DAVID DITTRICH

the conflicts facing those responding to cyberconflict

Dave Dittrich holds an appointment as an affiliate principal scientist with the Applied Physics Lab at the University of Washington. He has studied distributed denial of service (DDoS) attack tools, botnets, host and network forensics, and the legal/ethical framework for responding to computer attack (the "Active Response Continuum") for over a decade.

dittrich@speakeasy.net

THERE IS INCREASING TALK OF CYBER-warfare, where the computers of private citizens are the weapons being used. Out of a sense of frustration, some call for a right to self-defense and for going on the offensive against cyberattackers. Researchers today are regularly doing things that cause effects visible to others and publishing information under the banner of full disclosure without supporting their decisions through a systematic analysis of both the legal and the ethical issues involved. We are entering a dangerous time and have a lot to talk about and agree upon, lest someone with good intentions causes massive harm.

The purpose of this article is twofold. First, I want to encourage the computer security community to discuss how ethics apply to responses to cyberconflict. Second, I want those outside the computer security community to understand how much more damaging and serious the situation is becoming. Such an understanding is necessary to help get the policy and legal changes necessary to address today's increased threat landscape in ways that are acceptable to society.

I use my own story here as a case study in how some of these issues are raised and how they can be addressed, centering on events that led to the first distributed denial of service (DDoS) attacks over a decade ago. Anyone reading my analysis of the trinoo distributed denial of service attack tool written in October 1999 and released to the public on December 30, 1999, would have noticed the following:

> Trinoo daemons were originally found in binary form on a number of Solaris 2.x systems, which were identified as having been compromised by exploitation of buffer overrun bugs in the RPC services "statd", "cmsd" and "ttdbserverd". These attacks are described in CERT Incident Note 99-04:
>
> http://www.cert.org/incident_notes/IN-99-04.html
>
> The trinoo daemons were originally believed to be UDP based, access-restricted remote command shells, possibly used in conjunction with sniffers to automate recovering sniffer logs.
>
> During investigation of these intrusions, the installation of a trinoo network was caught in the act and the trinoo source code was obtained

from the account used to cache the intruders' tools and log files. This analysis was done using this recovered source code.

These statements play down the significance of distributed intruder attack tools, an advance that was taking place beyond the gaze of the public. Attacks have become more automated, more sophisticated, and more complex. This has put great pressure on incident responders to deal with the increase in abuse and compromised systems. Incident response teams face a choice. They can take the easy route, wiping and re-installing systems and spending just enough effort to keep up with the onslaught. Of course this "easy way out" makes it harder for law enforcement to do their job, possibly resulting in more harm to society (a concept known as *externalizing costs*). Or they can make the effort to find ways to be more efficient, effective, and proactive at countering cybercrime. This might mean taking aggressive actions to home in on crucial evidence for attributing criminal acts, or identifying key attacker assets (e.g., command and control servers) and finding ways of taking those assets out of the hands of attackers to neutralize their ability to cause harm to society. The latter option has its own potential risks of harm to society—starting with loss of privacy, but ranging up to possible disruption, destruction of computer data, or even physical damage—which can be mitigated through ethical decision-making that systematically balances potential harms and benefits.

The Advent of Distributed Attacks

Distributed denial of service (DDoS) became widely known when high profile targets such as Yahoo, CNN, Amazon, and eBay were attacked in February 2000. Denial of service itself was not new, but the remote control of thousands of computers at a time was. And it started many months earlier than the public knew. What had once been manual and limited to the number of people who could type on command lines became automated, distributed, and allowed a handful of malicious actors to create orders of magnitude more damage than before.

SNIFFER ATTACKS

In the mid to late 1990s, computer intrusions involving the installation of *sniffers*, programs that monitor network traffic for the purpose of stealing login credentials, were rampant. Many sites were still using older TCP/IP protocols, such as telnet, ftp, imap, pop, and rlogin, for remote terminal sessions, file transfers, and email. The problem was that networks at the time were often wired using *thin-wire Ethernet*, a shared network medium on which any host was capable of seeing all network traffic to/from any other host. Login names and passwords were visible to anyone who could control a computer on the same network segment. The result at large universities was massive account theft and abuse. Someone possessing a list of several hundred stolen accounts could hop from account to account, remaining active within the network for over a year. Intrusions also spread quickly from one host to many other hosts and many other sites. There was still a limiting factor: attackers had to manually install sniffers and come back later to retrieve the sniffer logs. The solution: client-server computing!

DISTRIBUTED SNIFFERS

In the early months of 1999, an attentive system programmer, responsible for the large clusters of IBM AIX systems that the University of Washing-

ton (UW) made available centrally, noticed some odd processes that showed very long up times. He had wisely obtained process memory dumps before killing the processes and made note of listening network ports that indicated that the program might be a remote access trojan.

I analyzed the memory dumps and identified what looked like a simple array data structure with a user name, a password, and a numeric value. I identified the numeric values as UNIX timestamps. Comparison of last login records showed a correlation between the account names and timestamps. Analysis of process lists and login records on other cluster members for the accounts involved showed that at one point, over a month earlier, someone had started running the same program on all hosts in the cluster. What we had discovered was the first known distributed sniffer for IBM AIX systems. Now the sniffer logs from all hosts in the cluster could be automatically retrieved over the Internet in rapid succession, increasing the scale and speed of credential harvesting.

While a single host on a shared Ethernet segment could net a few hundred login credentials in a month, having a local sniffer on every host in a large cluster used by tens of thousands of people could potentially yield orders of magnitude more. This was not the only type of attack being automated, though.

DISTRIBUTED DOS TOOLS

Denial of service attacks in the late 1990s relied on using stolen accounts to run programs such as synk4, teardrop, or smurf from the command line. The first two programs implemented point-to-point attacks, where bandwidth and the number of accounts used dictated who would win. A *smurf attack* was a form of reflected and amplified DoS attack that exploited poorly configured network routers and required far fewer accounts to initiate the attacks. Regardless, such floods were straightforward to identify and trace back to the accounts used to run these attack programs, which could then be easily shut down. The primary limiting factor was the number of attackers who could log into stolen accounts, download DoS attack programs, and initiate attacks.

Things changed significantly in the summer of 1999. UW began to get reports of DoS attacks against different sites around the world, all involving dozens of UW systems *all at the same time*. Some unknown program was being installed and run on dozens of compromised computers. It was controlled through remote connections from a small handful of central locations, using an unknown protocol. It looked similar to the distributed sniffers and had the same problem of being indirectly controlled. And it was capable of flooding remote hosts in several different ways, keeping them offline for days at a time.

Nobody had ever dealt with this kind of distributed attack before. Nobody knew exactly how it worked. Worst of all, nobody knew how to stop it. We desperately needed a detailed understanding of how to identify these distributed denial of service programs on infected computers, how to identify them by observing network traffic patterns, and how to scan our networks to quickly find infected hosts and get them cleaned up. Our responses had to scale as well as these new attacks, both in social and technical terms.

My Story

Responding to account abuse involves determining whether the account holder knowingly misused their account or gave their password to someone who abused the account (a policy violation), or unknowingly had their login credentials stolen by an outsider (i.e., they are innocent). One method of investigating these attacks involves examination of account contents, including files stored in the account and saved email messages. To respect the privacy of the user, while learning who was responsible for account abuse, I adopted an investigative method, patterned on the FBI's procedure used when wiretapping, known as *minimization*. This means starting with the least invasive methods first and only using more invasive analysis if evidence warranted it. Searching for keywords like *password*, *pwd*, *pw:*, *account*, *acct*, *acc:*, etc. could indicate purposeful sharing. Only if I found such keywords would I then expand the search to include the email headers for Subject, To, Cc, From, Sender, Date, etc. to provide more context. Finally, I would only look at the specific messages (identified by the header lines) and the specific paragraph in which the keywords occurred. If it looked like that portion of message had nothing to do with account abuse, I would immediately stop reading and go on to the next suspect message. My task was only to verify account sharing, not read personal communications. This conformed with policies for protecting UW systems, maximizing efforts to secure UW systems, and minimizing intrusion into account holder privacy.

The same minimization techniques can be applied to analysis of the content of files stored in suspect accounts. By correlating login history with the creation or modification dates of files in the account and looking at their names, it was possible to identify those files that were created during periods of suspected abuse (e.g., a DoS attack, spam run, or scanning activity). It was not necessary to wade through any/all files, which would be more invasive to privacy. Correlating this information across multiple abused accounts often illuminated a pattern. One host, or one domain name, might show up as the source for logins across multiple abused accounts. By looking at the exploit programs stored in the stolen accounts, it was sometimes possible to identify a specific target (e.g., Linux running imapd). A quick scan of the suspect network for any hosts with the vulnerability being used by the attackers often allowed me to locate the host running the sniffer. I would then target that host for forensic analysis.

I wrote scripts that parsed the log files produced by several of the most commonly used malicious sniffers. These sniffer logs typically showed the source and destination hosts, the login account name, what protocol or service was involved, and the first couple of dozen to couple of hundred characters typed. This latter text is where the password is found and sometimes also the first few commands that were typed (e.g., logins to other hosts, the root password in `su` or `sudo` commands). This script allowed me to extract a list of all accounts that were compromised by the sniffer and on which systems those accounts existed.

I modified another script, originally written by a brilliant programmer at UW named Corey Satten, to extract lines by domain name, IP address, and even network block in Classless Internet Domain Routing (CIDR) notation. I could process the compromised account and host lists from the sniffer logs and split them up by (1) those at UW and (2) those at remote sites. Another script would iterate over the list of remote sites and send one email per site reporting those accounts or hosts at just that site. Rather than sending one big list to everyone, which would expose information about all victims, I could do targeted reporting. This allowed the efficient reporting of compro-

mised assets at *all sites affected at one time* and I could proactively identify and remove from attacker control *all compromised resources at one time* instead of waiting for abuse reports to come in and spending far more time handling them individually. While this kind of response was more complex, on several occasions it resulted in distant hackers leaving the UW network and not coming back. This was fighting automation of attacks with automation of defensive response. The technique was quite effective, at least in those cases where affected sites were capable and cooperative in removing access to malicious hackers.

During the routine investigation of a suspected sniffer on a UW subnet in the summer of 1999, I was able to locate the sniffer on a Linux workstation, retrieve its log file, and began extracting account/host information to do the all-at-once cleanup. One entry in the sniffer log caught my eye, however. It showed a connection from a *different* host on the same shared network segment, but sourced from a computer owned by a completely different research group. Such shared networks were the easiest way for an intrusion to spread quickly. Effectively, one hacker's sniffer had managed to capture a connection exposing *another hacker's* activity! I recognized the name of the second computer as one of the Sun computers involved in DDoS attacks weeks earlier. The sniffer had been running long enough to capture historical evidence of the installation of a DDoS agent. I now knew where the DDoS attackers were caching their trinoo agent binaries (and possibly much more, given recent experience). Had I simply reported the sniffer to the owner, who would have been tempted to just wipe and reinstall the OS, this key piece of information would have disappeared forever.

Recognizing the significance of what I had learned, I quickly put on my headset, looked up the technical contact information from the relevant domain registry and initiated a telephone call. My intention was to (1) contact a responsible party at the affected site to report the intrusion into their network and (2) request that they preserve evidence and provide me with a copy of files in the directory to analyze. While they might not (and odds were good they probably didn't) have a skilled incident response team, with someone who understood computer attack tools, networking, programming, and scripting, I was prepared to make use of the information to try to put a stop to the harm being caused to systems around the globe. If this company didn't report the incident to law enforcement, it meant that the investigation hit a dead end. If they did report it to law enforcement, it might take weeks or months (if ever) to complete a detailed analysis. Even if such analysis was performed, it might not be widely enough distributed to achieve worldwide mitigation of the event. The likelihood was that if I did not get my hands on the data in that account, key information would simply disappear and the damage worldwide would continue to escalate.

When making blind contact with remote sites in situations like this, in the middle of active hostile activity, there are many common outcomes I have encountered over the years. Most of the time, explaining who I am and what I am doing results in cooperation by the site being contacted. Sometimes they even offer to provide a root password and let me clean up the system, which I decline and instead provide guidance to help them clean up their own systems. Sometimes they say, "Thanks for reporting this to us," and immediately take the system off-line, wipe the hard drive, and reinstall the operating system. This destroys most/all evidence on the system. They may not even report to law enforcement, which results in a dead end for investigation. Sometimes the person denies there could possibly be any problem or gets mad and hangs up. Sometimes they are distrustful and assume I am somehow involved. In a handful of cases, the person I spoke with professed

to be helping to stop the intruder, even asking that I contact them immediately whenever I noticed this person using their systems. They later turned out to be actively helping the intruder, or they themselves *were* the intruder. Sometimes my report gets sent around the organization in email which happens to be actively monitored by the bad guys.

I reached the operator at the victim site and informed them I had reason to believe that one of their main computer servers had been compromised and that I needed to speak with someone responsible for computer security investigations at their company. My call was transferred. The conversation began something like this:

> Hello. My name is David Dittrich and I am a computer security engineer at the University of Washington in Seattle. If you wish to verify my identity, you can call the main switchboard at the University of Washington and ask to be transferred to me, or get my contact information from my Web site, which you can find with a search engine by entering my name. I understand if you don't trust what I am saying to you.
>
> I am investigating a series of intrusions at the University of Washington that involve distributed denial of service attacks involving thousands of computers around the world. One attack last month disrupted the entire campus of the University of Minnesota for over three days. These attacks have been reported to the FBI and are under active investigation. I have a report that I can send you that details these attacks so you can understand their complexity and impact.
>
> I have evidence that your computer system *hostname* has been compromised and is actively being used by someone for engaging in remote attacks against systems around the globe. Your system holds files associated with these attacks that are central to understanding them and trying to identify who is perpetrating them. I urge you to report this intrusion to the RCMP.
>
> I am requesting your permission to analyze the files contained in the compromised account I have identified and I advise that you make your own bit-image copy of the hard drive to preserve evidence that may still exist in unused file space.

The response I got was positive. The person verified who I was by looking at my Web site and said they appreciated the call and wanted to help. I was granted the permission I requested on the condition that I promised (1) to give them full details of how their system was compromised and how it was being used by the attackers, (2) never to disclose the name of their company and (3) not to publish any corporate or customer data I might encounter that was unrelated to the illegal activity of the attackers. I have to this day adhered to and will continue to adhere to all aspects of this promise.

The fruits of my analysis were the first detailed technical understanding of distributed denial of service attack tools. I produced details of how to detect these programs on infected hosts, how to detect them on the wire, and how to scan for them remotely. Instructions and guidance on how to locally scan one's own network were given, along with cautions about likely countermeasures that could result in false-negative checks. I circulated these analyses privately at first, trying to provide as much lead time as possible for law enforcement, the military, policymakers, and the security industry to prepare for what might come next. My analyses were used as discussion points for the first workshop ever organized and sponsored by CERT/CC [9]. Several

years later I co-authored the first book on Internet denial of service attacks with Jelena Mirkovic, Peter Reiher, and Sven Dietrich [6].

In the years following the release of the initial DDoS attack tool analyses, those publications were widely cited in academic research as among the first references on the subject. A new class of security products and services designed to detect and mitigate DDoS attacks was also created, many starting out by addressing the specifics detailed in these same analyses. However, the technical details of these attack tools alone were not the only insights into the complex nature of responding to large-scale distributed attacks. DDoS attacks, distributed tools, and *botnets* (as they are now popularly known) are multi-phase attacks that involve a complex arrangement of compromised resources spread across networks around the globe and organized into a coordinated attack network. There is much more to countering these threats than just detecting and blocking a flood of packets, and it demands similar automation of response actions, coordination between involved sites, and a deep knowledge of the attackers' tools and tactics.

Responding to Complex Computer Network Attacks

The issues of responding to increasingly sophisticated and complex computer network attacks that are illustrated here are not new. Just a few years earlier, President Clinton created the President's Commission on Critical Infrastructure Protection (PCCIP) to advise his administration on how to deal with the threats to critical infrastructures that were emerging from widespread Internet connectivity. The PCCIP produced a series of reports, entitled the *Legal Foundations Study* [8], which addressed such issues as difficulties in detecting computer crime, resource constraints, the existing legal landscape, (in)adequacy of existing criminal statutes, strains on federal law enforcement investigation and prosecution capabilities, international agreements on cooperation in cyber-investigations, and proposed new approaches to enhancing cyber-intrusion response. Two of the principal authors of the PCCIP reports published a law review article in which they call for a balanced public/private approach for responding to cybercrime that includes oversight mechanisms such as licensing and certification [7].

There have been other discussions of these topics in the private sector [1, 2]. Kenneth Einar Himma and I co-authored an article on what I am calling the *active response continuum* (ARC) in which some of the legal and ethical issues are raised [5]. We describe the issues in responding to large-scale coordinated attacks in the face of differences in skill level and capacity of the victim sites involved and other issues brought up by the PCCIP reports. I prefer the term "active response continuum" over "active defense," to stress the range from low to high of the capacity to respond, aggressiveness of actions, and risk of harm that must be balanced against intended benefit. It is very common for discussions involving people new to this topic, who have never engaged in coordinated and collaborative response to computer intrusions, to jump to simplistic self-defense analogies and call for the right to "hack-back" or "counter-strike." This is both naïve and very risky, as these are at one extreme end of the spectrum. Arguing simply whether or not one has a right to "hack-back" in self-defense misses more subtle, and less risky, alternatives. Similarly, all the various options along the continuum are not viewed in relation to effects on others who are simultaneously investigating and responding to the same widespread events, or those using the computer systems and networks involved in criminal activity.

Conclusion

Times have changed since 1999. The days of finding the source code, log files, exploit tools, etc., being cached in one place for months at a time are quite rare for the most advanced attacks. The sophisticated attacker knows better and does a much better job of operational security. This requires a more sophisticated response with more difficult challenges to overcome than in years past. Law enforcement is now far more coordinated internationally, more highly staffed, and more engaged with groups I will call *mitigation communities* whose good intentions and talents are applied to counter today's sophisticated cybercrime.

Good intention alone is not sufficient in deciding whether or not to take aggressive or risky actions in response to cybercrime, or in choosing which action to take. There are a host of unintended consequences that result from one's actions. It is important to have as much knowledge as possible about the behavior of attackers and the capabilities of their tools. It is hard enough to reverse engineer sophisticated malware, but finding an attacker's weakness and immediately leaping to disclose it or attack it is unwise in the extreme. It is equally hard to develop a sophisticated counterattack that considers the *effects* of any action one might take, in terms of benefit or advantage as well as risk or harm (e.g., privacy violation.)

This is not a situation where one group of white hats congregating in an online vetted community goes toe-to-toe with another group of black hats who congregate in their underground equivalent. There are millions of innocent third parties standing between and around us who just want to go about their daily business, using the Internet to enhance their lives. They don't want to be harmed by getting caught in the cyber cross-fire. The effect of a mistake that causes widespread harm to the general public could be significant, resulting in a public-opinion backlash or knee-jerk legislation that significantly sets back the efforts of defenders and puts attackers in an even stronger position. Solidifying the gap between government agencies and the private sector, or allowing researchers to perform crime-scene-altering experiments in an uncontrolled and uncoordinated manner, will similarly prevent a comprehensive cyber-response capability and further the damage currently being done to our nation.

My colleagues and I presented a poster at the 16th ACM Conference on Computer and Communications Security [3] that is based on a technical report [4] in which we call for a structured debate of the ethical issues surrounding computer security research activities that will guide decision-making in a more sophisticated and deliberate manner. This technical report contains over two dozen case studies from the research and computer security communities, going back many years. We provide an overview of various ethical codes, analysis methods, and related discussions from the information warfare and software engineering disciplines.

At the most basic level are issues of privacy that apply across a large percentage of computer security research. It is important that these fundamental issues be addressed, as they are increasingly raised in the context of academic research. Even if a research exception were to be added to the Wiretap Act or Stored Communications Act, the public would likely still want requirements for researchers to adhere to ethical principles that include the kind of minimization techniques described above. Having a legal exception allowing collection of data involving private communications does not mean privacy rights can then be ignored.

At the far end of the spectrum, where the subjects of research are criminal activities that have financial, political, business continuity, or national security implications, there is a need to look beyond privacy rights and harmonize research activities with law enforcement investigation and security- or network-operational requirements. In this area, we need standards and decision-making guidelines that allow deconfliction[1] of researchers' activities, consider alternative actions in terms of risk/benefit, harmonize security operations and research with law enforcement investigations, and balance roles and responsibilities.

We, as a community, urgently need to continue and expand the discussions about sophisticated and potentially aggressive countermeasures to cybercriminal activities in order to minimize harm and maximize benefit in the ongoing conflict occurring in cyberspace.

1. The term *deconfliction* comes from the military, where flight plans of fighters are coordinated to avoid interference during action. In this context, it means coordinating researchers' activities to avoid interfering with each other or interfering with law enforcement investigations, both of which can have negative effects such as over-counting, obscuring criminal actors, or sending law enforcement down dead-end paths that waste scarce resources and time.

REFERENCES

[1] Agora workshop moderators, First Agora Workshop on Active Defense, August 2001: http://staff.washington.edu/dittrich/arc/AGORA\%208JUN01.ppt.

[2] David Dittrich, Second Agora Workshop on Active Defense, September 2003, sponsored by Cisco Systems, Inc.: http://staff.washington.edu/dittrich/arc/AD-workshop-091203.pdf.

[3] David Dittrich, Michael Bailey, and Sven Dietrich, "Have We Crossed the Line? The Growing Ethical Debate in Modern Computer Security Research," November 2009. Poster presented at the 16th ACM Conference on Computer and Communication Security: http://www.sigsac.org/ccs/CCS2009/pd/abstract_22.pdf.

[4] David Dittrich, Michael Bailey, and Sven Dietrich, "Towards Community Standards for Ethical Behavior in Computer Security Research," Technical Report CS 2009-01 (April 20, 2009), Stevens Institute of Technology: http://staff.washington.edu/dittrich/papers/dbd2009tr1/.

[5] David Dittrich and Kenneth E. Himma, "Active Response to Computer Intrusions," Chapter 182 in Vol. III, *Handbook of Information Security* (Wiley, 2005): http://papers.ssrn.com/sol3/papers.cfm?abstract_id=790585.

[6] Jelena Mirkovic, Sven Dietrich, David Dittrich, and Peter Reiher, *Internet Denial of Service: Attack and Defense Mechanisms* (Prentice Hall PTR, 2004).

[7] Stevan D. Mitchell and Elizabeth A. Banker, "Private Intrusion Response," 1998: http://jolt.law.harvard.edu/articles/pdf/v11/11HarvJLTech699.pdf.

[8] President's Commission on Critical Infrastructure Protection, PCCIP –reports archive: http://cip.gmu.edu/clib/PCCIPReports.php.

[9] Several, Results of the Distributed-Systems Intruder Tools Workshop, CERT/CC, December 1999: http://www.cert.org/reports/dsit_workshop.pdf.