity. Rumors surfaced that the FBI had installed their eavesdropping machines on broad civilian networks like Hotmail and AOL with the intent of intercepting terror-related communication. This new reality underscores some of the problems of surveillance today. First, data snooping is often associated with uses of power, be they legalized uses of power by the FBI or illegal abuses of power by black-hat hackers. Second, data surveillance is usually deployed at centralized locations within networks such as Internet Service Providers, not at the distributive fringes of networks. With CarnivorePE, RSG sought to invert these two problems by focusing first on more equitable and consensual uses of the technology instead of covert uses, and second on Local Area Network (LAN) traffic instead of broad network

A. R. Galloway
Culture and Communication, Department of Culture and Communication,
New York University, 239 Greene Street, 7th Floor, New York, NY 10003, USA
E-mail: galloway@nyu.edu · Tel.: + 1-212-9985423

communications. Using the CarnivorePE surveillance engine, one is able to visualize and interpret LAN traffic with a minimum of programming knowledge. To date, dozens of Carnivore interfaces have been created by artists and scientists around the world. This essay covers the goals of the project, the successes and failures of collaboration, as well as the pedagogical uses of the software.

What is Carnivore? It has its roots in ''DCS1000,'' a piece of software used by the American Federal Bureau of Investigation (FBI) to perform electronic wiretaps of email. Until recently, DCS1000 was known by its nickname ''Carnivore'' (EPIC 2005). In 2000 the software collective RSG undertook an initiative to release the FBI's Carnivore into the public domain. After a fair amount of guesswork, plus some deliberate deviations in design, RSG released a prototype for the public domain Carnivore in 2001 and by 2004 was supporting a stable release of the application for Windows and Macintosh. At the heart of the project is CarnivorePE (Carnivore Personal Edition), a software application that listens to all Internet traffic (email, web surfing, etc.) on a specific Local Area Network (LAN) (RSG 2001). Next, CarnivorePE serves this data stream to interfaces called ''clients.'' These clients are designed to animate, diagnose, or interpret the network traffic in various ways. (For the rest of this paper, the name ''Carnivore'' will refer to the RSG version of the software, not the FBI version.)

The Carnivore project sprung from a three-part desire to change the status of data surveillance in contemporary discourse on new media. First was a philosophical desire to break out of the restrictive dichotomy of data surveillance as either a threat or a salvation: on the one hand data surveillance today is associated with dystopian visions of the government spying on the private lives of normal citizens; but on the other hand data surveillance is cited as crucial to the American-led war on terror and therefore necessary, even beneficial, for monitoring potential threats. Neither of these two discourses is particularly satisfying. The dystopian discourse is unsatisfying because it is patently untrue: today's data tracking technologies are deployed much more widely in the commercial or civic sector than they ever have been in the juridical or governmental sector. Government spying initiatives pale in comparison to the massive amount of data tracking employed in fields such as commodity logistics or data mining (e.g., FedEx or Google). Further, much of this type of data surveillance is entered into on a purely non-coercive basis—the surveillance network for personal financial transactions using credit and debit cards coalescing in the ''big three'' credit bureaus, Experian, Equifax and TransUnion, is a good example in this regard. But the rhetoric of data surveillance as necessary and beneficial in law enforcement is also unsatisfying. This rhetoric is generally deployed in strategic ways to leverage certain political outcomes. Yet to date in the U.S. these strategies have also failed. On September 2, 2004 a Detroit federal judge overturned the sole jury conviction achieved by the U.S. Justice Department since the passage of the Patriot Act in 2001. This means that the increased powers for surveillance granted to American law enforcement have not resulted in a single terrorist conviction, yet threats to civil liberties have increased dramatically in the mean time (as witnessed in the FBI's ongoing harassment of the artists and authors Critical Art Ensemble). In response to this Carnivore is an attempt to explore a ''third way'' for data surveillance that is neither threatening nor liberating. The ''third way" focuses on localized, personal uses of network data, not the aggregation and sequestering of that data by powerful interests. In this sense the project is organized quite differently from how data surveillance and tracking is deployed by FedEx, TransUnion, or even the FBI.

Second was a technological desire to pull back the veil of secrecy surrounding networks and how they actually function. For non-scientists, knowledge of how networks work is relatively rare. Carnivore offers an easy one "one-click" tool for artists (and scientists), allowing them to get beyond any initial reticence toward the nitty-gritty of network protocols. Existing packet sniffers (a piece of software able to listen to all data traffic on a local area network), of which there are many, are often difficult to use and remain slightly opaque to those not trained in network administration. Carnivore thus helps demonstrate the ubiquity of packet sniffing technology—for example all Macintosh personal computers today ship with a packet sniffer called "tcpdump" (Fenner 2005) preinstalled, and Ethernet sniffers are as old as Ethernet, easily available for free on all operating systems—as well as promote awareness and understanding of the need for personal data encryption.

Third, Carnivore sprung from a political desire to generate media publicity via staging a "reverse-engineering" or "theft" of the FBI's Carnivore. The strategy was not one of critique or resistance but instead one of *hypertrophy* whereby the existing technology is developed further, pushed beyond its intended use. RSG's goal was to extend the FBI software, improve it, and in so doing inject a new design philosophy into the technology. The hope is that it will both increase public outrage over the excesses of data surveillance and also increase public awareness of those same technologies. Specific retrograde features of the FBI software that RSG wished to avoid include: secrecy, a centralized architecture, and the poverty of data interpretation—the FBI may collect, but they are not able (cognitively and legally) to understand, visualize, or otherwise use the data in a better and more creative way. More explanation of these various desires will emerge below.

2 Design prototype

Started in 2000, the design prototype for Carnivore consisted of a three-part application architecture running on Linux: a packet sniffer, a server, and a client. The packet sniffer is responsible for actually eavesdropping on the local area network. The server takes each small snippet of data, called a packet, from the sniffer and serves it via a socket connection. The client connects to the server, receives the packets in real time and then processes the packets in some way (for example, visualization or keyword parsing). For the prototype the following solution was cobbled together using existing software: an Internet Relay Chat (IRC) server provides simple client/server functionality; a Perl script grabs the output of tcpdump (a packet sniffer application) and prints the output on one of the IRC channels; finally, a client application connects to the IRC channel to receive the data stream.

We learned several lessons from building the prototype. First, the separation of sniffer and server from client is beneficial. The reason is that many animation environments (for example, Macromedia's software Flash) are not

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                             |
+     +------+                                +
|    /      /|         /-----------\          |
+   +------+ |        /...internet...\        +
|   |      | |        |............+----+     |
+   | data |<--------->|............|user|    +
|   |      | |    ^   |............+----+     |
+   |      |/     |   |..............|        +
|   +------+      |   \............../        |
+                 |    \------------/         +
|                 |                           |
+            +---------+                      +
|            |CARNIVORE|                      |
+            +---------+                      +
|                                             |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Diagram for Carnivore

able to perform packet sniffing simply due to their own limitations on low-level access to the machine's network adapter. By grouping the sniffer and the server together, a greater variety of clients can be written. The clients may be written in any computer language that is able to create a socket connection. Additionally, clients may run on a different computer entirely from the sniffer-server. This frees the project from being wholly limited to one operating system.

At the request of the beta testers (a small group of artists building Carnivore clients in 2001 and 2002) several features were added to the original prototype. The first was the addition of three different output channels. This happened as a direct result of the IRC server model. The first channel, dubbed simply "carnivore," converts each packet to ASCII characters (a space is inserted for any character outside of the ASCII range) and outputs the packet as text. This channel is designed mainly for clients wishing to perform keyword matching. The second channel, "hexivore," converts each packet to text-formatted hexadecimal values and outputs the entire packet as text. This channel is designed for clients who require the original packets in their entirety (for example to reassemble an original datagram). The third channel, "minivore," extracts only a few pieces of header information from each packet (timestamp, source IP address, source port, destination IP address, and destination port) and outputs this header as text. This channel is used for any client that is interested only with the source and destination of packets, not their contents. This is advantageous in some situations because it decreases load on Carnivore and on the client, due to the much smaller amount of data needed to be parsed and transmitted from server to client.

Another feature added as a result of beta testers' requests was volume buffering. This allows the user to declare the maximum packets per second that will be sent by Carnivore to the client, whereby avoiding debilitating "storms" of high traffic spikes. This feature makes Carnivore more compatible with clients programmed in languages such as Flash which may have a hard time parsing spikes of ten or more packets per second.

Other features include the ability to allow or block UDP packets, and the option to allow or block client connections from other machines. An IRC-style client username and password login, which was a legacy feature from the original prototype designed as a security precaution to protect from unwanted remote connections, was deprecated after the prototype stage. The IRC-style server had allowed for multiple simultaneous clients. Though it was determined that this feature was not a priority, because client designers tended not to run multiple clients on a single installation of Carnivore, support for ten simultaneous clients remains. The final two features lacking were (1) the ability for clients to switch channels programmatically by sending a message via the socket connection, and (2) support for switching between multiple network adaptors (wired Ethernet, wi-fi, etc.).

While the design limitations of the first prototype were easily solved, several other issues prompted a more fundamental redesign of Carnivore. We became concerned that the server had became privileged in relationship to the client. This was objectionable because it appeared to lapse back into the more centralized model of the FBI Carnivore that we were trying to avoid. Likewise the "web server" model of a single data source and multiple clients was not the goal of Carnivore.

Another design concern was the separation between the location of the Carnivore server and the location of the client. Unfortunately by separating these, and having a single Carnivore server (which during the prototype phase was running inside the offices of the nonprofit arts organization Rhizome.org at 115 Mercer Street in New York City), we were exacerbating rather then ameliorating the threat to individual privacy. The computer users at the server installation site in essence were required to "sacrifice" their data for use by the clients.

Other problems with the prototype included the fact that the application, composed as it was of three separate Perl scripts, was not very robust and crashed often. For this reason the project was also difficult for others to install and use. Finally, while Perl is indeed portable to other platforms, Carnivore was limited, without some gerrymandering, to Linux or Unix machines.

3 Development of Carnivore Personal Edition

The release of Carnivore Personal Edition (CarnivorePE) on October 1, 2001 was an attempt to solve problems with the first prototype and to test the platform in the public sphere. (The title "CarnivorePE" mimics marketing strategies for commercial software: WindowsXP, FlashMX, etc.) The various design developments can be clustered around two goals.

The first goal was to promote a distributed model, rather than client-server model. This means that instead of having a central Carnivore server, which clients can connect to from remote machines, the Carnivore application would be installed on each machine running a client. So the act of data surveillance would then be focused around an individual machine sniffing and listening from its own position in the network, and at the same time, making a localhost connection to the client and performing all visualization and diagnosis locally. This goal would reduce stress on the Carnivore server, both in terms of data volume (localhost connections produce no additional LAN traffic) and also in terms of the concerns for individual privacy (Carnivore client artists sniff their own traffic, not traffic from other LANs). In short, the shift was to make Carnivore a local "engine" rather than a remote "server." This meant that the project tended toward having as many installations of CarnivorePE as there are clients. Each carnivore client is running its own local CarnivorePE engine. This shift from server to engine also further dilutes privacy concerns because client artists end up sniffing themselves, and their immediate environment, instead of spying on a Carnivore server in another state or another country.

The second large goal was to make a one-click, robust application for both Windows and Macintosh. Sensing that a Linux-only application would severely limit the potential usership for Carnivore, we decided to port the application to Windows and Macintosh, while still publishing the Linux schematic for those desiring to recreate the prototype. This platform shift increased ease of use and accessibility, particularly for artists.

4 Community aspects

The Carnivore project, like several before it, allows broad collaboration between multiple Internet artists who may typically work independently. It also created an opportunity for Internet artists to experiment with packet sniffing who may not have had the notion to do so before. A good example is Lisa Jevbratt's client "Out of the Ordinary" (2002): Ms. Jevbratt had worked for several years visualizing aspects of the Internet as a whole, but with CarnivorePE she was able to turn her attention also to local, real-time traffic. Carnivore also presents an opportunity for newcomers to experiment.

In addition, the open source aspect of Carnivore has generated a small community of developers swapping ideas and tips. The CarnivorePE code is open source (Visual Basic code for Windows; Cocoa/Objective-C code for Macintosh) and has benefited from the contributions of several different coders in different cities. Additionally, with assistance from the Open Art Network, we have released open source client templates in six different programming languages and environments (Flash, Director, Java, Processing, Perl, and Visual Basic). Uncoupling the Carnivore platform from specific computer languages allows newcomers to get up and running using Carnivore with minimal effort.

CarnivorePE is also a useful teaching tool for the classroom. It has been used both in fine art and design courses, as well as networking and computer science courses. In my own use of the software in the classroom, I find that it offers a rare convergence for students interested in both political questions related to data surveillance and technological questions related to protocols and LAN administration. For example CarnivorePE teaches students technical knowledge such as how the various protocol layers encapsulate data, how datagrams are fragmented into packets, how port numbers are assigned to application services, and how IP addressing arranges different classes of networks into hierarchies. Since network flows are based on user behavior, they are unpredictable. In this way, CarnivorePE is a nice option for generative projects that require a pseudo-random input source to trigger action. But CarnivorePE also makes students aware of the relative visibility and invisibility of data flowing through networks, and the social debates surrounding privacy that follow from this. For example, CarnivorePE makes it clear to students very quickly that most email and web data is sent as clear text. Because of this, students generally begin to think in very different ways about the necessity for encryption due to the ubiquity of packet-sniffing technologies. The lesson is, to borrow a mantra from network security analysts, "don't prohibit what you can't prevent." In other words: don't prohibit infringements of privacy as a moral good unless you can't physically prevent those infringements (via encryption or other means).

The impact of the project has been quite broad. Several internationally known computer artists have used CarnivorePE in their work such as Vuk Cosic (2002) and Golan Levin (2002). The Carnivore project has been exhibited at over twenty galleries and museums around the world and is being used as a teaching tool in a number of universities. Carnivore was awarded a Golden Nica in the 2002 Prix Ars Electronica and has remained on permanent view at Ars Electronica's Museum of the Future in Linz, Austria.

5 Three tendencies in client design

The various Carnivore clients that have been authored to date are all quite different. They have been written in a half dozen different programming languages, and span from self generating three-dimensional worlds, to diagnostic animations, to text-only keyword parsing. I will describe here three current trends in client design, but these should in no way indicate a preferred or correct style of client design. I simply offer an overview for someone not familiar with the range of clients. (A characteristic of Carnivore is that it imposes no restrictions on client design, nor does it suggest that one interpretation strategy is better than another).

The first group is what might be called "network maps." These include the client "amalgamatmosphere" by Joshua Davis, Branden Hall, and Shapeshifter (2001), "Fuel" by Scott Snibbe (2002), and "Synapsis" by Marcos Weskamp (2003). Network maps create a real time visual topology of the network traffic. They generally use the "minivore" CarnivorePE channel, and map the communications based on IP addresses and services gleaned by a reverse-lookup of the packet port numbers.

The second group uses keywords as triggers of various actions or behaviors. This group includes "Trigger Words #2" by RSG (2002), "JJ" by Golan Levin (2002), and "Police State" by Jonah Brucker-Cohen (2002). These types of clients use the "carnivore" ASCII channel to listen for specific words in email, websites, and other text-based communications. The words are then associated, using a dictionary or lookup table, with client actions.

The third trend in client design aims to determine the "shape of data" via abstract visualizations of the packets. Included in this group are "Active Metaphore" by Limiteazero (2002), and "Black and White" by Mark Napier (2001). These types of clients use specific numerical values from the packets to plot visual shapes, colors and sounds.

In the future it is expected that entirely different approaches will emerge for Carnivore client design. For example, no Carnivore artists have yet attempted to reassemble groups of packets into their original datagrams, thereby being able to recreate actual webpages, images, or emails in their entirety as they flow through the network. Or, there currently exist few clients that focus specifically on audio. Further, no clients thus far attempt to intervene in the network by injecting packets back into the stream. (Mark Daggett's client [2002] comes the closest. It sends email to users based on specific triggers in the data stream).

6 Ongoing design challenges

Challenges continue to exist in the project. The largest problem has to do with the changing nature of Ethernet traffic. Gigabit Ethernet traffic volume can pose a formidable problem for machines unable to handle the load. This poses a problem both at the level of CarnivorePE and the client. Even more important is the general transition in recent years from unswitched to switched Ethernet hubs. Switched Ethernet hubs generally render Carnivore ineffective for broad LAN surveillance, for they make it impossible to sniff LAN traffic that is not specifically targeted to the local machine. Running Carnivore on a switched network is still possible, and indeed certain clients could be designed to thrive in such a digital environment (by keying exclusively off broadcast packets, for example), however these types of LANs fundamentally alter the original context in which Carnivore was developed. It remains to be seen if switched LANs will kill the Carnivore project or not.

The second ongoing design challenge has to do with public fear of the project. From time to time Carnivore is invited to participate in galleries or museum exhibitions. However in the majority of cases, the invitation comes with a restriction: Carnivore may only be exhibited if it is quarantined into its own subnet. RSG generally complies with such restrictions, opting to participate in rather than avoid the public arena. However since a Carnivore machine sniffing itself does not offer very interesting results, we have been required to invent ways of generating supplemental data traffic as fodder. One technique is to install peer-to-peer software on the machine or somewhere within the LAN, since such applications generate a steady volume of packets.

7 Criticism of the work

Carnivore has benefited from a reasonable amount of public discussion. There have been two main threads of criticism thus far of the work.

The first criticism is that Carnivore is an opportunist project whose notoriety has benefited more from hype than from actual substance. This line of criticism attacks the project as being too "slick" and for having a commercial-looking GUI design, as well as trying to piggy-back on contemporary social themes that are considered hip or timely such as data surveillance. I consider this criticism to be without merit. Data surveillance and privacy are two of the most pressing themes in society today, particularly in recent years as tracking technologies have proliferated both in the commercial and governmental sectors. Further, the commercial style of the work was a deliberate attempt to escape the ghetto of so-called political art and intervene in culture on a much broader scale.

The second criticism might be called the "screensaver" critique articulated by Andreas Broeckmann (2002) and others. This approach claims that the various Carnivore clients merely resemble screensavers and don't achieve any level of aesthetic innovation or political nuance. This criticism claims that the potential for Carnivore to be steadfastly critical of the FBI or other data snoops has been diluted and sidetracked toward data visualization projects that lack commentary or critique. I also consider this criticism to be invalid, as it is both misdirected and inaccurate. First, the criticism is misdirected because it confuses specific Carnivore clients with the entire project itself (which is composed of the engine, CarnivorePE, plus an unlimited number of clients that use the engine). I can imagine a compelling critique of *specific* clients based on this anti-screensaver rationale, but since CarnivorePE imposes no restriction on how the data shall be visualized (or even that it *should* be visualized), it is misdirected to say that Carnivore as a project falls short in this regard. Second, the criticism is inaccurate because I consider there to be something profound in taking a mystified, obfuscated technical apparatus such as a data network, and unmasking it in order to make it visible. This is one of the fundamental methods in the areas of ideology critique or apparatus critique (in cinema studies, for example), which I endorse and consider to be at play in Carnivore as well. Screensaver or not, Carnivore still denudes a hitherto obfuscated apparatus, and this is a fundamentally valuable process.

The work has thus far not been criticized on legal grounds. To my knowledge, packet sniffing is illegal in the U.S. if it is covert or unauthorized. But noncovert uses of packet sniffers, for example when a network administrator uses tcpdump to diagnose a network outage, are legal. This was one of the reasons that prompted the decision to shift from a client-server architecture to a distributed architecture: the potential for abuse is diminished when the sniffer engine and the client both reside at the fringes of the network rather than at centralized hubs.

Thus far there has also been a general lack of criticism from the mainstream Internet public. Judging from unsolicited email and anecdotal comments on Slashdot and elsewhere, most Internet users are supportive of the project.

Surprisingly, and to my relief, there has been a complete lack of correspondence from the FBI or other government representatives.

8 Summary

As I have hoped to convey, Carnivore is an attempt to explore a "third way" for data surveillance that is neither threatening nor liberating. This was achieved first by making the software distributed rather than centralized, and second by uncoupling the surveillance engine from the interpretive clients thereby hindering the act of interpretation as little as possible. By pulling back the veil of secrecy surrounding networks, Carnivore allows non-scientists to explore network data traffic more easily. The various Carnivore clients written to date have resulted in entirely novel ways for visualizing and interpreting data traffic and in doing so start to move beyond the somewhat limited consideration of data surveillance as it has existed thus far in the public imagination.

References

Broeckmann A (2002) ''Re: < nettime > How We Made Our Own 'Carnivore''' Nettime. http://www.amsterdam.nettime.org/Lists-Archives/nettime-l-0206/msg00114.html (accessed August 27, 2005)

Brucker-Cohen J (2002) ''Police State.'' http://www.mee.tcd.ie/~bruckerj/projects/policestate.html (accessed August 27, 2005)

Cosic V (2002) ''History of Art for the Intelligence Community.'' http://www.ljudmila.org/~vuk/intelligence/ (accessed August 27, 2005)

Daggett M (2002) ''Carnivore Is Sorry.'' http://www.flavoredthunder.com/dev/rhizome/carni/carnivore-is-sorryPE.exe (accessed August 27, 2005)

Davis J et al (2001) ''amalgamatmosphere.'' http://www.ps3.praystation.com/pound/assets/2001/11-20-2001 (accessed August 27, 2005)

EPIC (2005) http://www.epic.org/privacy/carnivore/ (accessed August 27, 2005)

Fenner B et al (2005) Tcpdump. http://www.tcpdump.org/ (accessed August 27, 2005)

Jevbratt L (2002) ''Out of the Ordinary.'' http://www.jevbratt.com/out_of_the_ordinary/ (accessed August 27, 2005)

Levin G (2002) ''JJ.'' http://www.flong.com/jj/ (accessed August 27, 2005)

Limiteazero (2002) ''Active Metaphore.'' http://www.limiteazero.com/carnivore/ (accessed August 27, 2005)

Napier M (2001) ''Black and White.'' http://www.potatoland.com/blackwhite/ (accessed August 27, 2005)

RSG (2001) Carnivore. http://www.rhizome.org/carnivore (accessed August 27, 2005)

RSG (2002) ''Trigger Words #2.'' http://www.rhizome.org/print.rhiz?3217 (accessed August 27, 2005)

Snibbe S (2002) ''Fuel.'' http://www.snibbe.com/scott/fuel (accessed August 27, 2005)

Weskamp M (2003) ''Synapsis.'' http://www.marumushi.com/apps/synapsis/ (accessed August 27, 2005)