Lawful Hacking:
Using Existing Vulnerabilities for Wiretapping on the Internet[1]

Steven M. Bellovin[*], Matt Blaze[†], Sandy Clark[§], Susan Landau[‡]

DRAFT – August 18, 2013

For years, legal wiretapping was straightforward: the officer doing the intercept connected a tape recorder or the like to a single pair of wires. By the 1990s, though, the changing structure of telecommunications—there was no longer just "Ma Bell" to talk to—and new technologies such as ISDN and cellular telephony made executing a wiretap more complicated for law enforcement. Simple technologies would no longer suffice. In response, Congress passed the *Communications Assistance for Law Enforcement Act (CALEA)*[2], which mandated a standardized lawful intercept interface on all local phone switches. Technology has continued to progress, and in the face of new forms of communication—Skype, voice chat during multiplayer online games, many forms of instant messaging, etc.—law enforcement is again experiencing problems. The FBI has called this "Going Dark":[3] their loss of access to suspects' communication. According to news reports, they want changes to the wiretap laws to require a CALEA-like interface in Internet software.[4]

CALEA, though, has its own issues: it is complex software specifically intended to create a security hole—eavesdropping capability—in the already-complex environment of a phone switch. It has unfortunately made wiretapping easier *for everyone, not just law enforcement*. Congress failed to heed experts' warnings of the danger posed by this mandated vulnerability, but time has proven the experts right. The so-called "Athens Affair", where someone used the built-in lawful intercept mechanism to listen to the cell phone calls of high Greek officials, including the

---

[1] This paper was presented at the Privacy Legal Scholars Conference in June 2013; the authors have very much benefitted from the discussion and comments made there. We would especially like to thank Deirdre Mulligan, Marty Stansell-Gamm, and Judge Stephen Smith, as well as Daniel Immerman.

[*] Steven M. Bellovin is a professor of computer science at Columbia University.
[†] Matt Blaze is an associate professor of computer science at the University of Pennsylvania.
[§] Sandy Clark is a Ph.D. student in computer science at the University of Pennsylvania.
[‡] Susan Landau is a 2012 Guggenheim Fellow.

[2] Pub. L. No. 103-414, 108 Stat. 4279, codified at 47 USC 1001-1010.

[3] Valerie Caproni, General Counsel of the FBI, Statement Before the House Judiciary Committee, Subcommittee on Crime, Terrorism, and Homeland Security, February 17, 2011, available at https://www.fbi.gov/news/testimony/going-dark-lawful-electronic-surveillance-in-the-face-of-new-technologies

[4] Declan McCullagh, "'Dark' motive: FBI seeks signs of carrier roadblocks to surveillance", CNET News, Nov. 5, 2012, available at http://news.cnet.com/8301-13578_3-57545353-38/dark-motive-fbi-seeks-signs-of-carrier-roadblocks-to-surveillance/

Prime Minister,[5] is but one example. In an earlier work, we showed why extending CALEA to the Internet would create very serious problems, including the security problems it's visited on the phone system.[6]

In this paper, we explore the viability and implications of an alternative method for addressing law enforcements need to access communications: legalized hacking of target devices through *existing* vulnerabilities in end-user software and platforms. The FBI already uses this approach on a small scale; we expect that its use will increase, especially as centralized wiretapping capabilities become less viable.

Relying on vulnerabilities and hacking poses a large set of legal and policy questions, some practical and some normative. Among these are:

- Will it create disincentives to patching?
- Will there be a negative effect on innovation? (Lessons from the so-called "Crypto Wars" of the 1990s, and in particular the debate over export controls on cryptography, are instructive here.)
- Will law enforcement's participation in vulnerabilities purchasing skew the market?
- Do local and even state law enforcement agencies have the technical sophistication to develop and use exploits? If not, how should this be handled? A larger FBI role?
- Should law enforcement even be participating in a market where many of the sellers and other buyers are themselves criminals?
- What happens if these tools are captured and repurposed by miscreants?
- Should we sanction otherwise-illegal network activity to aid law enforcement?
- Is the probability of success from such an approach too low for it to be useful?

As we will show, though these issues are indeed challenging we regard them as, on balance, preferable to adding more complexity and insecurity to online systems.

---

[5] Vassilis Prevelakis and Diomidis Spinellis, "The Athens Affair", *IEEE Spectrum* 44:7, July 2007, pp. 26-33, available at http://spectrum.ieee.org/telecom/security/the-athens-affair/0

[6] Steven M. Bellovin, Matt Blaze, Sandy Clark, and Susan Landau, "Going Bright: Wiretapping without Weakening Communications Infrastructure", *IEEE Security & Privacy*, Jan/Feb 2013.

I. Introduction

For several years, the FBI has warned that newer communications technologies have hindered the bureau's ability to conduct electronic surveillance.[7] Valerie Caproni, General Counsel of the FBI, put it this way in Congressional testimony:[8]

> Methods of accessing communications networks have similarly grown in variety and complexity. Recent innovations in hand-held devices have changed the ways in which consumers access networks and network-based services. One result of this change is a transformation of communications services from a straight-forward relationship between a customer and a single CALEA-covered provider (e.g. customer to telephone company) to a complex environment in which a customer may use several access methods to maintain simultaneous interactions with multiple providers, some of whom may be based overseas or are otherwise outside the scope of CALEA.
>
> As a result, although the government may obtain a court order authorizing the collection of certain communications, it often serves that order on a provider who does not have an obligation under CALEA to be prepared to execute it.

The FBI's solution is "legislation that will assure that when we get the appropriate court order…companies…served…have the capability and the capacity to respond..."[9]

While on the one hand this request is predictable (given past precedent), it is rather remarkable given current national cybersecurity concerns in light of stark evidence of the significant harm caused by CALEA . The request to expand CALEA to IP-based communications places the needs of the Electronic Surveillance Unit above all else, above the security risks that arise when you build wiretapping capabilities into communications infrastructure and applications—above that of other government agencies who face increased risk from hackers and nation states who may exploit this new vulnerability, and above to the national need for innovation which drives economic prosperity. Rather than examining the issue in terms of social good—an examination that occurs each time a decision is made in prioritizing certain types of

---

[7] See, for example, "Going Dark: Lawful Electronic Surveillance in the Face of New Technologies", *Hearing before the Subcommittee on Crime, Terrorism, and Homeland Security of the Committee on the Judiciary, House of Representatives*, 112th Congress, February 17, 2011, Serial No. 112–59, available at http://judiciary.house.gov/hearings/printers/112th/112-59_64581.PDF.

[8] *Id*. at 14.

[9] *See* Statement for the Record, Robert S. Mueller, III, Director, Federal Bureau of Investigation, Committee on the Judiciary, United States Senate, Oversight of the Federal Bureau of Investigation, May 16, 2012, 112th Congress; *see also* Declan McCullagh, "FBI 'Looking at' Law Making Web Sites Wiretap-Ready, Director Says", *CNET News*, May 18, 2012, available at http://news.cnet.com/8301-1009_3-57437391-83/fbi-looking-at-law-making-web-sites-wiretap-ready-director-says/.

investigations (terrorism cases, drug cases, etc.), or in determining whether to conduct a particular investigation—the FBI has thrown down a gauntlet that ignores long-term national interest.

The FBI's preferred solution—"requiring that social-networking Web sites and providers of VoIP, instant messaging, and Web e-mail alter their code to ensure their products are wiretap-friendly"[10]—will create security risks in our already-fragile Internet infrastructure leaving the nation more vulnerable to espionage and our critical infrastructure more open to attack, and hinder innovation.[11]. The need for securing communications infrastructure is a national priority. By weakening communications infrastructure and applications, the FBI's proposal would mostly give aid to the enemy. Surely that is neither what the bureau intends nor what sound national priorities dictate.

The problem is technology. Over the course of the last three decades, we have moved from a circuit-switched centralized communications network—the Public Switched Telephone Network (PSTN)—run by a monopoly provider, to a circuit-switched centralized communications network run by multiple providers, to a Internet-Protocol (IP) based decentralized network run by thousands of providers. The first change, from the monopoly provider to multiple providers, gave rise to the need for the Communications Assistance for Law Enforcement Act (CALEA), simplifying law-enforcement's efforts to manage wiretaps with multiple, though relatively few, providers. But on certain occasions, such as the use of peer-to-peer communications or communications encrypted end-to-end, legally authorized wiretaps may be impeded. Even if law enforcement does not currently have a serious problem in conducting authorized wiretaps, with time it will. Thus there is a serious question of what is to be done. In appearing to request controls on peer-to-peer networks and on the use of encryption, [12] the FBI has floated highly flawed solutions.[13]

We propose another approach. Instead of building wiretapping capabilities into communications infrastructure and applications, government wiretappers can behave like the bad guys. That is, they can exploit the rich supply of security

---

[10] Declan McCullagh, "FBI: We Need Wiretap-Ready Web Sites—Now", *CNET News*, May 4, 2012, available at http://news.cnet.com/8301-1009_3-57428067-83/fbi-we-need-wiretap-ready-web-sites-now/.

[11] Indeed, sometimes the benefits are directly to the military. One NSA program, Commercial Solutions for Classified uses products from government research "layered" with private-sector products to produce communication tools with high security (Fred Roeper and Neal Ziring, "Building Robust Security Solutions Using Layering and Independence," RSA Conference 2012).

[12] Charlie Savage, "U.S. is Working to Ease Wiretaps on the Internet," NEW YORK TIMES (September 27, 2010) at A1.

[13] Six months after the New York Times reported the FBI was seeking additional capabilities for Internet wiretapping (Savage, *id.*), FBI General Counsel Valerie Caproni testified, "Congressman, the Administration is still working on what the solution would be, and we hope to have something that we can work with Congress on in the near future." See "Going Bright," *supra* note 6 at 40. As of this writing, no bill has been proposed.

vulnerabilities already existing in virtually every operating system and application to obtain access to communications of the targets of wiretap orders.[14]

We are not advocating the creation of *new* security holes,[15] but rather observing that exploiting *those that already exist* represents a viable – and significantly better – alternative to the FBI's proposals for mandating infrastructure insecurity. Put simply, the choice is between formalizing—and constraining—the ability of law enforcement to occasionally use existing security vulnerabilities—something we note the FBI and other law enforcement agencies already do when necessary without much public or legal scrutiny—or living with those vulnerabilities *and* intentionally and systematically creating a set of predictable new vulnerabilities that despite best efforts will be exploitable by *everyone*.

Using vulnerabilities to create exploits and wiretap targets, however, raises ethical issues. Once an exploit for a particular security vulnerability leaves the lab, it may be used for other purposes and cause great damage. Any proposal to use vulnerabilities to enable wiretaps must minimize such risks.

In previous work,[16] we discussed the technical feasibility of relying on the vulnerability approach; here we focus on the legal and policy issues posed by this approach. In particular, we examine the tension between the use of naturally occurring software vulnerabilities to legitimately aid law enforcement investigations and the abuse of the same vulnerabilities by criminals. We propose that law enforcement adopt a strict policy of immediately disclosing to the vendor any vulnerabilities that come to their attention as soon they are discovered. As we will discuss, such a policy allows law enforcement to fully support crime prevention, and—because of the natural lag of the software lifecycle—can still allow law enforcement to build a sufficiently rich toolkit to conduct investigations in practice.

The discussion in this paper is limited to use of vulnerabilities for *communications intercepts*, rather than generic "remote search." While the two concepts have much in common, including the use of vulnerabilities to achieve access, there are distinct differences in both the technical and legal aspects.

Section II sets the stage, first by discussing how CALEA fit into the communications environment of the time, and then its disjunction with newly evolving communication systems. We then examine the reasons and risks of extending CALEA to IP-based communications. The continued existence of vulnerabilities, fundamental to our proposal, is discussed in Section III. In section IV, we discuss their use for wiretapping. Using exploits to enable wiretapping raises a number of

[14] See Bellovin *et al.,* footnote 6, *supra*.
[15] That is indeed far from the case. Some of the authors have devoted much of our professional careers to preventing or coping with them and the problems they cause.
[16] See Bellovin *et al.,* footnote 6, *supra*.

troubling questions. As the Stuxnet cyberattack[17] amply demonstrates, even carefully tailored exploits can extend past their intended target. Law-enforcement's use of vulnerabilities therefore requires careful consideration of how to limit the proliferation, which we discuss in section V, and whether law enforcement use of vulnerabilities should influence norms around vulnerability reporting which we discuss in section VI. In section VII we discuss how to implement vulnerability reporting. We conclude our argument in section VIII.

[17] *See* Nicolas Falliere, Liam O Murchu, and Eric Chien, *W.32 Stuxnet Dossier*, Version 1.4, February 2011, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf. Stuxnet was apparently developed and launched by intelligence or cyberwarfare agencies; as such, its design is likely quite from a law enforcement exploit.

II. CALEA: The Change in Wiretap Architecture

The Communications Assistance for Law Enforcement Act (CALEA) was born of a certain time and certain place. It was a law created with the expectation of multiple, but relatively few, communications providers, and of a telephone network, while not exactly the world of the Public Switched Telephone Network (PSTN) of the 1950s-1980s, not substantively removed from it. It was anticipated that both the technical and business structure of communications networks would remain centralized. The changing telecommunications industry of multiple providers and digitized transport underlay the law, but the impact of the more fundamental changes that were percolating at the time of CALEA's passage—IP-based communications and enormous numbers of services—were not anticipated at the time. In this section, we discuss the problems that CALEA was intended to address and the problems it was not, briefly mention the security risks created by these solutions, and the patchwork of solutions that have emerged to cover IP-based voice communications. We conclude by describing the impact on wiretapping and CALEA of these changes.

A. History of CALEA

CALEA had its roots in the nascent switch to digital transport of voice over the phone network's local loops in the early 1990s. ISDN was touted as the next wave of telephony, since it could provide what was for the time very high speed data over a switched line.[18] For all ISDN's advantages, however, it was not possible to tap ISDN lines with the traditional "two alligator clips and a tape recorder". Furthermore, cellular telephony was growing rapidly; because the communication was wireless and mobile, cellular communications, too, could not be tapped that way. While specialized interception gear could have been developed, the FBI instead proposed what was originally known as the Digital Telephony Bill, a standardized interface for wiretaps. After considerable debate over the scope of coverage,[19] the current form of CALEA was passed, specifically excluding "information services".[20]

CALEA was intended to apply only to telephony. More precisely, CALEA was intended to apply to "local exchange service", i.e., local phone service but not long

---

[18] ISDN—Integrated Services Digital Network—was defined in M. Decina; E. Scace (May 1986). "CCITT Recommendations on the ISDN: A Review". *CCITT Red Book* **4** (3): 320–25 In its most common form, it provided so-called 2B+D service: two 64 kilobit/second "bearer" channels, and a 16 Kbps data channel for signaling, e.g., call setup and teardown. The two bearer channels could be combined into a single 128 Kbps link for pure data; this is more than twice as fast as any single-line analog phone modem can ever provide. For a variety of reasons, it never caught on in the United States as a common service.

[19] In 1992, the FBI proposed legislation that would have "allowed the technical design mandates on any provider of any electronic communications, including the Internet." (*See* Corrected Petition for Rehearing *En Banc*, Case 15-0504, Am. Council on Educ. v FCC, Court of Appeals for the D.C. Circuit, July 28, 2006 at 12, available at https://www.cdt.org/wiretap/calea/20060731calearehearing.pdf.) The proposal was "rejected out of hand". (*Id.)*

[20] 47 USC 1001(8)(C)(i)

distance carriers. Then-FBI Director Louis Freeh made clear in his 1994 Congressional testimony that the Internet was not covered:[21]

> Mr. Freeh. We are really talking about phone-to-phone conversations which travel over a telecommunications network in whole or part. That is the arena of criminal opportunity that we are discussing.
>
> Senator Pressler. What other portions of the information superhighway could people communicate with the new technology that there is not now a means of listening in or following?
>
> Mr. Freeh. From what I understand, and again, I am probably the worst person in this room to answer the question, communications between private computers, PC-PC communications, not utilizing a telecommunications common net, would be one vast arena, the Internet system, many of the private communications systems which are evolving. Those we are not going to be on by the design of this legislation.
>
> Senator Pressler. Are you seeking to be able to access those communications also in some other legislation?
>
> Mr. Freeh. No, we are not. We are satisfied with this bill. I think it delimits the most important area and also makes for the consensus, which I think it pretty much has at this point.

This consensus was reflected in the law, which defined a "telecommunications carrier" to include "a person or entity engaged in providing wire or electronic communication switching or transmission service to the extent that the Commission finds that such service is a replacement for a substantial portion of the local telephone exchange service and that it is in the public interest to deem such a person or entity to be a telecommunications carrier for purposes of this subchapter".[22]

More recently, CALEA coverage has been extended to "last mile" service: the link between a residence or business and its ISP. While controversial because of Freeh's testimony and the exclusion of information services in CALEA, the FCC and the courts have held that this class of link is not covered by the information services

---

[21] *See* Joint Hearings before the Subcommittee on Technology and the Law of the Senate Judiciary Committee and the Subcommittee on Civil and Constitutional Rights of the House Judiciary Committee on H.R. 4922 and S. 2375, "Digital Telephony and Law Enforcement Access to Advanced Telecommunications Technologies and Services," Testimony of Federal Bureau of lnvestigations Director Freeh, at 203 (August 11, 1994).

[22] *See* 47 U.S.C. §1001(8)(B)(ii).

exclusion.[23] More precisely, the FCC made that ruling; relying on *Chevron* deference,[24] the Court of Appeals upheld that the FCC's ruling.

This change to CALEA, though important, is of less concern to law enforcement than is the fate of the traditional telephone network. It is going away, and far faster than anyone had forecast. Already, more than 35% of American households do not have landline phone service; about 16% more who have landlines never or almost never receive calls on them.[25] Indeed, the working assumption in the Federal Communications Commission (FCC) is that the PSTN will effectively cease to exist by 2018.[26]

B. Wiretap Consequences of Splitting Services and Infrastructure

It might be tempting to say that the coming end of the PSTN vindicates the FBI's vision when it proposed CALEA. The actual situation, though, is far more complex; the decoupling of services from the physical link has destroyed the chokepoint at which CALEA could therefore be applied. This does not appear to have been anticipated at the time of CALEA's passage.

A paradigmatic case in which the decoupling presents serious wiretapping problems is when communication occurs through use of Voice over Internet Protocol (VoIP). As was shown by Bellovin *et al.*, a VoIP phone provider can be located far from its subscribers; indeed, it could be in another, possibly unfriendly, country. Furthermore, the "signaling path"—the set of links that carry the call setup messages—can differ from the "voice path", the links that carry the actual conversation.[27] (Tapping the last mile connection is likely fruitless, since VoIP connections are often encrypted.)

This is best explained by a diagram. Figure 1 shows a plausible setup for a VoIP call from Alice to Bob.[28] Alice's and Bob's phones are each connected to their own ISPs, Net 1 and Net 4. They each subscribe to their own VoIP provider, which are in turn connected to their own ISPs. The signaling messages—that is, the messages used to set up the call, indicate ringing, etc.—go from Alice's phone, through her ISP to VoIP

---

[23] Am. Council on Educ. v FCC (2006, App DC) 371 US App DC 307, 451 F3d 226, 25 ALR Fed 2d 717, reh den (2006, App DC) 2006 US App LEXIS 23061.

[24] *See* Chevron U.S.A., Inc. v. Natural Res. Def. Council, Inc., 467 U.S. 837, 104 S.Ct. 2778, 81 L.Ed.2d 694 (1984).

[25] Stephen J. Blumberg and Julian V. Luke, Wireless Substitution: Early Release of Estimates From the National Health Interview Survey, January-June 20102, available from http://www.cdc.gov/nchs/data/nhis/earlyrelease/wireless201212.pdf .

[26] Technical Advisory Council, Federal Communications Commission, Summary of Meeting, September 27th , 2011, available at http://transition.fcc.gov/oet/tac/tacdocs/tac-meeting-summary-9-27-11-final.docx.

[27] *See* Steven M. Bellovin, Matt Blaze, Ernest Brickell, Clinton Brooks, Vint Cerf, Whitfield Diffie, Susan Landau, Jon Peterson, and John Treichler. *Security implications of Applying the Communications Assistance to Law Enforcement Act to Voice over IP*, 2006, available at https://www.cs.columbia.edu/~smb/papers/CALEAVOIPreport.pdf, especially Figure 1 at 4.

[28] This figure is adapted from Bellovin et al., *id.*

Provider 1's ISP, to her phone company. It then contacts VoIP Provider 2, via its ISP; VoIP Provider 2 sends a message through Net 4 to Bob's phone. The actual voice path, however, goes directly from Net 1 to Net 4; neither Net 2, Net 3, nor the VoIP providers even carry the actual conversation. As noted, any or all of the messages may be encrypted.

In this setup, where can a tap be placed? On any of the ISPs? Law enforcement has no a priori information where Alice and Bob will be—their current IP addresses—prior to their setting up a call, so law enforcement cannot serve the IPSs with a wiretap order. To make matters worse, the ISPs have nothing to do with the VoIP call, nor can they read the encrypted traffic. At one of the VoIP providers? They do not see the voice traffic. And, of course, they may be in a different jurisdiction (for example, Skype was originally hosted in Luxembourg). This is a scenario that has no points amenable to a CALEA-like solution.

Other services are more complex still. Consider the new phone service being offered by Republic Wireless, which uses a combination of IP and PSTN networks to call. The service is intended to operate primarily over WiFi networks and the Internet; however, it can switch to Sprint's 3G cellular network as needed.[29] Where could a CALEA tap be placed? Certainly, a tap could be placed on the Internet-facing side of Republic's facilities,[30] but that would miss Sprint calls. Conversely, there could be one on Sprint's network, but that would miss calls made via VoIP. It is of course possible to place taps on both networks, but the protocols are very different and special code would be needed to hand off not just the call but also the information necessary to carry out the tap, since the ordinary signaling mechanisms would not be used.[31] Pen register taps would be even more involved.

Apart from reasonably straightforward (though structurally different) PSTN replacements, a large variety of other communications schemes have gained popularity. Email and text messages are the obvious replacements, though even these pose challenges for law enforcement due to issues of jurisdiction and lack of real-time access to content. Skype is perhaps the most extreme case. Its architecture, which the FCC report calls "over the top,"[32] has no central switches. Even apart from questions of jurisdiction, there are *no* locations where a CALEA-

---

[29] Walter Mossberg, "For $19, an Unlimited Phone Plan, Some Flaws", *Wall Street Journal*, February 19, 2013, available at http://allthingsd.com/20130219/for-19-an-unlimited-phone-plan-some-flaws/.

[30] Tapping the customer's own Internet connection would not suffice, since the customer is likely to use multiple WiFi networks that such a tap would miss. Also note that while Republic Wireless is a U.S. company, there is no reason why a similar service could not be offered by an offshore company over which U.S. courts have no jurisdiction.

[31] As of this writing, the Republic Wireless network cannot do handoffs of an in-progress call from a WiFi network to Sprint or vice-versa. According to Mossberg, *supra* footnote 29, that feature is planned for the near future.

[32] FCC Critical Legacy Transition Working Group, "Sun-setting the PSTN" at 3, September 27, 2011, available at http://transition.fcc.gov/oet/tac/tacdocs/meeting92711/Sun-Setting_the_PSTN_Paper_V03.docx at 1.

style interface could be provided. Everything is done peer-to-peer; ordinary Skype users forward signaling traffic for each other.[33] Because of this, there are no trusted elements that could serve as wiretap nodes at least for pen register orders; furthermore, calls are always encrypted end-to-end.[34]

It is useful to contrast the Skype architecture with the conventional client-server architecture shown in Figure 1. In that configuration, the VoIP providers run servers to which the individual phones—the clients—connect. These are architecturally different roles; when setting up calls, phones talk only to their associated servers; the servers talk to the clients but also to each other. It is not possible for Alice's phone to contact VoIP Provider 2 directly; they have no business relationship, and therefore cannot set up a direct network link.[35] In a peer-to-peer setup such as is used by Skype, there are *no* servers, i.e., no architecturally distinguished roles.[36] Rather, *every* computer or device running a Skype client can participate in the signaling. Alice's phone (somehow) finds another Skype client and asks it to connect to Bob. This node finds another, which finds another, etc., until

[33] It is unclear how true this still is. Skype has long had the concept of a "supernode", a well-connected computer that carries considerably more traffic. Of late, Microsoft—the current owner of Skype—has been deploying dedicated supernodes in its own data centers; *see* Dan Goodin, "Skype replaces P2P supernodes with Linux boxes hosted by Microsoft (updated)", *Ars Technica*, May 1, 2012, available at http://arstechnica.com/business/2012/05/skype-replaces-p2p-supernodes-with-linux-boxes-hosted-by-microsoft/. There have been some allegations that the replacement was done precisely to permit surveillance (*see*, e.g., John D. Scudder, "Can Skype 'wiretap' video calls?", CNN, July 24, 2012, available at http://www.cnn.com/2012/07/24/tech/web/skype-surveillance); these are disputed by Mary Branscombe, "Forget the conspiracy theories: Skype's supernodes belong in the cloud", ZDNet, July 27, 2012, available at http://www.zdnet.com/forget-the-conspiracy-theories-skypes-supernodes-belong-in-the-cloud-7000001720/. The one-time principal architect of Skype, Matthew Kaufman, has explained that the change was done to accommodate the switch from always-on desktops to battery-powered mobile devices; *see* Zack Whittaker, "Skype ditched peer-to-peer supernodes for scalability, not surveillance", ZDnet, June 24, 2013, available at http://www.zdnet.com/skype-ditched-peer-to-peer-supernodes-for-scalability-not-surveillance-7000017215/. Microsoft has applied for a patent on mechanisms for eavesdropping on VoIP networks; some commentators have alleged that this technology will be incorporated into Skype. *See, e.g.,* Jaikumar Vijayan, "Microsoft seeks patent for spy tech for Skype", *Computerworld*, June 28, 2011, available at https://www.computerworld.com/s/article/9218002/Microsoft_seeks_patent_for_spy_tech_for_Skype.

[34] For a good, albeit dated—and paid for by Skype—review of the encryption architecture, *see* Tom Berson, "Skype Security Evaluation", October 18, 2005, available at http://www.anagram.com/berson/abskyeval.html.

[35] This is not a technical limitation per se; however, VoIP Provider 2 knows nothing of Alice's phone, and hence is not willing to believe any assertions about its phone number, the person who uses it, etc. More importantly, because of the lack of a business relationship it will not provide service to Alice's phone since it will not be paid for its efforts.

[36] This is not strictly true. The Skype servers, however, are involved only in registering new users and providing them with cryptographic credentials. They are not involved in call setup, let alone being in the voice path.

Bob's phone is located.[37] At point, Alice's and Bob's phones exchange signaling messages and set up the voice path. This voice path is in principle direct, though for various reasons including the existence of firewalls other Skype nodes may relay the (encrypted) voice packets. The lack of central servers, other than for user registration and enhanced services such as calling out to PSTN numbers, dramatically cut the operational costs and allowed Skype to offer free or extremely cheap phone calls.[38]

All that said, one of the Snowden revelations is that the NSA can indeed intercept Skype calls.[39] No technical details have been disclosed; all we know is that the NSA can intercept audio and video, with complete metadata. It remains unclear if the solution is one that is usable by ordinary law enforcement, or if it relies on techniques (such as advanced cryptanalysis) that are peculiar to the intelligence community.[40]

Text messaging has also changed. Originally, it was a simple protocol for mobile phones. Recently a number of variant implementations that either provide a better experience in some fashion (Apple's iMessage, for example, will send copies of inbound messages to all of a user's devices; these can include tablets and Mac computers as well as phones), or can provide phone-like text messaging have been introduced for non-phone devices such as tablets.[41]

Non-traditional text messaging applications have already proven problematic. According to one report, attributed to a Drug Enforcement Administration memo,[42] the encryption used by Apple's iMessage has already stymied wiretap orders.[43]

---

[37] How the call eventually reaches Bob's phone is a rather complex technical matter, and not relevant here. Let it suffice to say that Skype nodes regularly exchange enough navigational messages that it can be done.

[38] The lack of central servers was a deliberate architectural choice, designed to evade legal constraints. Architecturally, it was based on the Kazaa file-sharing network; it in turn was designed to operate without vulnerable nodes that could be targeted by copyright infringement lawsuits. That notwithstanding, the operator, Sharman Networks—which profited from ads displayed by the Kazaa software—eventually shut down the service to settle several suits.

[39] *See* Glenn Greenwald, Ewen MacAskill, Laura Poitras, Spencer Ackerman and Dominic Rush, "How Microsoft handed the NSA access to encrypted messages", *The Guardian*, July 11, 2013, available at http://www.guardian.co.uk/world/2013/jul/11/microsoft-nsa-collaboration-user-data/print.

[40] Microsoft has claimed that in 2012 it has produced "no content" to law enforcement from Skype calls. *See* Brad Smith, "Microsoft Releases 2012 Law Enforcement Requests Report", March 21, 2013, available at https://blogs.technet.com/b/microsoft_on_the_issues/archive/2013/03/21/microsoft-releases-2012-law-enforcement-requests-report.aspx; *also see* the linked-to reports at https://www.microsoft.com/about/corporatecitizenship/en-us/reporting/transparency/.

[41] There are many such applications available. http://ipod.about.com/od/iphoneappsreviews/tp/4-Ways-To-Text-With-The-Ipod-Touch.htm gives one list, but new ones are constantly appearing.

[42] *See* Declan McCullagh, "Apple's iMessage Encryption Trips up Feds' Surveillance", *CNET News,* April 4, 2013, available at http://news.cnet.com/8301-13578_3-57577887-38/apples-imessage-encryption-trips-up-feds-surveillance/.

[43] Since the design of the protocol has not been published, it has not been possible for outside experts to assess this claim. Some have asserted, based on certain externally-visible characteristics (e.g., the ability to do a password reset and still see old messages), that the messages must be stored

There are even instant messaging applications designed not just to encrypt traffic, but to provide "repudiation", the ability to deny that you sent certain traffic.[44]

Beyond that, many non-obvious communications mechanisms can serve for direct communications as well. In one well-known case, General David Petraeus and Paula Broadwell apparently sent each other messages by creating and saving draft email messages in a shared Gmail account.[45] Many multiplayer games include text or even real-time voice communications between players; while nominally intended to lend realism to the game—soldiers in the same unit in action games can talk to each other; fighters on opposing sides can yell challenges or insults—such applications can also be used for surreptitious communications. Given that the Internet *is* a communications network, this raises the specter that *all* programs can be considered communications systems.

C. New Technologies: Going Dark or Going Bright?

Collectively, the changes in telephony, the rise of new communications technology, and (to some extent) the increasing use of encryption have been called the "Going Dark" problem: law enforcement has been unable to keep up with these changes and is losing access to criminals' communications. Technology works both ways, however; others have claimed rightly that modern developments have actually *increased* the practical ability of law enforcement,[46] perhaps even without the need for probable cause-based warrants. How serious is the Going Dark problem? How has the balance changed?

A firm, quantitative answer to the former question is probably not possible. We cannot say how many tap attempts have failed because law enforcement has said that it will not seek wiretap orders for calls it cannot intercept. Furthermore, the

---

unencrypted on Apple's servers; *see*, for example, Julian Sanchez, "Untappable Apple or DEA Disinformation?", April 4, 2013, available at http://www.cato.org/blog/untappable-apple-or-dea-disinformation. If that is true, a court order under the Stored Communications Act, 18 USC 2071 *et seq.,* would provide law enforcement with the content, albeit perhaps not in real-time.

[44] *See* Nikita Borisov, Ian Goldberg, and Eric Brewer. "Off-the-record communication, or, why not to use PGP." *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*. ACM, 2004. Note that "repudiation" (derived from its more cryptographic common counterpart, "nonrepudiation") is used here as a computer scientist would use it. It refers to certain cryptographic properties: in terms of the encryption mechanisms used, it is not possible to show mathematically that a given person has sent certain messages. Concepts that a lawyer might rely on, *e.g.*, circumstantial evidence or eyewitness testimony to the contrary, are not part of this mathematical model.

[45] *See* "Here's the E-Mail Trick Petraeus and Broadwell Used to Communicate", *Washington Post*, November 12, 2012, available at http://www.washingtonpost.com/blogs/worldviews/wp/2012/11/12/heres-the-e-mail-trick-petraeus-and-broadwell-used-to-communicate/.

[46] The claim is that the existence and availability of other information, such as location data, commercial data dossiers, and readily available contact information has given law enforcement for more than technology has taken away. *See, e.g.,* Peter Swire and Ahmad, Kenesa, Encryption and Globalization (November 16, 2011). Columbia Science and Technology Law Review, Vol. 23, 2012; Ohio State Public Law Working Paper No. 157. Available at SSRN: http://ssrn.com/abstract=1960602 or http://dx.doi.org/10.2139/ssrn.1960602

situation is not static; both criminals and police adapt their tactics in response to the other side's abilities and tactics. Consider cellular telephony. Under the *Omnibus Crime Control and Safe Streets Act*, the Administrative Office of the U.S. Courts (AO) reports annually on all Title III wiretaps, including the offense under investigation, who the prosecuting attorney was, who the authorizing judge was, how many intercepts, how many incriminating intercepts, the cost of the surveillance, etc.[47] In 2000, the report began listing how many wiretaps were of portable devices; there were 719 out of a total 1190 Title III wiretaps.[48] By 2009 it was 2276 out of 2376, or 96%.[49] This, of course, mirrors the trends of society as a whole; as noted, a majority of Americans rely on mobile phones for most of their incoming calls.[50]

That last fact provides a partial answer to the question of gaining and losing capabilities as a result of modern communication systems. Because they are far more likely to capture the target's conversations—rather than a spouse or business associate's—mobile phone taps are more valuable than wireline taps. Furthermore, mobile data can include information on where someone is. This means that 96% of wiretapped communications provide law enforcement with extremely valuable location information. The same is true of many Internet connections, whether fixed or mobile.[51] In other words, the prevalence of immediate communications—texting, cellular calls, and the like—and centralized services—Gmail, Facebook—has vastly simplified law-enforcement's ability to both track suspects and access their communications.

Another way to assess the overall risk is to look at the net effect of prior threats: how much has the police ability to monitor communications affected by prior technological changes, such as encryption? The issue has long been a concern, so much so that in 1993, the government announced the so-called "Clipper Chip", an encryption device designed so that the government could read otherwise-encrypted traffic.[52] The AO wiretap reports now include data on how often encryption has

---

[47] Administrative Office of the U.S. Courts, Wiretap Reports, http://www.uscourts.gov/Statistics/WiretapReports/WiretapReports_Archive.aspx [last viewed February 25, 2013].

[48] Administrative Office of the U.S. Courts, Wiretap Report 2000, Table 7.

[49] Administrative Office of the U.S. Courts, Wiretap Report 2009, Table 7.

[50] *See* Stephen J. Blumberg and Julian V. Luke, *Wireless Substitution: Early Release of Estimates From the National Health Interview Survey, January–June 2012,* December 2012, available at http://www.cdc.gov/nchs/data/nhis/earlyrelease/wireless201212.pdf.

[51] A technology known as "IP geolocation" can be used to determine where an Internet user is. It is frequently used to enforce geographic restrictions on access to content; *see, e.g.,* http://mlb.mlb.com/mlb/official_info/about_mlb_com/terms_of_use.jsp#4I. While many IP geolocation services provide fairly coarse resolution, some companies have done far better by combining IP address information with outside data such as search queries, purchase delivery records, etc.

[52] *See* John Markoff, "Electronics Plan Aims to Balance Government Access With Privacy", *New York* Times, April 16, 1993, available at http://www.nytimes.com/1993/04/16/us/electronics-plan-aims-to-balance-government-access-with-privacy.html. *See also* Matt Blaze, "Notes on key escrow meeting with NSA", *Risks Digest* 15:48, February 8, 1994, at http://catless.ncl.ac.uk/Risks/15.48.html#subj1: "They indicated that the thinking was not that criminals would use key escrowed crypto, but that

been encountered.[53] The data are interesting. The total between 2001-2011 is 87; of these, only one of these was the subject of a federal wiretap order.[54] The AO noted that law enforcement was able to decrypt all the wiretapped communications.

There is not a lack of communications products that provide end-to-end encryption; RIM's Blackberries, Skype, etc. While there are there are smart criminals who do use—and even build—their own encrypted communications networks,[55] the AO numbers demonstrate that criminals against whom Title III wiretaps are used are typically not in that category. Instead they tend to simple solutions: Commercial Off-The-Shelf (COTS) equipment and communications in the cloud (Gmail, Facebook). Few use the peer-to-peer communication channels that are problematic for law-enforcement wiretaps. The implication for law-enforcement use of vulnerabilities for performing Title III wiretaps is simple: law enforcement will not need to go that route very often.

Put another way, criminals are like other people: few use cutting edge or experimental devices to communicate. Instead they stick with COTS. If nothing else, COTS products are generally easier to use and work better, a definite advantage. Furthermore, understanding of the fine details of new technologies such as encryption is limited. The distinction between end-to-end encryption and client-to-server encryption is lost on most people, criminals included; similarly, the question of whether the encryption is going to the right party is often not even asked. Good

---

they should not field a system that criminals could easily use against them. The existence of key escrow would deter them from using crypto in the first place. The FBI representative said that they expect to catch 'only the stupid criminals' through the escrow system."

[53] As a result of Public Law 106-197, since 2000 the AO has reported the annual total of state and federal wiretap orders encountering encryption.

[54] There were (Administrative Office of the U.S. Courts, Wiretap Report 2001, at 5), an additional 18 for 2001 reported in 2002 as well as 16 for 2002 (Administrative Office of the U.S. Courts, Wiretap Report 2002, at 5), one in 2003 (Administrative Office of the U.S. Courts, Wiretap Report 2003, at 5), two in 2004 (Administrative Office of the U.S. Courts, Wiretap Report 20014, at 5), 13 in 2005 (Administrative Office of the U.S. Courts, Wiretap Report 2001, at 5), none in 2006 (Administrative Office of the U.S. Courts, Wiretap Report 2006, at 5), none in 2007 (Administrative Office of the U.S. Courts, Wiretap Report 2007, at 5), two in 2008 (Administrative Office of the U.S. Courts, Wiretap Report 2008, at 5), one in 2009 (Administrative Office of the U.S. Courts, Wiretap Report 2009, at 9), six in 2010 (Administrative Office of the U.S. Courts, Wiretap Report 2010, at 9), and twelve in 2011 (Administrative Office of the U.S. Courts, Wiretap Report 2011, at 8-9); all but one these were state wiretaps (the one federal case occurred in 2004).

[55] Spencer Ackerman, "Radio Zeta: How Mexico's Drug Cartels Stay Networked," WIRED, December 27, 2011, http://www.wired.com/dangerroom/2011/12/cartel-radio-mexico/ (last viewed February 18, 2013].

software usually performs the proper checks,[56] but even production code has had serious errors.[57]

From this perspective, the most serious threat to legally authorized wiretapping is exemplified by the Skype architecture. Virtually all email services feature (at most) encryption from the client to the mail server; the messages reside in plaintext on the mail providers' disks.[58] By contrast Skype provides transparent end-to-end encryption from the sender to the receiver; there is no middle man that sees the communication "in the clear." Skype is gaining an increasing share of the international telephony market.[59] But even with Skype, though, investigators are not shut out completely; as it turns out, and even without reading the encrypted text, Skype leaks the IP addresses of its users.[60] This provides the equivalent of pen register data and often location information as well.[61]

Technological changes will also play a role. However, it is difficult at this point to make confident predictions about the future direction of technology. The two popular trends, cloud computing and peer-to-peer networking, have opposite effects on law enforcement's ability to monitor communications.

Cloud computing moves more and more storage and computation to distant, network-connected servers. Today's email scenario is an old but telling example: all of a target's email passes through easily monitored remote servers. These servers tend to have stringent backup regimens and log everything, out of operational necessity. Even deletion operations are less than permanent;[62] preservation of data

---

[56] The best example is how web browsers use encryption. When a browser connects via HTTPS, the web server sends its "certificate" to the browser. A full explanation of certificates is out of scope here; what is important is that they contain a cryptographically protected association between the web site's name and a unique cryptographic key. Browsers verify that the name of the web site contacted actually appears in the certificate; thus, you won't end up with an encrypted connection to EvilHackerDudez.org when you are trying to log in to your bank.

[57] *See, e.g.,* Sascha Fahl, Marian Harbach, Thomas Muders, Matthew Smith, Lars Baumgärtner, Bernd Freisleben, "Why Eve and Mallory Love Android: An Analysis of Android SSL (In)Security," *Proc. ACM CCS 2012.*

[58] Although probably technically feasible (though difficult, given the need to comply with industry standards), it is highly unlikely that providers such as Google's Gmail and Microsoft's Hotmail will switch to end-to-end encryption. There is little consumer demand, it is difficult, and Google at least relies on being able to scan messages in order to display appropriate ads. It cannot do so if the messages are encrypted.

[59] *See* "The bell tolls for telcos?", *Telegeography*, February 15, 2013, available at http://www.telegeography.com/products/commsupdate/articles/2013/02/13/the-bell-tolls-for-telcos/.

[60] *See* Joel Schectman, "Skype Knew of Security Flaw Since November 2010, Researchers say", *Wall Street Journal*, May 1, 2012, available at http://blogs.wsj.com/cio/2012/05/01/skype-knew-of-security-flaw-since-november-2010-researchers-say/.

[61] *See* Footnote 51, *supra*.

[62] *See, e.g.,* Section 4.3 of the Microsoft Services Agreement: "please note that while content you have deleted or that is associated with a closed account may not be accessible to you, it may still remain on our systems for a period of time." Available at http://windows.microsoft.com/en-us/windows-

is paramount, even under extreme circumstances.[63] In theory, cloud storage could be encrypted; in practice, because of users' desire to be able to search their email messages and the lack of customer demand, there has been little, if any, real-world deployment.[64] In fact, in order to better serve ads, the Facebook and Google business models rely on the cloud data being unencrypted.

The other trend, peer-to-peer, is decentralized, with no convenient points for wiretaps or content monitoring. Rather than clients and servers, computers, phones, and other gadgets talk to each other. Why, for example, must email from Alice to Bob flow from her phone to her ISP's outbound mail server to Bob's ISP's inbound mail server to Bob's computer? Indeed, in some scenarios even ISPs disappear; in a technology known as "mesh networking"[65] computers ask other peer computers to relay their traffic. One very active area of development for mesh networks is car-to-car traffic for automotive safety and congestion control;[66] this could end up denying law enforcement access to location data from cellular networks.

In a cloud world, monitoring will be easier, in a peer-to-peer world, harder. It is quite possible that both trends will continue, with different applications and different markets opting for one solution over the other.

D. The Difficulties of CALEA II

CALEA II, the extension of CALEA to cover all communications applications, poses three serious problems: it hinders innovation by restricting communications application developers to certain topological and trust models, it imposes a financial tax on software, and it creates security holes (and hence increases the risk of computer crime, cyberepionage, and cyberterrorism,). This last point is perhaps the least-mentioned in the debate. Arguably, though, it is the most important, since it is

---

live/microsoft-services-agreement. Other providers have similar provisions, out of technical necessity.

[63] In 2010, a software problem caused thousands of Microsoft's Hotmail users to lose their entire mailboxes. Although it took several days, Microsoft was able to retrieve and restore the data from backup media. *See* Sebatian Anthony, "Hotmail users lose entire email inboxes, Microsoft restores them 5 days later", *Huffpost Tech Switched*, January 3, 2011, http://downloadsquad.switched.com/2011/01/03/hotmail-users-lose-entire-email-inboxes-microsoft-restores-them/.

[64] Encrypted storage and encrypted search are active research areas. However, except under special circumstances (e.g., a structured database, as opposed to email), encrypted remote search remains much more expensive than the plaintext equivalent and is likely to remain that way.

[65] *See, e.g.,* Rafe Needleman, "Unbreakable: Mesh networks are in your smartphone's future", *CNET*, July 13, 2013, available at http://www.cnet.com/8301-30976_1-57471447-10348864/unbreakable-mesh-networks-are-in-your-smartphones-future/.

[66] *See* Jon Brodkin, "Wireless mesh networks at 65MPH—linking cars to prevent crashes", *Ars Technica*, January 10, 2013, http://arstechnica.com/information-technology/2013/01/wireless-mesh-networks-at-65mph-linking-cars-to-prevent-crashes/.

the one not addressable by perfect (or at least very, very good) software development practices and/or reuse of standard CALEA compliance libraries.

An implicit assumption behind CALEA-style laws is that there is a "good" place where intercepts can take place. Such a place would be run by trustworthy people who are not implicated in the investigation,[67] and where the tap cannot be detected. More or less of necessity, this translates to relying on a centralized facility, preferably one run by a large, accountable company. This worked well for the telephone taps, where all lines were connected to a phone switch run by a conventional phone company. By contrast, consider a Skype-like architecture with transmissions over a mesh network. There are *no* large companies involved in either the call setup or data paths; rather, both use effectively random links. Furthermore, there may be little or no logging present; not only is the path used for one call probably not the path used for another, there will be no logs to show what paths were used. This means little or no accountability for any parties who leak information, and no assurance whatsoever that any will be able to complete the tap.

The fact that a peer-to-peer service is not facilities-based—that is, it does not rely on provider-owned equipment—also means there may be no parties to whom the law applies. For example, CALEA requires that "a telecommunications carrier shall ensure that its equipment, facilities, or services… enable the government… to intercept… all wire and electronic communications carried by the carrier… concurrently with their transmission to or from the subscriber's equipment."[68] There are, within the definitions of the statute, no carriers in some peer-to-peer architectures: "The term "telecommunications carrier" means a person or entity engaged in the transmission or switching of wire or electronic communications as a common carrier for hire"[69] or "a person or entity engaged in providing wire or electronic communication switching or transmission service to the extent that the Commission finds that such service is a replacement for a substantial portion of the local telephone exchange service."[70] In a peer-to-peer network, there is no such thing as "local" service; a "peer" need not be geographically close to any of the parties. Similarly, there may be no "manufacturer of telecommunications transmission or switching equipment" who can be compelled to "make available to the telecommunications carriers using its equipment, facilities, or services such features or modifications as are necessary to permit such carriers to comply with the capability requirements";[71] they, the peer nodes, and any commercial entities

---

[67] Per 18 U.S.C. §2511, "No provider of wire or electronic communication service, officer, employee, or agent thereof … shall disclose the existence of any interception or surveillance or the device used to accomplish the interception or surveillance with respect to which the person has been furnished a court order or certification under this chapter… Any such disclosure, shall render such person liable for the civil damages provided for in section 2520." Damages after the fact are one thing, but law enforcement would much rather the tap were not disclosed in the first place.
[68] 18 U.S.C. §1002(a).
[69] 18 U.S.C. §1001(8)(A).
[70] 18 U.S.C. §1001(8)(B)(ii)
[71] 18 U.S.C. §1005(b)

involved in the service operation (and there need not be any such) may be located outside of U.S. jurisdiction.[72]

To sum up, the laws assume a trustable, disinterested intermediary within the courts' jurisdiction. But as the net moves towards a more decentralized architecture, such third parties simply do not exist. Current technological trends pose a serious (and probably insurmountable) philosophical challenge to CALEA-style laws.

If CALEA were to be extended to cover IP-based communications, the law would have to specify which part of the service is responsible for supplying wiretap capability. As noted earlier, peer-to-peer networking is one plausible path for the technical future. Imposing requirements that effectively block this approach would have a very serious effect on innovation. Peer-to-peer communications have enabled some important applications such as BitTorrent, used by NASA for sharing satellite images, by various computer companies for sharing large files (e.g., open source operating systems), by gaming companies for sharing updates, and even by content providers such as CBS and Warner Bros. for delivering programming.[73]

There is a second burden on innovation: the extra cost, both in development effort and development time, to include wiretap interfaces in early versions of software is prohibitive. CALEA compliance, at first blush, seems simple: "all" that is wanted is dialed and dialing phone numbers, and voice. At that level, it is simple; nevertheless, the document defining the standard interface to a CALEA-compatible switch is more than 200 pages long.[74] Imagine, then, the standards necessary to cover interception of email, web pages, social networking status updates, instant messaging (for which there are several incompatible protocols), images, video downloads, video calls, video conference calls, file transfer layered on top of any of these, very many different sorts of games that have voice or instant messaging functions included, and more. It is simply not a feasible approach. Nor are these improbable uses of the Internet; all of them are used very regularly by millions of people.

Applying CALEA to Internet applications and infrastructure will be a "tax" on software developers. The much lower barriers to entry provided by the open architecture of the Internet to entry have bred many startups. These are small and agile; they're often the proverbial "two guys in a garage". Many will fail; even the eventual successes often start slowly. That said, they are essential to the Internet's

---

[72] A service without any operators does not imply that no one profits. The original KaZaA filesharing service was ad-supported (*see* https://en.wikipedia.org/wiki/Kazaa). It is unreasonable and probably infeasible to impose wiretap requirements on advertisers; the chain of indirection from the software developer to the advertisers is too long and tenuous; *see, e.g.,* Kate Kaye, "The Purchase-to-Ad Data Trail: From Your Wallet to the World", *Ad Age*, March 18, 2013, available at http://adage.com/article/dataworks/purchase-targeted-ads-data-s/240300/.

[73] *See, e.g.,* Brad King, "Warner Bros. to Distribute Films Using Bit Torrent", *MIT Technology Review*, May 9, 2006, available at http://www.technologyreview.com/view/405794/warner-bros-to-distribute-films-using-bit-torrent/.

[74] *See Lawfully Authorized Electronic Surveillance,* J-STD-025, Rev. A, 2000, http://cryptome.org/espy/TR45-jstd025a.pdf.

success. Skype started small; it is, as noted, now one of the largest international phone carriers.[75] For that matter, one need look no farther than Facebook (started by an undergraduate in his dorm room) for an example. Indeed, the Web began as an information distribution system at a European physics lab. It is hard to say at what point an experiment has become large enough to be a "service" worthy of being wiretap-friendly; it is clear, though, that requiring such functionality to be built in from the start is a non-trivial economic burden and a brake on innovation. By contrast, the PSTN is primarily composed of large, established companies who buy essentially all of their equipment from other large, established companies.[76]

The most serious problem with CALEA, though, is that it has created a new class of vulnerabilities. A wiretap interface is, by definition, a security hole, in that it allows an outside party to listen to what is normally a private conversation. It is supposed to be controlled, in that only authorized parties should have access. Restricting access to such facilities is far more difficult than it would appear; the history of such mechanisms is not encouraging.

The risks are not theoretical. In the 2004-2005 "The Athens Affair",[77] new code that used the lawful intercept mechanisms to eavesdrop on about 100 mobile phones, up to and including the Prime Minister's, was injected into the phone switch. In a similar, though less publicized, incident in Italy, between 1996-2006, about 6,000 people were the target of improper wiretaps, apparently due to corrupt insiders who sought financial gain. Again, the lawful intercept mechanism was abused.[78]

The U.S. is at risk, too. Phone switches are already large, extremely complex computer systems; as such, they are *inherently* at risk. An NSA evaluation of CALEA-compliant phone switches found vulnerabilities in every single one evaluated.[79] It is not known publicly if any American phone switches have been penetrated; however,

---

[75] *See* footnote 59, *supra.*

[76] Even for such companies, the expense of adding CALEA facilities was non-trivial. The statute (18 U.S.C. §1007-1008) authorized $500 million "to pay telecommunications carriers for all reasonable costs directly associated with the modifications performed by carriers in connection with equipment, facilities, and services installed or deployed on or before January 1, 1995, to establish the capabilities necessary to comply with section 1002 of this title." The funding was approved in the Omnibus Consolidated Appropriations Act, and it provided for funding through a combination of money supplied by various intelligence agencies, as well as $60 million in direct funding. An additional $12 million was provided through unspent Department of Justice funds. More than 95% of the money was actually spent; about $40 million was rescinded by Congress in 2007. *See* "Implementation of the Communications Assistance for Law Enforcement Act by the Federal Bureau of Investigation", Audit Report 08-20, U.S. Department of Justice, Audit Division, Redacted for public release, March 2008, available at http://www.justice.gov/oig/reports/FBI/a0820/final.pdf.

[77] *See* Vassilis Prevelakis and Diomidis Spinellis, "The Athens Affair", *IEEE Spectrum* 44:7, July 2007, pp. 26-33, available at http://spectrum.ieee.org/telecom/security/the-athens-affair/0.

[78] *See* Piero Colaprico, "Da Telecom dossier sui Ds Mancini parla dei politici," *La Repubblica*, January 26, 2007.

[79] *See* Susan Landau, "The Large Immortal Machine and the Ticking Time Bomb," J. Telecommunications and High Technology Law, vol. 11, no. 1, 2013, pp. 1–43.

news reports do suggest foreign interest in American use of surveillance technology to determine who the surveillance targets are.[80]

There is one more aspect of security that has to be taken into account: who the enemies are. As has been widely reported in the press, various countries have or are creating cyberespionage and cyberwarfare units. These are highly skilled and well-equipped groups, easily capable of finding and exploiting subtle flaws in systems. To use an easy analogy, comparing the capabilities of such units to those of garden-variety hackers is like comparing the fighting power of modern infantrymen to that of a comparable-sized group of drug gang members. When considering the security of any Internet-connected systems that might attract the hostile gaze of foreign powers, this must be taken into account.

Communications systems fall into this category and have done so for many, many years. Even apart from their purely military significance, American economic interests have long been targeted by other nations. In the early 1970s, for example, the Soviets reportedly used high-tech electronic eavesdropping devices to listen to the phone calls of American grain negotiators.[81] These days the attempts at economic espionage come not just from Russia, but also from China, France, Germany, Israel, Japan, South Korea, India, Indonesia, and Iran.[82]

In 2000, the Internet Engineering Task Force, the engineering group that develops Internet communications standards through its "Requests for Comment" (RFCs) documents, concluded, "adding a requirement for wiretapping will make affected protocol designs considerably more complex. Experience has shown that complexity almost inevitably jeopardizes the security of communications; there are also obvious risks raised by having to protect the access to the wiretap. This is in conflict with the goal of freedom from security loopholes."[83] The security vulnerabilities that a wiretap introduces into a communications system is a serious problem, yet it apparently gets little attention from law enforcement in its efforts to expand CALEA to IP-based communications.

---

[80] *See* Kenneth Corbin, "'Aurora' Cyber Attackers Were Really Running Counter-Intelligence", *CIO,* April 22, 2013, available at http://www.cio.com/article/732122/_Aurora_Cyber_Attackers_Were_Really_Running_Counter_Intelligence?taxonomyId=3089.

[83] Internet Engineering Task Force, RFC 2804, IETF Policy on Wiretapping (May 2000). One of the authors of this paper was on the Internet Architecture Board at the time and helped write the document.

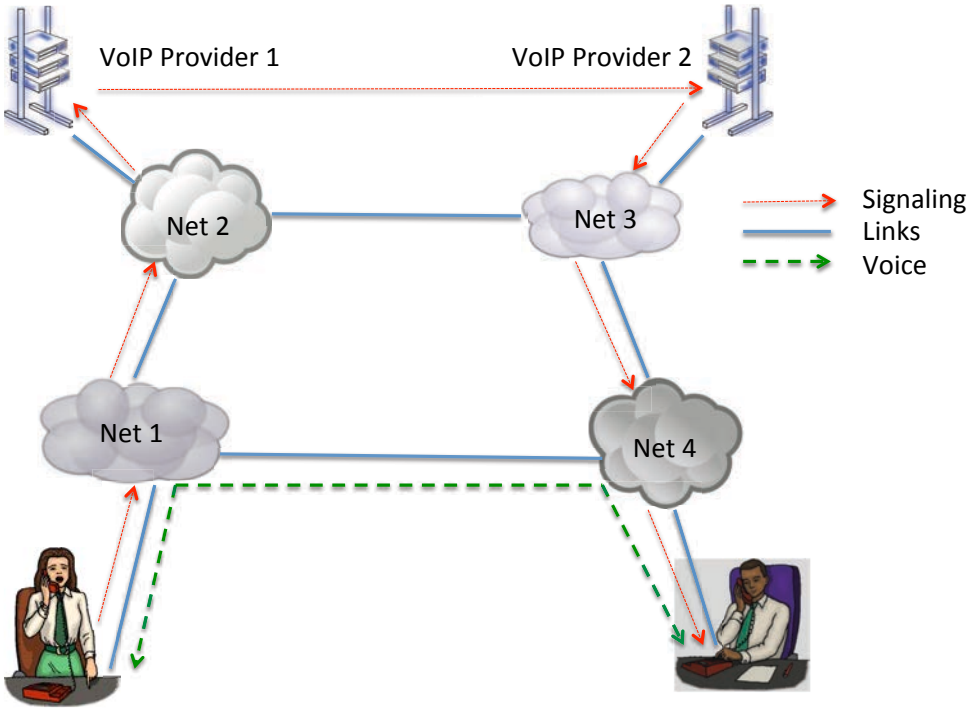

Figure 1: A Voice over IP (VoIP), showing physical links, the signaling path, and the voice path.

III. The Vulnerability Option

We have argued that extending CALEA to IP-based communications presents intolerable security risks and how modern communications systems are likely to impede wiretapping efforts. Given that, how might law enforcement wiretap modern communications?. Here we describe the vulnerability option: how they can resolve the wiretap problem, why vulnerabilities exist, and why the vulnerability "solution" must, in fact, always be part of the law-enforcement wiretap toolkit. We begin with a definition of terms.

A. Definition of Terms

We need to define a few commonly used technical terms in order to present the mechanics of employing a vulnerability for accessing a target system.

Vulnerability: A vulnerability is a weakness in a system that can potentially be manipulated by an unauthorized entity to allow exposure of some aspect of the system. Vulnerabilities can be bugs (defects) in the code, such as a "buffer overflow"[84] or a "'use-after-free instance"[85]', or misconfigurations, such as not changing a default password or running open, unused services.[86] Another common type of vulnerability results from not correctly limiting input text (this is also known

[84] A buffer overflow is caused by a program accepting more input than memory has been allocated for. Conceptually, imagine a clerk writing down someone's name, but the name as given is so long that it doesn't fit in the box on a form and spills over into the "Official Use Only" section of the form. A buffer overflow error was a central part of the Internet Worm of 1988, which resulted in the first case ever brought under the *Computer Fraud and Abuse Act*, 18 U.S.C. §1030; *see* United States v. Morris, 928 F.2d 504; 1991 U.S. App. LEXIS 3682. In some programming languages, e.g., Java, such overflows are detected automatically by the system; programmers using older languages, such as C, can use safe programming techniques that avoid the problem. A variety of tools can be used to detect potentially unsafe areas of programs. These have become increasingly common in the last 10 years, to very good effect.

[85] Programs can request storage space, then release—"free"—it when they are done; after that, the space is available for other uses. A use-after-free bug involves carefully crafted accesses to memory no longer allocated for its original purpose; if some other section of the program is now reusing that storage, this section of the program may be confused by the improper reuse.

[86] A service is a mechanism by which programs listen for and act on requests from other programs; often, these services are available to any other computer that can contact this one via the Internet. The best analogy is to room numbers in a building. The building itself has a single address (the computer analog is the IP address), but the mailroom is in room 25, the information counter is in room 80, and so on. Secure computer systems generally "listen" on very few ports, since each one represents a potential external vulnerability. Suppose, for example, that a computer that is not intended to act as a web server is in fact running web server code. A flaw in that web server can result in system penetration; the simplest fix is to turn off the web service since it is unneeded on that computer. *See* CERT Advisory CA-2001-19, July 19, 2001, for an example of problems caused by open, unneeded services.

as not sanitizing input), e.g., "SQL injection";[87] alternatively, a vulnerability can be as simple as using a birth-date of a loved one as a password. A vulnerability can be **exploited** by an attacker. A special instance of vulnerability is the:

Zero-day (or 0-day vulnerability): A zero-day is a vulnerability discovered and exploited prior to public awareness or disclosure to the vendor. Zero-days are frequently sold in the vulnerabilities market. The vendor and the public often only become aware of a zero-day after a system compromise.

Exploit: an exploit is the means used to gain unauthorized access to a system. This can be a software program, or a set of commands or actions. Exploits are usually classified by the vulnerability of which they take advantage, whether they require local (hands-on) access to the target system, or can be executed remotely or through a web page or email message (Drive-by).[88] The type of result obtained from running the exploit (rootkit, spoofing, key-logger) depends on the **payload.** The payload is chosen when the exploit is run or **launched**. An exploit demonstrates the use of the vulnerability in actual practice.

Payload: The payload of an exploit is the code that is executed on the target system giving the attacker the desired access. Payloads can be single action, such as surreptitiously creating a new user account on the system that allows future access, or multi action, such as opening a remote connection to an attacker's server and executing a stream of commands. The payload generally must be customized to the specific system architecture of the target.

Dropper: A dropper is a malware component or malicious program that installs the payload on the target system. A dropper can be single stage, a program that executes on the target system as a direct result of a successful exploit and carries a hidden instance of the payload, or it can be multi-stage, executing on the target system, but downloading files (including the payload) from a remote server.

Man-in-the-Middle attack: A Man-in-the-Middle attack is a method of gaining access to target information in which an active attacker interrupts the connection between the target and another resource and surreptitiously inserts itself as an intermediary. This is typically done between a target and a trusted resource, such as a bank or email server. To the target the attacker pretends to be the bank, while to the bank the attacker pretends to be the target. Any authentication credentials required (e.g., passwords or certificates) are **spoofed** by the attacker, so that each side believes they are communicating with the other. But because all

[87] In some contexts, parts of the input to a program can be interpreted as programming commands rather than as data. SQL injection attacks—in variant forms, they date back to at least the 1970s—occur when programmers do not filter input properly to delete such commands.
[88] A drive-by download is an attack perpetrated simply visiting a malicious or infected web site. No further action by the user is necessary for the attack to succeed. Such attacks *always* result from underlying flaws in the web browser.

communications are being transmitted through the attacker, the attacker is able to read and modify any messages it wishes to.

Spoofing: In the context of network security, a spoofing attack is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.[89]

B. How Vulnerabilities Help

Our claim is that pre-existing vulnerabilities in software makes extending CALEA unnecessary.[90] To understand the scenarios, it is necessary to give a simplified description of the structure of modern computer operating systems.[91] Systems are described in terms of "layers"; each layer provides some services to the layer above it, and requests services of the layer below it. Often, a combination of hardware and software enforces the boundary between layers, ensuring that only certain requests can be made of the lower layer.

The lowest layer we will mention is the hardware: CPU chips such as Intel's Pentium series, devices such as network interfaces and hard drives, USB ports, etc. For our purposes, we will assume that this layer is error-free and secure. While not strictly true, attacks at this level are generally more feasible for national-security purposes than for law enforcement.[92]

The next layer is generally called the "kernel". The kernel protects itself (with aid from the hardware); it is also the only component that directly communicates with external hardware such as the network. When a program needs to read or write from the network or a disk drive, it cannot do so directly; instead, it asks the kernel to perform the action for it. A consequence of this is that the kernel has to enforce "file permissions": which users of the computer own which file, who can read or write them, etc. That in turn implies that there must be some strong separation between programs run by different users; again, the kernel enforces this.

The last layer of interest is the "user level" or "application level". Virtually all programs of interest—web browsers, mailers, document editors and viewers, and so on—run at user level. Programs running are typically associated with some user. The user may be a physical individual; however, all modern systems have a large number of helper processes, sometimes known as "daemons," running as some

---

[89] Shirey, *id*., defines "spoofing" as equivalent to "masquerade attack" and defines the latter as "a type of threat action whereby an unauthorized entity gains access to a system or performs a malicious act by illegitimately posing as an authorized entity."

[90] Some of this material appeared in different form in Bellovin et al., *supra* note 6, *Going Bright* paper.

[91] These days, smart phones are built the same way; there is no need to discuss them separately.

[92] We will not discuss attacks like eavesdropping on encrypted WiFi signals. In principle, though, there might be exploitable vulnerabilities in the target's WiFi access point or router. These devices, though, are just computers and can be hacked like any other computers.

flavor of system pseudo-user. These handle such applications as the audio system, indexing files, insertion of USB devices, and more. A quick check of a modern Apple Mac showed no fewer than 10 different pseudo-users active on the machine.

All modern operating systems have a feature known as a "sandbox". A sandbox is a way of enforcing security by allowing program to run with fewer privileges than the user who has invoked it. Sandboxes are frequently used for programs perceived as exceptionally vulnerable to security holes; these include PDF viewers, web browsers, etc.

Vulnerabilities—and hence exploits of use to law enforcement—can occur at any layer, but the capabilities available to the exploit are different at different layers. While we defer details until Section IV, we note that for an exploit to work, more code is needed than just something that targets the vulnerability. In particular, to perform a wiretap—that is, to acquire the contents of a communication—the actual data sent or received has to be captured. This can be done in a particular application (e.g., Skype or a game with a voice communications feature), or it could be done at kernel level by tampering with a "device driver,"[93] in which case data from any application could be captured. A kernel exploit is well-positioned to modify device drivers; however, for complex technical reasons such an attack would find it more difficult to read and write files, export captured data via the network, etc.[94]

Most initial penetrations take place at application level.[95] The mechanisms vary widely, including infected attachments in email, malware on web pages, poor implementations of network protocols, and users downloading and voluntarily executing booby-trapped programs under a misapprehension as to the programs' purpose, provenance, and good intent.[96] The results are the same: some program the user had not intended is being run with the user's file access rights.

Under certain circumstances, this is sufficient for law-enforcement purposes. It generally provides adequate means for intercepting email; it may also suffice for

---

[93] A *device driver* is a special part of the kernel that communicates with input/output devices such as disks, audio ports, network interfaces, etc. *See*, e.g., Andrew S. Tanenbaum and Albert S. Woodhull, *Operating Systems Design and Implementation*, 3rd Edition, Prentice-Hall, 2006.

[94] Even a sketchy explanation of this is well beyond the scope of this paper. The primary problems are the nature of I/O APIs—they're generally designed to copy essential parameters from application level—and the difficulty of waiting for an I/O operation to complete without a "process context". *See* any standard operating systems textbook, e.g., *Tanenbaum and Woodhull*, footnote 93, *supra*.

[95] It is generally believed that since kernels do almost no processing of network packet contents (as opposed to their "headers"), they are therefore much less vulnerable to attacks. Examination of various compendia of vulnerabilities confirms this.

[96] A significant percentage of software downloaded via peer-to-peer networks contains malware; *see*, e.g., Michal Kryczka et al. "TorrentGuard: stopping scam and malware distribution in the BitTorrent ecosystem." *arXiv preprint arXiv:1105.3671* (2011). Andrew D. Berns and Eunjin EJ Jung, at 4 in "Searching for malware in BitTorrent." *University of Iowa, Tech. Rep. UICS-08-05, April* 24 (2008), note that much of this is "key generation or activation utility[ies]", i.e., tools for stealing software.

looking at the transcript files kept by some instant messaging programs. User level exploits are also useful for "remote search", though that poses other issues beyond the scope of this paper.

On the other hand, if the program penetrated is not used for the actual communications of interest, these exploits alone will not suffice. Consider that on most modern platforms, users—and hence the programs they run—do not have the ability to tamper with the kernel or system-owned files; this latter category generally includes applications such as Skype. Accordingly, if a law enforcement penetration for the purpose of eavesdropping is executed at user level, a second exploit known as a "local privilege escalation" attack is needed. This second attack gives the program elevated privileges and hence the ability to change device drivers, modify other files, etc.[97] While the two exploits are generally independent, frequently both are necessary; this complicates the attack.

There is one special case worth mentioning. Some daemons run with full system privileges; if these have faulty implementations of network protocols, only a single attack is needed. This is a venerable technique, going back to the first Internet worm.[98] While modern system designs try to avoid daemons with full privileges, in some situations this is unavoidable.

Historically, some applications have been considerably more vulnerable to user level attacks than others; these include web browsers and PDF viewers. As noted, modern operating systems often run these programs in "sandboxes", to prevent theft of or damage to user files.[99] Sandboxes may also deny the confined program the ability to run other system commands that may be utilized for privilege escalation. Accordingly, a third exploit may be necessary, to escape from the sandbox; following that, privilege escalation is used as before.

To summarize: there are many different points for initial attack; all have their limitations. System privileges are needed to modify applications or device drivers; these can be obtained via either a direct kernel attack, an attack on a system-level daemon, or via privilege escalation following an application level penetration.

C. Why Vulnerabilities Will Always Exist

We are suggesting use of pre-existing vulnerabilities for lawful access to communications. To understand why this is plausible, it is important to know a

---

[97] On Windows, the privileged user is known as "Administrator"; on Unix-like systems, including MacOS and Linux, it is known as "root".

[98] *See,* e.g., Eugene Spafford, "The Internet Worm Program", *Computer Communications Review* 19:1, January 1989, at 17-57, and J.A. Rochlis and M.W. Eichin, "With Microscope and Tweezers: The Worm from MIT's Perspective", *Comm. ACM* 32:6, June 1989, at 689-703.

[99] A "sandbox" is a mechanism to give application programs fewer privileges than those of the user who has invoked them.

fundamental tenet of software engineering: bugs happen. In his classic *The Mythical Man-Month*, Frederick Brooks explained why:[100]

> First, one must perform perfectly. The computer resembles the magic of legend in this respect, too. If one character, one pause, of the incantation is not strictly in proper form, the magic doesn't work. Human beings are not accustomed to being perfect, and few areas of human activity demand it. Adjusting to the requirement for perfection is, I think, the most difficult part of learning to program.

Because computers, of course, are dumb—they do exactly what they're told to do—programming has to be absolutely precise and correct. If a computer is told to do something stupid, it does it, while a human being would notice there's a problem. A person told to walk 50 meters then turn left would realize that there was an obstacle present, and prefer the path 52 meters down rather than walking into a tree trunk. A computer wouldn't, unless it had been specifically programmed to check for an impediment in its path. If it hasn't been programmed that way—if there is virtually any imperfection in code—a bug will result. This might be a rare one, but it will nonetheless be a bug.[101] If this bug should happen to be in a security-critical section of code, the result may be a vulnerability.

A National Research Council study described the situation this way:[102]

> [A]n overwhelming majority of security vulnerabilities are caused by "buggy" code. At least a third of the Computer Emergency Response Team (CERT) advisories since 1997, for example, concern inadequately checked input leading to character string overflows (a problem peculiar to C programming language handling of character strings). Moreover, less than 15 percent of all CERT advisories described problems that could have been fixed or avoided by proper use of cryptography.

It would seem that bugs should be easy to eliminate: test the program, and fix any problems that show up. Alas, bugs can be fiendishly hard to find. And complex programs have simply too many possible branches execution paths to be able to test them all.[103]

---

[100] Frederick P. Brooks, *The Mythical Man-Month*, Addison-Wesley, 20th Anniversary Edition, 1995, at 3.

[101] In one classic incident, a single missing hyphen in a program contributed to the loss of the *Mariner 1* space probe. *See* http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=MARIN1.

[102] Fred Schneider, ed., *Trust in Cyberspace*, National Academy Press, 1999, at 110.

[103] The single capability that gives a computer most of its power is the ability to do things conditionally. That is, it can test a condition—is this number greater than zero? Does this string of characters contain an apostrophe? Is there room on the page for another line?—and continue along one program path or another, depending on the result of the test. Each conditional operation can in principle double the number of possible execution paths. (The reality is not quite that bad, because not all tests are independent.) This means that a program with just 20 conditionals has more than

Brooks shows a diagram on bug comparing the predicted and actual rate of bugs in some complex code.[104] The projection assumed a slow start, a rapid increase in the debugging rate, and a leveling off that suggested that the last bugs had been found. Instead, the rate never leveled off, and the total number of bugs found was significantly higher than had been forecast.[105] Brooks himself suggests that testing takes about half of total development time.[106] Even this isn't enough, though: "Testing shows the presence, not the absence of bugs."[107]

We will not recount the myriad techniques other than testing that have been tried in an effort to eliminate bugs; let it suffice to say there have been many. These include formal mathematical methods, better programming and debugging tools, different organizational and procedural schemes, improved programming languages, and more. Many of these ideas have helped, but none have proved a panacea. The ability to produce error-free code is the Holy Grail of systems development: heavily desired but unattainable.[108]

When we are dealing with computer security, though, the question is somewhat different than "does this program have bugs?" Rather, the proper question is "do the security-sensitive parts of this system have bugs?" When formulated this way, there would seem to be an obvious solution: divide a complex system up into security-sensitive and security-insensitive pieces; bugs in the latter, though annoying, would not result in disaster. Such an approach has the added advantage of improving the correctness of the security-critical components. The bug rate in code increases more than linearly in the size of the program; a program that is twice

---

2^{20}—1,000,000—possible paths through it; one with 40 conditionals (a very tiny number for a realistic program) has more than 1,000,000,000,000. Exhaustive testing is not possible under these circumstances.

[104] *See* Brooks, *supra* foonote 100, at 42. The diagram is a previously unpublished one by John Harr.

[105] Neither the graph nor the text make it clear whether the graph ended because the project was finished or simply because it is a snapshot of a single year's experience and doesn't look at the entire project. The graph, presented at the 1969 Spring Joint Computer Conference, shows one year of experience building the #1 ESS; the programming undoubtedly took longer. *See* Phil Lapsley, *Exploding the Phone*, Grove Press, 2013 at 235 and W. Keister, R.W. Ketchledge, and H.E. Vaughn, "No. 1 ESS: System Organization and Objectives", *Bell System Techncial Journal* 43:5, Part 1 (September 1964) at 1832. New versions of the code were unlikely to have fewer bugs; rather, the bug rate *increases* after some point (Brooks, *supra*, at 53-54).

[106] *See* Brooks, *supra* footnote 100, at 10; *see* also the later explanation of the complexity of that model at 117.

[107] Edsger Dijkstra, quoted in J.N. Buxton and B. Randell, eds., *Software Engineering Techniques: Report on a conference sponsored by the NATO Science Committee, Rome, Italy, 27–31 October 1969*, April 1970, at 16.

[108] Operational errors are common, too. See, e.g., Barton Gellman, "NSA broke privacy rules thousands of times per year, audit finds", *Washington Post*, August 16, 2013, available at http://www.washingtonpost.com/world/national-security/nsa-broke-privacy-rules-thousands-of-times-per-year-audit-finds/2013/08/15/3310e554-05ca-11e3-a07f-49ddc7417125_story.html: "One in 10 incidents is attributed to a typographical error in which an analyst enters an incorrect query and retrieves data about U.S phone calls or e-mails." Another bug was confusing the country and city codes for Cairo, Egypt (20 2) with the area code for Washington, DC (202). These sorts of errors led to literally thousands of incidents of improper collection of surveillance data.

as large has more than twice as many bugs. Perhaps the security-sensitive section, which by definition is smaller, will thereby have many fewer bugs than the system as a whole.

This approach has been at the heart of most secure system designs for more than 50 years. It was set out mostly clearly in the so-called "Orange Book", the 1985 Department of Defense criteria for secure operating system design.[109] The Orange Book prescribed something called a "Trusted Computing Base", the security-essential portions of a system:[110]

> The heart of a trusted computer system is the Trusted Computing Base (TCB) which contains all of the elements of the system responsible for supporting the security policy and supporting the isolation of objects (code and data) on which the protection is based. The bounds of the TCB equate to the "security perimeter" referenced in some computer security literature. In the interest of understandable and maintainable protection, a TCB should be as simple as possible consistent with the functions it has to perform.

This dream has proved elusive for two very different reasons. First, modern TCBs are themselves extremely large, significantly bigger than the entirety of the 1970s- and 1980s-vintage systems. Although modern software is far more reliable, that does not translate into absolutely reliability. Second, the notion of the TCB is less clear than it once was. More and more serious security incidents target components that fit no one's definition of "trusted", but the attacks are effective nevertheless. Indeed, the very first Internet worm, in 1988, exploited holes outside what would likely have been considered part of the TCB.[111] It was, in essence though not by intent, a denial of service attack: it consumed most of the capacity of the infected machines. This all happened at user level; the affected programs were not part of the TCB.[112] Put another way, trying to break up the system into trusted and untrusted parts does not work as well as had been hoped; bugs anywhere can be and have been exploited by malware. It is worth noting that even one of today's complex applications is tens of times larger than entire systems of the 1980s, when the Orange Book was written. Today's operating systems are vastly larger.

We conclude that for the foreseeable future, computer systems will continue to have exploitable, useful holes. The distinction between flaws in the TCB and flaws

---

[109] DoD Computer Security Center, *DoD Trusted Computer System Evaluation Criteria*, 1985, 5200.28-STD, available at http://csrc.nist.gov/publications/secpubs/rainbow/std001.txt. The nickname comes from the color of its cover; it is part of a series of publications known collectively as "The Rainbow Series".

[110] *Id.* at 65.

[111] *See* Spafford or Eichin and Rochlis, fn 98, *supra*.

[112] This is not strictly true. For technical reasons, one of the programs that was successfully attacked did run with elevated privileges; however, neither the penetration nor the excess resource consumption by it were related to those privileges. It ran as privileged (and hence by definition as part of the TCB) because the importance of avoiding excess privilege was not as well understood in the general community at the time as it is today.

outside it is important. Non-TCB programs—frequently known as "user mode" or "application mode" program—have the privileges of the user who runs them; TCB programs are generally all-powerful and have access to more files, including the ability to change them.[113]

D. Why the Vulnerability Solution Must Exist Anyway

Considering lawful intercept purely as an economic question, it is tempting to ask which is a cheaper solution, a vulnerability-based approach or a CALEA-like law. The question, however, is not that simple. Even apart from our overriding theme—that applying CALEA to Internet software carries many very serious risks, to both security and innovation—and apart from the cost-shifting issue (with CALEA-like solutions, the bulk of the cost is not carried by law enforcement), there is a further, more fundamental issue: a vulnerability-based intercept capability must exist in any event. The question, then, is not which costs less but whether the incremental cost of CALEA is justifiable given that the other approach must be pursued in any case.

No matter what a CALEA-like law says, there will always be important cases where CALEA interfaces will not help law enforcement to conduct surveillance. Often, these instances will be extremely important, urgent cases: national security or counterterrorism investigations, or major drug gangs.[114] These groups, especially the first two, are more likely than are common criminals to use non-American or even custom-written communications software and procedures.[115] Other situations in which a new law won't help include people who use older software that hasn't been upgraded to include a lawful intercept feature, and more generally any communications application that automatically provides end-to-end encryption capability.[116]

In situations like these, where the case is important and built-in lawful intercept mechanisms are not available, using vulnerabilities becomes an attractive

---

[113] This stark dichotomy, between all-powerful and relatively powerless code, is generally seen by the computer security and operating system communities as a bad idea. Many schemes have been proposed to create intermediate levels of privilege; few, if any, have caught on *and* been more than miminally effective at protecting the system. There has been more success of late with "sandboxes.",

[114] The Mexican Zeta drug gang uses a home-built, encrypted radio network; *see* Michael Weissenstein, "Mexico's cartels build own national radio system", Associated Press, December 27, 2011, available at http://news.yahoo.com/mexicos-cartels-build-own-national-radio-system-200251816.html.

[115] Witness the case of the Russian sleeper agent ring arrested in 2010. They used special programs for *steganography*, a way of concealing the very existence of messages. *See* Noah Schactman, "FBI: Spies Hid Secret Messages on Public Websites", *Wired Danger Room Blog*, June 29, 2010, http://www.wired.com/dangerroom/2010/06/alleged-spies-hid-secret-messages-on-public-websites/.

[116] Even the current CALEA statute states that "A telecommunications carrier shall not be responsible for decrypting, or ensuring the government's ability to decrypt, any communication encrypted by a subscriber or customer, unless the encryption was provided by the carrier and the carrier possesses the information necessary to decrypt the communication." 47 USC 1002(b)(3). The "information necessary to decrypt the communications" is, typically, a cryptographic key. If end-users do their own key management, the provider is unlikely to have the keys.

alternative. The alternative—a so-called "black bag job", a covert search—is far riskier.[117]

As with so many other things in high technology, the notion of using vulnerabilities for eavesdropping has a relatively high start-up cost; continued use does not. Apart from the obvious drop in the cost per interception, the operational software is likely to improve over time. That is, as the developers have more time and gain more experience, the overall package will improve. It will provide more functionality, higher efficiency, and stronger resistance to detection. The actual exploits used will, as noted, change over time; even those, however, are likely to be usable in many more cases than in a CALEA-based world; this, too, will drive down the cost of each interception. In other words—and to a much great degree than in a CALEA-based approach—using vulnerabilities will improve law enforcement's abilities in all cases, especially the most critical ones.

IV. Vulnerability Mechanics

In this section we examine the potential use of vulnerabilities. We begin by exploring warrant issues for using exploits to wiretap. We discuss how vulnerabilities may be exploited, and consider minimization in this environment and what tools and procedures are available that law enforcement authorities might use or modify to gain access. We also discuss the vulnerability and exploit markets. Finally, we discuss what steps would be needed for productizing an exploit specifically for lawful access.

A. Warrant Issues

Obviously, any use of vulnerabilities for wiretapping requires proper authorization. However, the technologies involved suggest that the processes may be somewhat more involved than for conventional wiretaps.

One issue is that there are two distinct steps, exploiting the vulnerability—that is, hacking the target's machine, albeit with proper permission—and actually carrying out the desired interception. Arguably, two different court orders should be obtained. This is done on some occasions today in similar situations. Consider, for example, the process used in at least one case for the FBI's Computer and Internet Protocol Address Verifier (CIPAV), which has been installed on subjects' computers to send addressing and protocol data of the target machine to the FBI.[118] The technical details of CIPAV are not public, but information from documents released

---

[117] Such searches are performed when necessary; *see*, e.g., Schactman, *supra* footnote 115.

[118] J. Lynch, "New FBI Documents Provide Details on Government's Surveillance Software," Electronic Frontier Foundation, 29 Apr. 2011; https://www.eff.org/deeplinks/2011/04new-fbi-documents-show-depth-government#footnote2_01mhuxa . CIPAV is a current FBI software package analogous to what we are proposing here. Its capabilities, as described in an affidavit for a search warrant, include collecting the target machine's IP address, MAC address, operating system type and version, browser type and version, "certain registry-type information", last URL visited, etc.

under the Freedom of Information Act shows that the FBI did indeed use a two-step process to obtain the information in that case. The bureau first sought a search warrant to install CIPAV on the target's machine. Having obtained the IP address and other relevant information conducting surveillance, the FBI then sought a pen register/trap-and-trace order from the court. This, however, is not always done. In *In Re Warrant to Search a Target Computer at Premises Unknown*, Southern District of Texas, Houston Division, 2013 WL 1729765 (S.D. Tex. April 22, 2013), the FBI submitted a single Rule 41 warrant application, covering all activities: finding the target, installing their own software, gathering addresses, taking pictures, etc.[119]

Another issue that can cause complications is the need for ``technical reconnaissance'' to identify the proper target machine.[120] This may involve listening to other content of other conversations; this would, presumably, require its own authorization.

Finally, the design of this sort of tap presents some opportunities for minimization by technical means, prior to the usual minimization that is required by law. Arguably, this should be specified in the warrant as well.[121]

B. Architecture

How should a law enforcement exploit software platform be designed? The special legal requirements, the technical quirks involved in exploitation, the speed technology changes, the lifetime of a vulnerability, the need for non-proliferation, and even budgetary constraints all suggest that any framework of tools developed for surveillance be easily configurable and readily adaptable. This in turn suggests that a highly modular architecture is needed for a vulnerability-based communications intercept vehicle.[122]

---

[119] Mark Eckenwiler, formerly a top Justice Department authority on surveillance, has indicated that intrusions needed to execute pen register orders can be performed solely on that lesser standard; *see* "FBI Taps Hacker Tactics to Spy on Suspects", *Wall Street Journal*, August 3, 2013, available at http://online.wsj.com/article/SB10001424127887323997004578641993388259674.html.

[120] *See* Section IV.D, *infra*.

[121] *See* Section IV.C, *infra*.

[122] Designing systems to use modules is standard software engineering practice. By definition, modules communicate via well-defined interfaces, allowing easy substitution of different versions. (See, e.g., D.L. Parnas, "On the criteria to be used in decomposing systems into modules." *Communications of the ACM* 15.12 (1972): 1053-1058.) A good example of a modular framework is a picture editor. Many different file formats—JPEG, TIFF, PNG, etc.—can be imported. The editing is done the same way, regardless of the input format; following that, the new version can be stored in any of these formats. In other words, a file format input/output routine is a separate module. The same is true for vulnerability-based surveillance. With a well-designed framework, execution of a wiretap could be as simple as choosing a wiretap module, an exploit, and warrant information, entering the target information, and pressing "Go". The system will then build the payload for

The particular components to be used against any given target will vary widely. Consider the choice of initial exploit. For a target with an older (and unpatched) system, an older and publicly-known exploit might be sufficient. For wiretapping someone using a newer operating system, or one that's fully patched, an old vulnerability will no longer suffice, thus forcing the use of a newer but more sensitive one. Another target, not using the common application targeted by either of the previous two, might require yet a third vulnerability. Any of these exploited weakness could potentially be closed on the targets' systems at any time, which could require the use of yet another one.[123]

There are other considerations as well. If only voice communications are to be picked up, there is no need to include any keystroke-logging capability in the payload. Indeed, the less code is included, the less the risk of the tap being discovered. Perhaps more important, code that isn't included can't be repurposed by someone else, thus aiding in non-proliferation.[124] Beyond that, selective inclusion aids in warrant compliance, by limiting what is collected to what the court's order permits. This is discussed in more detail below.[125]

A modular framework can be extremely cost-effective relative to other designs. By design modules are plug-and-play. No matter how different they may be on the inside, the way the modules communicate with the framework is standardized. The design makes it easy to have many different people to develop exploits for the same framework, and straightforward for people to use new ones. When an exploit become obsolete, only the module containing that exploit needs to be rewritten. Pre-configured warrant modules provide assurance to law enforcement that exploit will collect the communications they need,[126] and to the judge that the exploit and payload behave as specified in the warrant. If the investigation changes and a new warrant module is needed, the exploit executable only needs to be recompiled with the new module, and reinstalled.

C. Technical Aspects of Minimization

The wiretap statue specifies that "Every order and extension thereof … shall be conducted in such a way as to minimize the interception of communications not otherwise subject to interception under this chapter".[127] While this is normally a matter for trial and appellate judges to rule on, a properly designed intercept package can carry out some of this task. This provides greater privacy for

---

automatic installation. New exploits or new warrant information are separate modules; the rest of the program isn't affected.

[123] *See* Section IV.E, *infra,* for a discussion of the lifetime of these components

[124] *See* Section V, *infra.*

[125] *See* Section IV.C, *infra*.

[126] *See* Section IV.C, *infra*.

[127] 18 U.S.C. §2518(5).

individuals not targeted by the warrant. More subtly, by automatically eliminating a lot of the extraneous content, it eases the task of humans charged with minimization and thus likely reduces their error rate.[128]

A warrant must clearly specify what communications may and may not be collected:

> Each order authorizing or approving the interception of any wire, oral, or electronic communication under this chapter shall specify…
>
> (a) the identity of the person, if known, whose communications are to be intercepted;
>
> …
>
> (c) a particular description of the type of communication sought to be intercepted.[129]

Intercepts that collect more than is authorized are legally problematic, to say the least.[130]

A modular architecture greatly simplifies the execution of the warrant. Modules for common warrant specifications would contain pre-configured values (such as types of data to collect or ignore, specified ports to listen on, and time-limits). The framework would compile these values into a properly tailored exploit executable automatically, without the need for any special configuration by the law enforcement technicians.[131]

---

[128] While we do not suggest or think that a program can perform full minimization, it can certainly carry out mechanical aspects, e.g. excluding services and perhaps users not covered by the warrant.

[129] 18 U.S.C. §2518(4).

[130] According to documents obtained by the Electronic Privacy Information Center under FOIA, when the FBI's UBL unit (Usama bin Laden) was conducting FISA surveillance, "The software was turned on and did not work properly. The FBI software not only picked up the E-mails under the electronic surveillance of the FBI's target [redacted] but also picked up E-mails on non-covered targets. The FBI technical person was apparently so upset that he destroyed all the E-mail take, including the take on [redacted] under the impression that no one from the FBI [redacted] was present to supervise the FBI technical person at the time." (Memo from [redacted] to Spike (Marion) Bowman, Subject: [redacted], April 5, 2000, http://www.epic.org/privacy/carnivore/fisa.html, last viewed August 18, 2006).

[131] "Compilation" is the process of turning human-readable "source code", written in a language like C or C++, into the string of bytes that are actually understood by the underlying hardware. At compilation time, it is possible to select which sections of the program should be included in the eventual module.

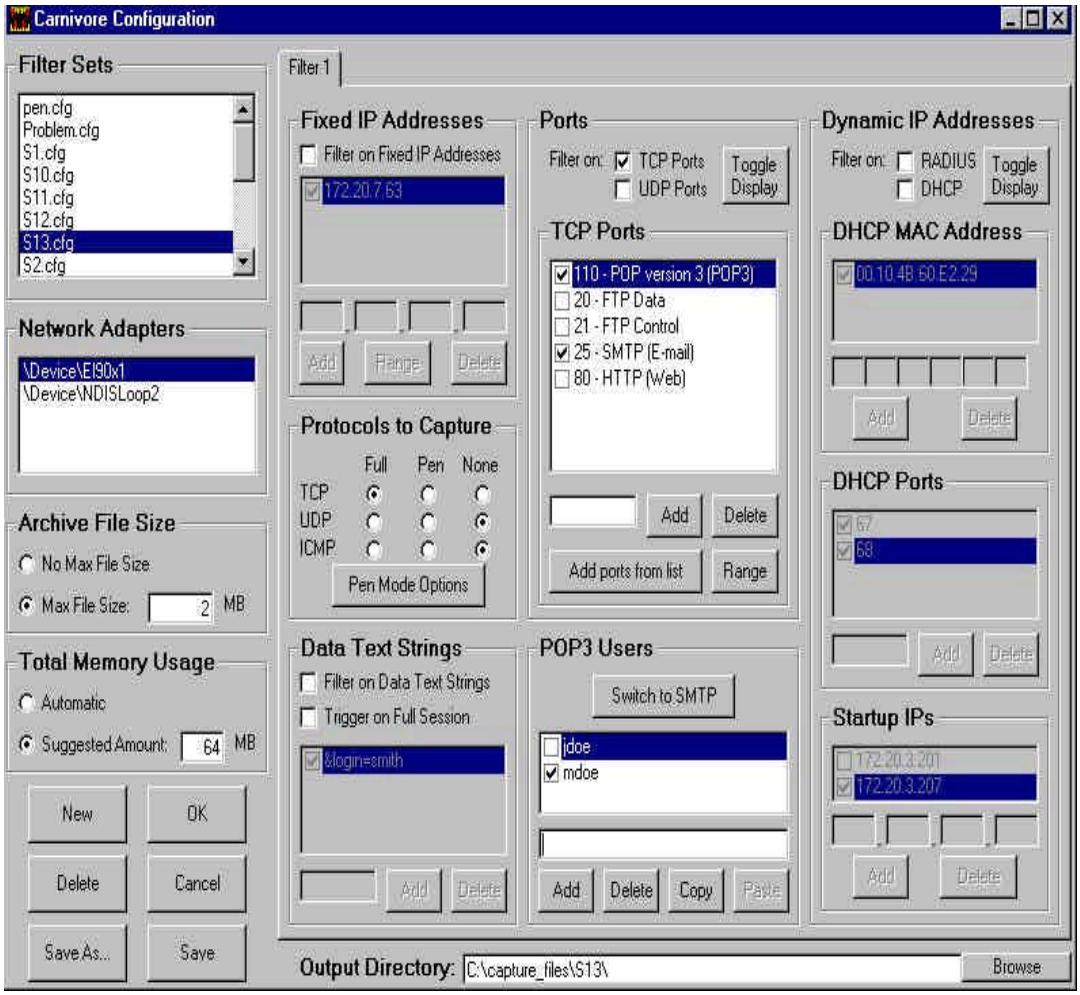

Figure 2: A sample warrant configuration screen from Carnivore. This filter is set up to intercept all inbound (POP) and outbound (SMTP) email from user mdoe.

The warrant configuration screen[132] from the (now obsolete) Carnivore wiretapping system[133] provides a useful example. Note how it has options for full content and pen register capture, fields for identifying which protocols should be captured, which IP addresses or users should have their data monitored, and so on. A similar scheme should be used here, with a crucial difference: modules not selected would not be included in the payload installed on the target's machine.

---

[132] This image is taken from Figure C-16 of Stephen P. Smith, Henry H. Perritt, Jr., Harold Krent, Stephen Mencik, J. Allen Crider, Mengfen Shyong, and Larry L. Reynolds, *Independent Review of the Carnivore System*: *Final Report*, December 8, 2000, IITRI CR-030-21, available at http://www.epic.org/privacy/carnivore/carniv_final.pdf .

[133] Carnivore was later renamed as the DCS 1000, and has since been retired in favor of commercial solutions. The apparent abandonment of the package is discussed in the 2002 and 2003 FBI reports to Congress (*Carnivore/DCS-1000 Report to Congress,* February 24, 2003, and December 18, 2003, available at https://epic.org/privacy/carnivore/2002_report.pdf and https://epic.org/privacy/carnivore/2003_report.pdf).

Other information can also be used for minimization. Assume, for example, that police know from other means that their suspect uses only one of the user profiles (i.e., logins) on a shared computer.[134] The intercept module, if properly configured, can operate only when that user is logged in. Similar filters can be used for communications applications, e.g., Skype, that have their own logins.

D. Technical Reconnaissance

The reconnaissance phase—learning enough about the target to install the necessary monitoring software—is essential to a successful compromise. Because exploits must be exquisitely tailored to particular versions and patch levels, using the wrong exploit frequently results in failures, and can even raise alerts or cause suspicious crashes. There are a number of widely used, readily available tools. Many of the best tools are even available in a free, ready-to-use downloadable toolbox; see, e.g., the Backtrack-Linux Penetration Testing Distribution.

The most common first step is to check publicly available information. DNS[135] and Whois[136] lookups are used to find Internet domain and IP information. Simple use of search engines and scouring the social media sites often provides some information about the target's operating system, cell-phone platform, service provider, and commonly used applications. With the appropriate legal process, e.g., a subpoena or court order under 18 U.S.C. §2703(d), some of this information may also be available from the service provider.

If the investigators have access to some emails from the target, a great deal of information may be found by studying the headers. An examination of some of our test emails showed such lines as:

> Mime-Version: 1.0 (Mac OS X Mail 6.2 \(1499\))
> X-Mailer: Apple Mail (2.1499)

and

> X-Mailer: iPhone Mail (10B146).

which are rather clear indicators of which operating system is in use.

---

[134] This is sometimes the case; *see, e.g.,* State of Ohio v. Castagnola, 2013 Ohio 1215, 2013 Ohio App. LEXIS 1115 (2013).

[135] The DNS—the Domain Name System—is used to convert human-friendly names such as www.fbi.gov to the number IP address understood by low-level Internet hardware. Information in the DNS is especially useful when trying to break into organizations rather than individual users' computers; *see, e.g.,* Chapter 6 of William Cheswick, Steven M. Bellovin, and Avi Rubin, *Firewalls and Internet Security*, Second Edition, Addison-Wesley, 2003.

[136] Whois is a service listing the ownership of domain names, address blocks, etc.

To remotely access a machine an attacker generally needs to know the IP and/or MAC addresses of the machine,[137] the operating system (including exact version and patch level), what services are running on the machine, which communications ports are open, what applications are installed and whether there the system contains any known vulnerabilities. This process of discovery is referred to as "Mapping" and "Enumeration".[138]

Mapping can be of the system or the network (or both). Network mapping can be WiFi or Ethernet, and can refer to finding hidden networks, or to enumerating all the devices and their addresses connected to a particular network. Mapping the target device or system requires finding the so-called "MAC address", a hardware address transmitted when speaking over Ethernet, WiFi, and Bluetooth networks. If the target of a tap is using a smartphone at a public hotspot, detecting that person's MAC address could, for example, reveal what brand of phone is being used.

Another way to ascertain the system version is to perform "OS fingerprinting". OS fingerprinting involves looking for subtle differences in the network protocol implementations of different operating systems, and in particular the response of the system being examined to various probes. NMAP, a freely available popular network security tool,[139] is most commonly used. In addition to OS fingerprinting, NMAP does open service and open port identification and limited vulnerability scanning.

The final step in the information-gathering phase is to scan the target system to see if it is vulnerable to common vulnerabilities.[140]

---

[137] IP and MAC addresses are networking concepts. MAC addresses are generally hard-wired in a computer's communications hardware, though sophisticated users can change them. IP addresses are often transient, but tend to remain the same for a given computer in a given location. While IP addresses are typically assigned by the network administrator of the site at which the computer is located, MAC addresses are assigned by the manufacturer and therefore indicate the computer type and model.

[138] On networked computer systems, services offered are assigned to particular (and generally standardized) "port numbers", a more or less arbitrary value between 1-65535. Port enumeration is the process of seeing what ports, and hence what services, are available on a given system. Using open ports for intrasystem communication, rather than more secure alternatives, was one of the items cited in the FTC complaint against HTC; *see In the Matter of HTC America, a Corporation*, Complaint, FTC File No. 122 3049, 2013.

[139] Gordon "Fyodor" Lyon, *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*, Nmap Project, 2009.

[140] There are a number of widely-used vulnerability scanning systems. Nessus (available from http://www.tenable.com/products/nessus) is the most widely used one; it can scan for thousands of vulnerabilities and plug-ins, and even provides detailed mobile device information (serial numbers, model, version, last connection timestamps). Another popular one is Nexpose (https://www.rapid7.com/products/nexpose/).

E. Finding Vulnerabilities

Once the target has been adequately identified and scanned, a suitable vulnerability must be identified. The primary criterion, of course, is compatibility with the user's operating system; another crucial one is mode of delivery. Some exploits, for example, can be delivered by email messages; others require the user visiting a particular web page, or opening a file with a specific, vulnerable application. Email delivery is easiest because it doesn't require the user to take any particular action, but apart from the fact that it might be noticed there is always the risk that a spam filter will catch it.[141] Another class of exploits requires being on the same local network[142] as the victim, or on an interconnected network if there are no intervening firewalls.[143] Even infected USB flash drives have been used; indeed, the Stuxnet attack on the Iranian nuclear centrifuge plant is believed to have started that way.[144]

Many exploits are publicly announced;[145] these are often available in easy-to-launch pre-packaged scripts. The Metasploit Project hosts the largest database of these scripted publicly available exploits (called 'modules').[146] These modules can be utilized by a number of different exploitation applications such as the Metasploit Framework and Core Impact Pro.[147] The NIST National Vulnerability

---

[141] Sending email messages crafted to appear genuine to a particular target is known as "spear-phishing". In skilled hands, spear-phishing is extremely effective. Press reports suggest that is one of the primary schemes used by cyberespionage units; *see, e.g.,* Jaikumar Vijayan, "DHS warns of spear-phishing campaign against energy companies", *Computerworld*, April 5, 2013, available at https://www.computerworld.com/s/article/9238190/DHS_warns_of_spear_phishing_campaign_against_energy_companies.

[142] A LAN (Local Area Network) is generally a high-speed network that covers a relatively small area. Typical LANs include most home networks, WiFi hotspots, or, in an enterprise, a single department. LANs are interconnected to each other or to WANs (Wide Area Network) by *routers*.

[143] Most home routers are technically known as Network Address Translators (NATs). For these purposes, NATs serve the same purpose as firewalls; these attacks cannot be launched at a target that is behind a NAT.

[144] *See Stuxnet, supra* footnote 17. It is unclear how the initial Stuxnet infection was launched. One theory is advanced in Bamford, "NSA Snooping Was Only the Beginning. Meet the Spy Chief Leading Us Into Cyberwar", Wired Threat Level Blog, June 12, 2013, available at http://www.wired.com/threatlevel/2013/06/general-keith-alexander-cyberwar/all/.

[145] The US Computer Emergency Readiness Team US-CERT maintains a frequently updated list of vulnerabilities. Security researchers and privately owned research laboratories such as Vulnerability Lab and Immunityinc announce vulnerabilities on websites and Twitter when they are discovered. Verified vulnerabilities are collected, categorized, and enumerated in the comprehensible, searchable NIST NVD database.

[146] The Metasploit Project (owned by Rapid7). Each of these exploits in the database consists of a specific vulnerability packaged into a module which can be loaded into an attack application, such as the Metasploit Framework, to run. Because of the popularity of the Metasploit Framework, many exploits sold are available as Metasploit modules. See https://exploithub.com for some examples.

[147] The Metasploit Framework, available from http://www.metasploit.com, is the most widely used exploitation application available today. It is available in both free and commercial versions and has a wide developer base. Core Impact Pro is a separate commercially product. Core Impact Pro may be purchased from http://www.coresecurity.com

Database(NVD) lists all the known vulnerabilities, including what versions of what systems are affected and references to more information (but no actual exploit information). Actual information about the exploit, including an executable script or some proof-of-concept source code is often published on one of a number of well-regarded websites and public mailing lists.

The second class contains the privately held exploits; these include the zero-days described above, as well as exploits for sale by professional security vulnerability researchers. We discuss these in detail in Section G.

Sometimes, no publicly available vulnerabilities will be usable, and the option of purchasing one from the vulnerabilities market will be undersireable or unavailable. In that case, law enforcement agents—more likely, a central "Vulnerability Lab"—must find one.[148] While this issue is out of scope here, we note there are many commonly available tools regularly used for this purpose by software vendors trying to protect their products and by attackers.

Finally, in the rare case where directly compromising a target platform through an exploit is not possible, a technique known as a "Man-in-the-Middle" (MitM) attack might be used.[149] Such attacks involve interrupting the communications path between the target and some site the target is trying to access; the attack tool then intercepts communications intended for that resource. A successful MitM attack might be another way to launch an attack; alternatively, it could permit acquisition of passwords and account information that would provide law enforcement with access to other useful resources.[150]

F. Exploits and Productizing

While off-the-shelf exploits may be available to law enforcement on the black market, police do not require their functionality, which is installing general-purpose remote-access malware to send spam, steal bank account numbers, etc. Rather, they wish to gather specific items of data authorized by the warrant, and to do so in a form suitable for presentation in court. In addition, access to a target system by a

---

[148] The FBI already operates the Domestic Communications Assistance Center, which apparently does at least some of this; *see, e.g.,* Caproni, "Going Dark: Lawful Electronic Surveillance in the Face of New Technologies," Subcommittee on Crime, Terrorism, and Homeland Security, Committee on Judiciary, February 17, 2011, http://judiciary.house.gov/hearings/hear_02172011.html, and D. McCullagh, "FBI Quietly Forms Secretive Net-Surveillance," *CNET*, May 22, 2012, http://news.cnet.com/ 8301-1009_3-57439734-83/fbi-quietly-forms-secretive -net-surveillance-unit.

[149] MitM attacks can be used at any time. However, they are almost always harder to do, since they require interfering with the traffic of exactly one user who may be at an unknown location. They are also more detectable than other attacks, albeit only by very sophisticated users.

[150] Depending on the provisions of the original warrant, it may be necessary to seek a modification. In particular, a warrant permitting interception of communications does not grant the right to search stored email archives; that would require an order under the Stored Communications Act (18 U.S.C. 2701 *et seq*).

law enforcement agent must take care to preserve evidence and chain of custody.[151] This implies due attention to precise logging of exactly what was done, when, and by whom. Consequently, off-the-shelf exploits (as opposed to vulnerabilities) are by themselves not likely to be particularly useful to law enforcement, except as a starting point or perhaps under exigent circumstances.[152]

The three parts of a law enforcement eavesdropping product—the exploit, which provides access to the system, the eavesdropping code, and the supporting infrastructure—all have different characteristics and lifetimes. Due to their specificity, installation characteristics, vendor patches, etc., exploits have the shortest lifetime. Accordingly, a good methodology for their use is the dropper/payload model, where the product is composed of two parts, a *penetrator* and a specially encrypted payload *that is specifically encrypted for the particular target*. The penetrator is the dropper, the initially injected part that exploits the actual vulnerability and thus gaining access to the target system. Once access is acquired, the penetrator will decrypt the law enforcement-specific payload. Encrypting the payload is a security measure to ensure that the penetration code can't easily be detected or reused by criminals; it also ensures that the payload targets the correct system.

The latter is accomplished by using target-specific information, such as serial numbers, the MAC address, IP address, etc., as the key to encrypt and decrypt the payload.[153] The penetrator picks this up at payload installation time; the earlier technical reconnaissance would have acquired the same information. This method protects untargeted machines from compromise: if the code is executed on the wrong machine, decryption will fail.

The payload itself should be designed to provide the access specified in the warrant with minimal changes to the target system. What changes are necessary should be logged and time-stamped in such a way as to provide documentation that vital evidence was neither altered nor destroyed. If the warrant includes provisions for recording communications, the payload should also contain provisions for minimization, including "recording on/off" switches and the length and time of communications recorded. Payloads don't change very much over time. While they may need to adapt to different major versions of operating systems, they generally rely on features not likely to change very often. Similarly, malicious payloads that have already been installed are rarely disabled by vendor patches.

---

[151] See Timothy M. O'Shea and James Darnell, "Admissibility of Forensic Cell Phone Evidence", in US Attorneys' Bulletin 56:6, November 2011, available at http://www.justice.gov/usao/eousa/foia_reading_room/usab5906.pdf . Also see Department of Justice Electronic Surveillance Manual, June 2005, for a discussion of sealing intercepts to protect their integrity.

[152] *See* Section VII.B, *infra.*

[153] Encryption is accomplished through the use of an algorithm, which may be public, and a *key*, a piece of secret data. If the encryption algorithm is strong, it should be effectively impossible to decrypt the file without knowledge of the key.

The infrastructure has an intermediate lifetime. Some of it, such as the code to set up encrypted channels to the investigators, is straightforward and not particularly tied to unusual law-enforcement needs; this code will be quite long lived. The command-and-control subsystem—the mechanism with which investigators control the tap, turn recording on and off, etc.—is similarly straightforward, although the fine details will be specific to this application. Much of this code will be virtually the same even across many different operating systems. On the other hand, the concealment mechanisms—the code that hides the existence of the payload from the computer's owner, and even from specialists who may have been hired to "sweep" the computer for bugs—is likely to be highly dependent on the operating system, including the particular version, and will change fairly frequently.

It is a good idea for the payload to have a self-destruct option, perhaps the time-limit set by the warrant after which the law enforcement software restores the target system to its pre-exploit state and erases itself and removes all evidence of its presence.[154] This not only helps prevent proliferation, it may be necessary to comply with the legal requirements for time limits on wiretap orders.[155]

A good example of how this might work in practice is demonstrated in a variant of Stuxnet[156] called Gauss.[157] Discovered in August 2012, Gauss is apparently an espionage tool. It uses a known vulnerability and shares some code with other known malware in its dropper, but even today, after several months of intense analysis, the behavior of its payload remain unknown. Gauss uses cryptographic methods and tools, and only installs and runs on machines specifically targeted by Gauss's developers; on non-targeted machines it remains encrypted and inert. Gauss also sets up a secure method to send data to its command and control centers. *Ars Technica* reports that "The setup suggests that the command servers handled massive amounts of traffic."[158]

G. The Vulnerabilities Market

With the availability of openly published vulnerability information and free exploitation tools, one might question why we discuss purchasing vulnerabilities or

---

[154] Fritz Hohl, "Time limited blackbox security: Protecting mobile agents from malicious hosts." *Mobile Agents and Security*. Springer Berlin Heidelberg, 1998. 92-113, available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.8427

[155] 18 U.S.C. §2518(4)(e).

[156] *See Stuxnet*, *supra* footnote 17.

[157] Dan Goodin, "Nation-sponsored malware with Stuxnet ties has Mystery Warhead", *Ars Technica Security*. August 9, 2012 See, http://arstechnica.com/security/2012/08/nation-sponsored-malware-has-mystery-warhead/
The report mentioned in the article can be found at https://www.securelist.com/en/downloads/vlpdfs/kaspersky-lab-gauss.pdf. Again, this was an intelligence effort, not a law enforcement one; nevertheless, it provides a proof of concept.

[158] Dan Goodin, March 14, 2013. "Puzzle box: The quest to crack the world's most mysterious malware warhead", Ars Technica Security, http://arstechnica.com/security/2013/03/the-worlds-most-mysterious-potentially-destructive-malware-is-not-stuxnet/

exploits from researchers at all. The answer is the improved security of target systems. As software developers and vendors have improved the quality of their software and incorporated defenses such as firewalls and anti-virus packages, vulnerabilities have become harder to find and to exploit.

Software companies have also generally accelerated the rate at which they release security patches after critical vulnerabilities have been announced. This may result in a well-patched and well-maintained system being more difficult to compromise. Additionally, as stated above, exploits must be carefully tailored to the individual target machine. This means it requires more skill to develop a working exploit, making new effective exploits a valuable commodity for their creator. Thus a technically savvy target, someone who is conscientious about maintaining their system with up-to-date security patches, is careful about not installing software from unverified sources, uses encryption, doesn't open links from email, and doesn't access questionable websites may not be vulnerable to the easy public exploits. If law enforcement wishes to use a zero-day or lesser-known vulnerability to exploit a target, it must either have the appropriate vulnerability and exploit already on the shelf or else it must purchase one on the open market, itself a relatively recent phenomenon.

Finally, there may sometimes be a need to tap a particular suspect as quickly as possible. If there are no suitable off-the-shelf exploits available to the investigators and no time to find a new one, purchasing one may be the best option.[159]

The overt vulnerabilities marketplace had its start in 2004 when Mozilla launched the first successful bug-bounty program.[160] This program, still in effect today, pays security researchers for original vulnerabilities they discover.[161] Many other companies have followed suit with their own bug-bounty programs. Product developers are not the only groups that are interested in obtaining information regarding software vulnerabilities. Governments and computer security service providers such as iDefense and ZDI also pay for vulnerability information particularly if the details on how to use it have not been made public (zero-days)[162]

---

[159] That an exploit has been purchased instead of being developed in-house does not change the need to report it promptly. However, under urgent conditions some delay may be appropriate. *See* Section VII.B.

[160] The original Mozilla Foundation press release announcing the Mozilla Security Bug Bounty Program can be found here: https://www.mozilla.org/en-US/press/mozilla-2004-08-02.html, For further examples see, Kim Zetter, "With Millions Paid in Hacker Bug Bounties, Is the Internet Any Safer?," *Wired Magazine* – which lists prices, total paid out and date launched for several Bug Bounty programs. Available at http://www.wired.com/threatlevel/2012/11/bug-bounties/all/

[161] See https://www.mozilla.org/security/bug-bounty.html

[162] In Feb 2006, iDefense, a vulnerability research company owned by Verisign Inc, offered $10,000 prize for a 'previously unknown' Microsoft security vulnerability. One of the requirements for winning the prize was that the vulnerability be submitted exclusively to iDefense. Similarly Tipping Point's ZDI's FAQ states that once a vulnerability has been assigned to TippingPoint, it cannot be distributed—or even

The overt and underground markets in vulnerabilities, exploits and zero-days has expanded in recent years.[163] Many legitimate security research firms have made finding vulnerabilities and developing exploits for sale part of their business model.[164] Companies and individual security researchers sell information about privately discovered vulnerabilities (often with a proof-of-concept) or full-blown exploit code to groups of subscribers and to individuals. The prices and amount of detail made public varies. Some companies (e.g., Vulnerability-Lab) and researchers publicly announce that a vulnerability has been discovered in a particular product, but reserve actual details for their customers. Other firms, such as Endgame, keep even the knowledge of the existence of the vulnerability for private sale. Prices range from $20.00 - $250,000.00[165], but exclusive access to a critical zero-day is generally the most expensive. Recent news reports suggest that national governments (in particular intelligence and military agencies) have become major buyers.[166]

Companies such as Vupen and Vulnerability-Lab sell subscription services, which provide private and exclusive detailed information on disclosed or private critical vulnerabilities, to governments, law-enforcement authorities, and corporations. Annual subscriptions can run as high as $100,000 a year.[167] These companies also sell working exploits and offer special targeted exploit development for additional fees; exploit prices range between $5000 and $250,000.00. The most valuable are those zero-days that can be used for cyber warfare (e.g., the Endgame Systems pricelist includes a 25 exploit package for $2.5 million[168]). Zero-days and exploits can also be purchased from exploit brokers such as Netragard or private brokers

---

discussed—elsewhere until a patch is available from the vendor. See e.g., http://blog.washingtonpost.com/securityfix/2006/02/wanted_critical_windows_flaw_r.html

[163] Presumably, if criminals were the only ones interested in purchasing vulnerabilities, the market would still exist, but it would be underground. Similar markets do exist for other forms of criminal software, such as bots, credit card number loggers, etc.

[164] Some prominent examples include: Vupen Security, Vulnerability-Laboratory, Immunityinc, Netragard, NSS Labs, Inc and Raytheon.

[165] Exploits currently offered for public sale from a wide variety of independent researchers may be purchased from http://exploithub.com. Further examples can be found in Andy Greenberg, "Meet the Hackers Who Sell Spies the Tools to Crack Your PC and Get Paid Six Figure Fees", *Forbes Magazine*. Available at: http://www.forbes.com/sites/andygreenberg/2012/03/21/meet-the-hackers-who-sell-spies-the-tools-to-crack-your-pc-and-get-paid-six-figure-fees/

[166] *See* Nicole Perlroth and David E. Sanger, "Nations Buying as Hackers Sell Flaws in Computer Code", *New York Times*, July 14, 2013, available at https://www.nytimes.com/2013/07/14/world/europe/nations-buying-as-hackers-sell-computer-flaws.html.

[167] *Id.*

[168] Michael Riley and Ashlee Vance, "Cyber Weapons: The New Arms Race", *Bloomberg Businessweek Magazine.* Quoting David Baker, vice-president for services at the security firm IOActive, "Endgame is a well-known broker of zero-days between the community and the government," and By "community," he means hackers. "Some of the big zero-days have ended up in government hands via Endgame," Baker says. Available at: http://www.businessweek.com/magazine/cyber-weapons-the-new-arms-race-07212011.html#p4

who bid on exploits from sellers and negotiate with buyers on behalf of individual exploit developers.[169]

The FBI has apparently already used vulnerabilities to download exploits and extract information from various targets machines. But if law enforcement uses vulnerabilities and exploits to conduct wiretaps when other methods fail (and as an alternative to CALEA-style taps in the IP world), it will face a difference in scale in the use of such techniques—and thus a difference in kind. That raises not just technical questions, but complex ethical and legal concerns as well. In the sections that follow, we turn to those.

[169] A number of recent reports have been published recently documenting the vulnerabilities market and the brokers who negotiate between buyers and sellers. See: The Economist, "The Digital Arms Trade" author unknown, available at:. http://www.economist.com/news/business/21574478-market-software-helps-hackers-penetrate-computer-systems-digital-arms-trade, Netragard, "Zero Day Exploit Acquisition Program, available at: http://www.netragard.com/zero-day-exploit-acquisition-program, and Andy Greenberg, "Shopping For Zero-Days: A Price-List for Hackers Secret Software Exploits, Forbes, March 23, 2013. Available at http://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-exploits/.

V. Preventing Proliferation

As might already be clear, the use of an exploit to download a wiretap is far more complex than the simple placing of two alligator clips upon a wire. But what is far more critical is that the exploits employed in the installation of the wiretap may spread beyond the targeted device. Given that possibility, does the government even have the right to use vulnerabilities in its efforts to combat crime and protect national security? We consider this issue, then move on to examine techniques to prevent proliferation of the exploit beyond the intended target.

A. Policy Concerns in Deploying Exploits to Wiretap

We have started from some assumptions. There is probable cause that the suspect is committing a serious crime and using the targeted communications device to do so; other means of investigation have been tried and not netted the requisite information. A wiretap order has been authorized. But the target is using a communications device, whether end-to-end encryption or peer-to-peer technology or something not yet dreamt of that prevents the standard methods of interception from working. Is it moral to use an exploit to intercept the communication when there is some risk, however small—but perhaps larger than anticipated—that the exploit may escape the device and be used elsewhere, causing great harm?

The issue of doing good but potentially doing harm in the process is a well-known problem in philosophy: "the doctrine of double effect," in which one pursues a moral action that has a consequence of causing harm. The philosopher Phillipa Foot argued that the distinctions should be between what we do (direct intention) and what we allow (oblique action), between negative duties—avoidance of harm—and positive ones—activities to help,[170] and between duties and voluntary actions.

In using vulnerabilities to execute wiretaps, law-enforcement investigators are performing their required duty of investigating a criminal activity. Under Title III, being granted a wiretap order means that evidence is essentially unobtainable in other ways.[171] The duty of investigating the criminal activity may require

---

[170] Phillipa Foot, The Problem of Abortion and the Doctrine of the Double Effect, in VIRTUES AND VICES AND OTHER ESSAYS IN MORAL PHILOSOPHY, Oxford (1978) at 19–32

[171] Recall that Section 2518(3)(c) requires that "normal investigative procedures have been tried and have failed or reasonably appear to be unlikely to succeed if tried or to be too dangerous." This does not mean, however, that wiretaps may only be the last resort; see United States v. Smith, 893 F.2d 1573, 1582 (9th Cir.1990).

wiretapping. If the only way to effect the wiretap is through the use of an exploit, then, following the logic presented by Foot regarding duty, this is the way to proceed. *But there must be due diligence to contain the harm*. There are several aspects to this, both the requirement to fully vet necessity and balance that against the good that may result, and the requirement that the exploit be designed to prevent proliferation beyond the target.

The law is all about balancing competing social goods. For example, the Fourth Amendment does exactly that, balancing the social good of society to protect itself against the social good of protecting individual privacy and security.[172] Consider law enforcement's use of vulnerabilities in the context of competing social goods. Use of vulnerabilities, at least without reporting them is not unlike police use of confidential informants (CIs). CIs inform investigations even while aiding criminal activity.

A common law enforcement tactic is to use a lesser criminal to gather evidence about a higher-up. Within limits, crimes (including further crimes) committed by a "flipped" individual are largely forgiven, so long as that person is providing good evidence against the real target of the investigation. As Daniel J. Castleman, chief of the Investigative Division of the Manhattan district attorney's office, explained, "With confidential informants we get the benefit of intimate knowledge of criminal schemes by criminals, and that is a very effective way to investigate crime."[173]

What happens with wiretaps implemented via exploits is ultimately not very different. In both cases law enforcement seeks to catch what it believes to be a genuinely dangerous criminal. But here it seeks to do so by the collection of wiretap evidence. Installing the tap requires exploiting a vulnerability that law enforcement hopes will not be repaired before the tap is in place.

The purchase—and secrecy—of vulnerabilities raises several similar moral dilemmas as the use of confidential informants (CIs). The history of police use of CIs is replete with instances where an informant has gone much too far, committing or failing to stop serious criminal activity; this has even included murder.[174] With wiretaps the "much too far" is of a somewhat different character, but with similar consequences: some crimes that the government could have stopped may not be

---

[172] While the usual interpretation of the Fourth Amendment is that it centers on protecting the privacy of the individual against searches by the state, Jed Rubenfeld convincingly argues that the amendment really concerns providing security for individuals against searches by the state; *see* Jed Rubenfeld, *The End of Privacy*, Stanford L.R. Vol. 61, Issue 1, (October 2008), 118-199.

[173] Alan Feuer and Al Baker, "Officers' Arrest Puts Spotlight on Police Use of Informants," NEW YORK TIMES, January 27, 2008.

[174] There are multiple such examples, including the well-known one of the shooting of Viola Liuzzo, a white supporter of the Civil Rights movement who was shot by Ku Klux Klan members while driving from a march in Selma, Alabama, one of whom was an FBI informant (Diane McWhorter, CARRY ME HOME: BIRMINGHAM, ALABAMA: THE CLIMACTIC BATTLE OF THE CIVIL RIGHTS REVOLUTION, Simon and Schuster, at 572-573).

prevented. By not reporting the vulnerability to the vendor and speeding its repair, law enforcement's inactivity is, by silence, potentially enabling criminal activity against other users of the same hardware or software. It is thus useful to examine how law views the competing interests of preventing crime versus investigating criminal activity in the use of confidential informants, the closest analogy there is in practice to the use of unreported vulnerabilities.

In *United States v. Murphy,*[175] the Seventh Circuit considered a case in which FBI agents created fictitious cases in the Cook County Courts in order to uncover corruption within the legal system. The Seventh Circuit ruled that the false cases were a legitimate investigatory tool, observing that, "the phantom cases had no decent place in court. But it is no more decent to make up a phantom business deal and offer to bribe a Member of Congress. In the pursuit of crime the Government is not confined to behavior suitable for the drawing room. It may use decoys, and provide the essential tools of the offense. The creation of opportunities for crime is nasty but necessary business." (internal citations omitted).[176]

The choice to use vulnerabilities without also simultaneously reporting them to the vendor is not precisely "the creation of opportunities for crime," but rather the choice not to pro-actively use opportunities to prevent crime. *Murphy* makes clear that this type of approach can be legally legitimate. Whether it is acceptable is a moral, policy, and political question.

Consider another approach, namely the Department of Justice's guidelines[177] on the use of confidential informants. These state that a Justice Law Enforcement Agent (JLEA) is never permitted to authorize a CI to "participate in an act of violence; participate in an act that constitutes obstruction of justice (e.g., perjury, witness tampering, witness intimidation, entrapment, or the fabrication, alteration, or destruction of evidence); participate in an act designed to obtain information for the JLEA that would be unlawful if conducted by a law enforcement agent (e.g., breaking and entering, illegal wiretapping, illegal opening or tampering with the mail, or trespass amounting to an illegal search); or initiate or instigate a plan or strategy to commit a federal, state, or local offense." [178] The guidelines do not state that a CI *must* work to prevent a crime from occurring. In the use of vulnerabilities the analogous situation would be that law enforcement is not required to let vendors know about the vulnerabilities they find and exploit.

Immediately reporting versus using for some time before reporting is a clash of competing social goods. That is what we need to weigh here. If our primary concern

---

[175] 642 F.2d 699 (2d Cir. 1980).
[176] *Id.* at 1529.
[177] Illegal activity must be authorized in advance for a period of up to ninety days.
[178] Department of Justice, GUIDELINES REGARDING THE USE OF CONFIDENTIAL INFORMANTS, January 8, 2001, http://www.justice.gov/ag/readingroom/ciguidelines.htm#suitability [last viewed February 23, 2013].

is preventing the proliferation of exploits, that makes a strong argument that society is better protected by reporting the vulnerability early even if that risks the ability of the criminal investigation to conduct its authorized wiretap.

We also note that the danger of proliferation means that each use of an exploit, even if it that has been successfully run previously, increases the risk that the exploit will escape the targeted device. As we know from other situations, whether rare diseases or the effect of cold weather on shuttle O-rings,[179] a rare side effect is more likely to appear when working with a larger population sample.

B. Ethical Concerns of Exploiting Vulnerabilities to Wiretap

Even though wiretaps are a long-accepted tool in the law-enforcement collection, there is something somewhat distasteful about using an exploit to download interception capability. Undoubtedly, part of that stems from the strong sense that vulnerabilities are to be patched, not exploited. But one thing even if law enforcement were never to report the vulnerabilities it discovers or purchases, law-enforcement's use of vulnerabilities would not make the vulnerability situation worse. Law enforcement is not currently a supplier of vulnerabilities to vendors. Thus, were law enforcement to use vulnerabilities and not report them to the vendors, there would be no change to the status quo ante. That said, there are some concerns raised by law enforcement 's use of vulnerabilities.

One danger of law-enforcement's participation in the zero-day market is the possibility of skewing the market, either by increasing incentives against disclosure of the vulnerability or by increasing the market for vulnerabilities and thus encouraging greater participation in it. Because of the size of the market and the relatively minimal need by law enforcement, we do not believe that this will be the case. Since the FBI has not discussed under what technical circumstances they have encountered difficulties wiretapping, it is hard to know exactly under which circumstances vulnerabilities will be used, but we do believe usage will be rare.

What is the government's responsibility in cases where the operationalized vulnerability does the wrong thing and escape the target? It is not unknown for physical searches to go amiss. Sometimes law enforcement executes a warrant on the wrong location, sometimes law enforcement executes a wiretap warrant on the wrong phone line.[180] Such a search would, of course, invalidate collection. But a

---

[179] Howard Berkes, *Reporting a Disaster's Cold, Hard Facts* (January 28, 2006, 1:27 pm), NPR, http://www.npr.org/templates/story/story.php?storyId=5175151 [last viewed March 12, 2013].

[180] See, for example, Intelligence Oversight Board Matter, [XXX] Division, Federal Bureau of Investigation, IOB Matter 2005-160, June 30, 2010. It is rare that such activity is publicly reported ("Documents Obtained by EFF Reveal FBI Patriot Act Abuses," March 31, 2011,

wiretap exercised through a operationalized payload changes the situation in a substantive way. Unlike an incorrectly executed wiretap warrant, which might simply collect information on the wrong party, the effect of a operationalized payload gone awry is worse; a badly designed payload could escape its target and potentially affect a much larger group of people.

If the operationalized software were to escape its target, it might be adapted for malicious purposes by others, a second-order effect that increases the need for great care in developing the systems. While the government may have some liability when it knocks down the wrong door in the course of exercising a search warrant,[181] with wiretap software the liability—in dollars or simply in costs to society— is less well understood.

It is critical that the tools employed by law enforcement be trustworthy and reliable. In particular, the technical implementation must capture exactly what is authorized. In addition, all the usual security provisions apply: the system must employ full auditing[182], each user of the system must log on individually, etc. Such careful controls have not always been exercised in the past, as is evidenced by flaws discovered in the FBI's DCS 3000 system[183] as well poor documentation of telephone transactional data requests during FBI investigations post-September 11th.[184] This argues for not only judicial oversight, but technical oversight as well.

Finally, one might imagine a scenario in which law enforcement puts pressure on vendors not to fix vulnerabilities so as to facilitate exploits. Aside from being bad public policy, such an approach would represent a dangerous poison pill for both government and industry. If such pressure became publicly known, the vendor would suffer serious reputational harm. It is not inconceivable that the vendor would also be liable for customer damages if the company knew of a serious vulnerability about which it had neither informed its customers nor patched.

---

https://www.eff.org/deeplinks/2011/03/documents-obtained-eff-reveal-fbi-patriot-act [last viewed March 3, 2013]).

[181] Jim Armstrong, "FBI Uses Chainsaw On Wrong Fitchburg Apartment," CBS Boston, January 31, 2012, http://boston.cbslocal.com/2012/01/31/fbi-uses-chainsaw-in-raid-on-wrong-fitchburg-apartment/ [last viewed March 3, 2013].

[182] This was missing in the Greek wiretapping case*; see* Prevelakis and Spinellis*, supra* note 5.

[183] The system was previously known as Carnivore. *See* Steven M. Bellovin, Matt Blaze, David Farber, Peter Neumann, and Eugene Spafford, Comments on the Carnivore System Technical Review (December 3, 2000), unpublished manuscript, http://www.crypto.com/papers/carnivore_report_comments.html [last viewed March 11, 2013).

[184] U.S. DEPARTMENT OF JUSTICE, OFFICE OF THE INSPECTOR GENERAL, OVERSIGHT AND REVIEW DIVISION, A REVIEW OF THE FEDERAL BUREAU OF INVESTIGATION'S USE OF EXIGENT LETTERS AND OTHER INFORMAL REQUESTS FOR TELEPHONE LETTERS (January 2010) at 46-47 and 70.

C. Technical Solutions to Preventing Proliferation

The principle of only harming the target must be a governing one for the use of vulnerabilities by law enforcement. One means of ensuring this is to employ technical mechanisms to restrict an exploit to a given target machine. The simplest forms check various elements of their environment when they run, e.g., the machine's serial number or MAC address; if they're on the wrong machine, they silently exit. Stuxnet[185] employed more or less this technique. A more sophisticated technique uses environmental data to construct a cryptographic key; if this isn't present, the data will not decrypt properly, and the code will not be comprehensible to any analyst. As noted,[186] the Gauss malware uses this technique; it has stymied top cryptanalysts for months.

From one perspective, the part of the exploit that contains the vulnerability is the most important piece, since knowledge of it will let people write their own exploit code. The best defense there is to use a dropper/payload architecture; that way, after the initial penetration, there's no further need for the vulnerability and the code relying on it can be deleted.

Promiscuous spread of penetration tools also increases the risk of proliferation. The more machines a piece of code is on, the more likely it is that someone will notice the code and reverse-engineer it. This would expose not just a carefully husbanded security hole, but also the surrounding infrastructure necessary to use it for lawful intercepts. This calculus is similar to that long used in the intelligence community: if one acts on intelligence, it risks giving away the source of information, which will then be unavailable in the future.[187] Here, though, there is the additional constraint of legal requirements against doing harm, harm that becomes more likely if malefactors discover the penetration tools.

VI. Reporting Vulnerabilities

With the CIPAV cases demonstrating[188] that the state employs vulnerabilities for searches and the like—the "can" problem—we turn to the "may" problem, namely should law enforcement do so?[189] We have already argued that the risks of extending CALEA to IP-based communications make that particular trade—the security provided by the extra surveillance versus the security risks created by introducing security breaches into network infrastructure and applications—a poor choice. As the vulnerability being used to introduce a wiretap already exists, the issue is somewhat different, and the question instead concerns patching. If a

---

185 See *Stuxnet*, *supra* footnote 17.
186 See Section IV.F; also see the footnotes describing Gauss.
187 See David Kahn, *The Codebreakers*, Macmillan, 1967. The theme pervades the book, but see especially the discussion of the assassination of Admiral Isoroku Yamamoto at 595.
188 Lynch, *supra* note 118.
189 We are indebted to Marty Stansell-Gamm for the phrasing of the "may" versus "can" problem.

vulnerability in a communications application or infrastructure is patched, the vulnerability cannot be exploited for a wiretap. But if the vulnerability is left unpatched, the result is that many are left open to attack. Thus the issue is not so much about introducing an exploit, but about when, and perhaps whether, to inform the vendor of the security problem.

What is law enforcement's responsibility with regard to reporting? We start by examining the security risks created by using vulnerabilities, then put that risk in the context of law-enforcement's role in crime prevention.

A. Security Risks Created by Using Vulnerabilities

As we have already noted in section V, there is a danger that even the most carefully crafted exploitation tools may not function as intended. There are at least two security concerns that must be weighed in choosing to use a vulnerability to conduct a wiretap: (i) the risk that the vulnerability's use will lead to overcollection, and (ii) the danger that the vulnerability will accidentally escape its target device and find use elsewhere.

Unfortunately there is much precedent for overcollection. Recent examples include the NSA's overcollection[190] as a result of the FISA Amendments Act[191] and the FBI's use of "exigent" letters to collect communications transactional data.[192] Use of the vulnerabilities requires close scrutiny by judges to ensure that what is collected is exactly what is to be collected, no more and no less. Judges will therefore need to evaluate just how intrusive a particular exploit may be, a technical as well as legal issue.

Law enforcement use of vulnerabilities poses at least two risks of unintended harm. First, the penetration tools may have unintended side-effects on the targeted system.[193] Second, the tools may somehow escape into the wild, harming innocent parties. Both are problematic and bear further examination.

The wiretap statute requires that taps be done "with a minimum of interference" with the service being monitored.[194] If an exploit causes other harm to the target computer, such as damaging files or applications or leading to frequent crashes, use

---

[190] James Risen and Eric Lichtblau, "E-Mail Surveillance Renews Concern in Congress," NEW YORK TIMES, June 16, 2009.
[191] P.L. 95-11, 92 Stat. 1782.
[192] U.S. Department of Justice, Office of the Inspector General, Oversight and Review Division, A REVIEW OF THE FEDERAL BUREAU OF INVESTIGATION'S USE OF EXIGENT LETTERS AND OTHER INFORMAL REQUESTS FOR TELEPHONE RECORDS, January 2010.
[193] Side effects could include disrupting other functionality, as occurred in the Greek wiretapping case (see Prevelakis and Spinellis, *supra* note 5).
[194] *See* 18 USC 2518(4).

of the exploit would violate this provision. At least one court has already quashed an eavesdropping order on these grounds:[195]

> Looking at the language of the statute, the "a minimum of interference" requirement certainly allows for *some* level of interference with customers' service in the conducting of surveillance. We need not decide precisely how much interference is permitted. "A minimum of interference" at least precludes total incapacitation of a service while interception is in progress. Put another way, eavesdropping is not performed with "a minimum of interference" if a service is *completely* shut down as a result of the surveillance.

(Emphasis in original.) It is worth noting that in this case, there were no allegations of instances of the customer trying and failing to use the service; however, use of the wiretap would make the original service unavailable to the customer if requested.

Apart from legal considerations, it is worth noting that interference can lead to discovery of the tap. This has happened at least twice in what appear to have been intelligence operations. In one, a very sophisticated wiretap operation mounted against a Greek cellphone operator, a bug in the attacking software caused some text messages not to be delivered. The resulting error messages led to discovery of the implanted code.[196] In a better-known case, the Stuxnet virus aimed at the Iranian nuclear centrifuge plant was discovered when some computer user became suspicious and sent a computer to a Belarussian antivirus firm for analysis.[197]

B. **Preventing Crime**

The question of when to report vulnerabilities that are being exploited is not new for the U.S government. In particular, the National Security Agency (NSA) has faced this issue several times in its history.

NSA performs two missions for the U.S. government: the well-known one of signals intelligence, or SIGINT, "reading other people's mail,"[198] and the lesser-known one of communications security, COMSEC, protecting U.S. military and diplomatic communications.[199] It has been an extremely useful to house the U.S. signals intelligence mission in the same agency as the U.S. communications security

---

[195] *Company v. United States* (in re United States) (2003, CA9 Nev) 349 F3d 1132.
[196] *See* Prevelakis and Spinellis, *supra* note 5, at 31.
[197] *See Stuxnet, supra* footnote 17.
[198] Henry Stinson, the Secretary of State who shut down the "Black Chamber," the Army's signals intelligence section during and after World War I, famously said, "Gentlemen do not read each other's mail." His views changed during World War II when he was Secretary of War; the U.S. relied heavily on signals intelligence during that conflict. Though the quote is attributed to Stinson, there is some evidence that he was acting on President Hoover's orders; *see* David Kahn, *The Reader of Gentlemen's Mail: Herbert Yardley and the Birth of American Codebreaking*, Yale University Press, 2004.
[199] The COMSEC mission is performed by the NSA's Information Assurance Division.

mission. Each is in a position to learn from the other. SIGINT's ability to penetrate certain communication channels could inform COMSEC's knowledge of potential weaknesses in our own; COMSEC's awareness of security problems in certain communications channels might inform SIGINT's knowledge of a target's potential weakness.

That's an "if only"; reality is in fact very different. COMSEC's awareness of the need to secure certain communications channels has often been thwarted by SIGINT's desire that patching be delayed so that it can continue to exploit traffic using the vulnerability in question. How this contradictory situation is handled depends primarily on where the vulnerable communications system is operating. If the insecure communications system is being used largely in the U.S. and in smaller nations that are unlikely to harm the U.S., then patching would not hurt the SIGINT mission. In that situation, COMSEC would be allowed to inform the vendor of the problem. In most other instances, informing the vendor would have been delayed so that SIGINT could continue harvesting product. Although this was never a publicly stated NSA policy, this modus operandi was a fairly open secret.

But law enforcement operates in a different domain than the military, and its considerations and values are different. The FBI concern that it is "going dark" is precisely on domestic wiretapping; law enforcement will want to exploit the vulnerabilities *exactly* when there are users in the U.S. Thus the balancing that NSA does between its SIGINT and COMSEC missions does not particularly illuminate what the state of affairs should be for the FBI. We must instead examine the issue from other vantage points.

One differentiator is the likelihood of collateral damage from using vulnerabilities. By their nature some vulnerabilities are easier to exploit than others. More critically, some (but not all) vulnerabilities are likely to be easier for law enforcement to exploit than for the general population of attackers to do so. Any attack that is aided by the ability to use compulsory legal process against a third party, such as an ISP, falls into this category. In these cases, failure to report the vulnerability to the vendor is less likely to have an effect on its exploitation by others.

There may also be a number of other factors that can also complicate launching an exploit, including knowledge of special information or material about the target. If such possession is necessary for the vulnerability to be exploited, then law enforcement can be fairly confident that there is little risk in not reporting the vulnerability to the vendor.

In considering whether to report, one might attempt to consider is how dangerous a particular vulnerability may be. Some aspects of the question are very easy to answer. If the vulnerability is in a network router or a switch, its impact is likely to be very large. Indeed, vulnerabilities in network infrastructure are fundamentally a national security risk because network devices are either ISP-grade gear, whose

compromise could be used to shut down or tap a large portion of the network, or enterprise gear, in which case compromise could be used for targeted espionage attacks, or else consumer gear, likely to be of wide usage and thus the compromise would effect a large population. Without question such vulnerabilities should be reported to the vendor immediately.

On the other hand, there are subtleties involved even if a vulnerability does not initially appear to be one that could create a national-security risk (per the issue just vulnerability just described). If the vulnerability is for an uncommon platform, it would seem that not informing the vendor of the problem is unlikely to create much risk. If the vulnerability is for an outdated version of a platform, depending on how outdated the platform is, the risk may also be relatively minor.[200] The latter is especially true for devices that are replaced frequently, e.g., smart phones. Yet it is often the case that outdated systems may be widely deployed in non-critical systems or deployed in critical systems.[201] So a vulnerability that applies to an outdated version of a platform may still be widely dangerous; it depends on exactly on who is using the platform and in what situation. This points to the complexity of determining when the situation is such that the vendor should be told about the vulnerability.

This raises the concern of whether the FBI will actually be able make such an evaluation. The ability to discern the potential risk from any particular vulnerability ranges from relatively trivial to quite difficult. One limitation is that the Domestic Communications Assistance Center (DCAC) will not be a cybersecurity vulnerability research center.[202] Nor should it be; that expertise lies in the NSA's Information Assurance Directorate, and duplicating the expertise is neither possible nor appropriate. Making such judgements would require vast knowledge about systems being employed in the U.S. across a wide array of industries. Even a decade after September 11th, this information is not being tracked by the U.S. government.

---

[200] This issue makes for an interesting insight into pirated software. The fact that a high percentage of software in China is illegally obtained has several implications for electronic surveillance. Probably the most significant is that the versions are not only out of date—e.g., as of January 2013, 64% of Chinese Windows users had Windows XP installed, while 32% had Windows 7 (StatCounter Global Stats, http://gs.statcounter.com/#os-CN-monthly-201202-201301 [last viewed February 17, 2013]) —but also less secure than more modern systems. Thus they are more open to exploitation.

[201] One example of this is Windows XP; the eleven-year-old OS is still the most common operating system in use at most government agencies (Shawn McCarthy, "8 reasons agency IT will change course in 2013," GCN, November 16, 2012, http://gcn.com/articles/2012/11/16/8-reasons-agency-it-will-change-course-in-2013.aspx [last viewed February 18, 2013]). Another is the backend systems supporting voting machines in Ohio (Patrick McDaniel, Kevin Butler, William Enck, Harri Hursti, Steve McLaughlin, Patrick Traynor, Matt Blaze, Adam Aviv, Pavel Cerny, Sandy Clark, Eric Cronin, Gaurav Shah, Micah Sherr, and Giovanni Vigna, "EVEREST: Evaluation and Testing of Election-Related Equipment, Standards, and Testing," Final Report, December 7, 2007, http://www.sos.state.oh.us/SOS/upload/everest/14-AcademicFinalEVERESTReport.pdf [last viewed February 18, 2013]).

[202] *See* Declan McCullagh, "FBI quietly forms secretive Net-surveillance unit", May 22, 2012, available at http://news.cnet.com/8301-1009_3-57439734-83/fbi-quietly-forms-secretive-net-surveillance-unit/.

Certainly the FBI is not in a position to know this, or to be able to make the determination about how dangerous to the U.S. a particular vulnerability may be.

The point is that except for some obvious cases,[203] it is usually very difficult to determine a priori whether a particular vulnerability is likely to create a serious problem. It may be that some obscure, but critical, part of society relies on the code with the vulnerability. It may be that it lies in some hidden part of the infrastructure; for example, for literally decades American Airlines relied on old software for planning flight operations.[204] Furthermore—and especially in an open-source world, where it may be impossible to determine all the users of a system—there is no way that law enforcement would be in a position to do a full mapping from software to users.

As we have alluded to earlier, this is a clash of competing social goods. There is the value of security obtained through patching as quickly as possible and the value of security by downloading the exploit to enable the wiretap to convict the criminal. Although there are no easy answers, we believe the answer is clear. In a world of great cybersecurity risk, where each day brings a new headline of the potential for attacks on critical infrastructure, where the Deputy Secretary of Defense says that thefts of intellectual property may be "may be the most significant cyberthreat that the United States will face over the long term,"[205] public safety and national security are too critical to take risks and leave vulnerabilities unreported and unpatched.We believe that law enforcement should always err on the side of caution in deciding to refrain from informing a vendor of a vulnerability. Any policy short of full and immediate reporting by default is simply inadequate. "Report immediately" is the policy that any crime-prevention agency should have, even though such an approach will occasionally hamper an investigation.[206]

Note that a "report immediately" policy does not foreclose exploitation of the reported vulnerability by law enforcement, Vulnerabilities reported to vendors do not result in immediate patches; the time to patch varies with each vendor's patch release schedule (once a month, or once every six weeks is common) but, since

---

[203] A striking example of one such occurred with the February 2013 US CERT alert concerning Java; the organization recommended disabling Java in web browsers until an adequate patch had been prepared (https://www.us-cert.gov/ncas/alerts/TA13-032A).

[204] Robert Mitchell and Johanna Ambrasio, "From build to buy: American Airlines changes modernization course midflight" (January 2, 2013), COMPUTERWORLD, https://www.computerworld.com/s/article/9234936/From_build_to_buy_American_Airlines_changes_modernization_course_midflight_ [last viewed March 11, 2013].

[205] William J. Lynn III, *Defending a New Domain*, FOREIGN AFFAIRS, 89, no. 5 (September/October 2010) at 102.

[206] There are persistent rumors that government agencies have sometimes pressured vendors to leave holes unpatched; *see, e.g.*,, "Microsoft gives zero-day vulnerabilities to US security services—Bloomberg", *Computing.co.uk*, June 14, 2013, available at http://www.computing.co.uk/ctg/news/2274993/microsoft-gives-zeroday-vulnerabilities-to-us-security-services-bloomberg. This is a very dangerous path, one that should not be followed by law enforcement agencies.

vendors often delay patches[207] the lifetime of a vulnerability is often much longer. Research shows that the average lifetime of a zero-day exploit is 312 days.[208] Furthermore, users frequently do not patch their systems promptly, even when critical updates are available.[209]

Immediate reporting to the vendor of vulnerabilities considered critical will result in a shortened lifetime for particular operationalized exploits, but it will not prevent the use of operationalized exploit. Instead, it will create a situation in which law enforcement is both performing criminal investigations using the wiretaps enabled through the exploits, and crime prevention through reporting the exploits to the vendor. This is clearly a win/win situation.

It is interesting to ponder whether the policy of "immediately report vulnerabilities" might have a positive impact on the zero-day industry. Some members of the industry, such as HP DVLabs, "will responsibly and promptly notify the appropriate

---

[207] On the second Tuesday of every month Microsoft issues patches both for software defects and vulnerabilities. This date is known as *'Patch Tuesday'.* Vendors who use a 6-week *'rapid-release cycle'* such as Google (Chrome) and Mozilla (Firefox, Thunderbird) frequently roll their security patches into their new releases. However, not all vulnerabilities discovered are patched in the next release, see http://www.pcworld.com/article/2033649/patch-tuesday-leaves-internet-explorer-zero-day-untouched.html and http://threatpost.com/oracle-leaves-fix-java-se-zero-day-until-february-patch-update-101712/ for some examples. Some vendors do issues patches considerably more rapidly; it is unclear, though, that this is always a good idea. Rapid patches often block a particular path to reach the underlying buggy code rather than repairing it. Accordingly, attackers often find new variants of the exploit without much trouble. Sometimes patches contain their own flaws. Thus there is likely an irreducible average minimum time.

[208] Zero-day vulnerabilities average a 10-month lifespan. See Bilge and Dumitras *An Empirical Study of Zero-day Attack in The Real World*, ACM Conference on Computer and Communications Security, Oct 2012.

[209] There is a paucity of peer-reviewed research results on how soon individual users apply patches. The best studies (*e.g.,* E. Rescorla, "Security holes... who cares." *Proceedings of the 12th USENIX Security Symposium*. 2003, or S.M. Bellovin, W.R. Cheswick, and A. Rubin, *Firewalls and Internet Security: Repelling the Wily Hacker,* second edition, at 275, Addison-Wesley, 2003) are old and apply to enterprise servers, not individual users. Enterprises have their own needs and dynamics for patching, such as compatibility with critical local software; furthermore, all system administration is generally under the control of a centralized support group. Most wiretaps are of individuals, especially drug dealers (*see* Wiretap Report*, supra* footnote 47); their behavior is likely very different. There have been a number of statements by industry consistent with our assertion (*e.g.,* "Survey Finds Nearly Half of Consumers Fail to Upgrade Software Regularly and One Quarter of Consumers Don't Know Why to Update Software", Skype press release, July 23, 2012, http://about.skype.com/press/2012/07/survey_finds_nearly_half_fail_to_upgrade.html). A recent study (Websense Security Labs Blog, "How are Java Attacks Getting Through?", March 25, 2013, available at http://community.websense.com/blogs/securitylabs/archive/2013/03/25/how-are-java-attacks-getting-through.aspx) is more useful, since it measures actual exposure of real-world web browsers. Only about 5% of users had up-to-date Java versions, despite warnings of ongoing attacks. The best evidence, though, is empirical: the prevalence of attacks against holes for which patches are available suggests that attackers still find them useful.

product vendor of a security flaw with their product(s) or service(s)."[210] Others, such as VUPEN, which "reports all discovered vulnerabilities to the affected vendors *under contract* with VUPEN"[211] (emphasis added), do not. Although it would be a great benefit to security if the inability to sell to law enforcement would cause the sellers to actually change policy, in point of fact, the U.S. law-enforcement market is unlikely to have a major impact on the zero-day market, which is international and dominated by national-security organizations.

---

[210] "The first attempt at contact will be through any appropriate contacts or formal mechanisms listed on the vendor Web site, or by sending an e-mail to security@, support@, info@, and secure@company.com with the pertinent information about the vulnerability. Simultaneous with the vendor being notified, DVLabs may distribute vulnerability protection filters to its customers' IPS devices through the Digital Vaccine service.

If a vendor fails to acknowledge DVLabs initial notification within five business days, DVLabs will initiate a second formal contact by a direct telephone call to a representative for that vendor. If a vendor fails to respond after an additional five business days following the second notification, DVLabs may rely on an intermediary to try to establish contact with the vendor. If DVLabs exhausts all reasonable means in order to contact a vendor, then DVLabs may issue a public advisory disclosing its findings fifteen business days after the initial contact." Zero Day Initiative, Disclosure Policy, http://www.zerodayinitiative.com/advisories/disclosure_policy/ [last viewed March 1, 2013].

[211] Vupen, Vupen Security Research Team, http://www.vupen.com/english/research.php [last viewed March 1, 2013].

C. A Default Obligation to Report

The tension between exploitation and reporting can be resolved if the government follows *both* paths, actively reporting and working to fix even those vulnerabilities that it uses to support wiretaps. As we noted, the reporting of vulnerabilities (to vendors and/or to the public) does not preclude exploiting them. Once a vulnerability is reported, there is always a lead time before a "patch" can be engineered, and a further lead time before this patch is deployed to and installed by future wiretap targets. Because there is an effectively infinite supply of vulnerabilities in software platforms,[212] provided the discovery enterprise finds new vulnerabilities at a rate that exceeds the rate at which they are repaired, reporting vulnerabilities need not compromise the government's ability to conduct exploits. By always reporting, the government investigative mission is not placed in conflict with its crime prevention mission. In fact, such a policy has the almost paradoxical property that the more active the law enforcement exploitation activity becomes, the more zero-day vulnerabilities are reported to – and repaired by – vendors.

However, this does not mean that a government exploitation laboratory will be naturally inclined to report the fruits of its labor to vendors. From the perspective of an organization charged with developing exploits, reporting might seem anathema to the mission, since it means that the tools it develops will become obsolete more quickly. Discovering and developing exploits costs money, and an activity that requires more output would need a larger budget. [213]

An obligation mandating that law enforcement agencies report any zero-day vulnerabilities they intend to exploit would thus have to be supported by a strong legal and policy framework. Such a policy would have to create bright lines for what constitutes a vulnerability that is required to be reported, when the report must occur, to whom the report should be made, and which parts of the government are required to do the reporting. There are many grey areas.

First, what would constitute a reportable vulnerability? Sometimes, this will be obvious. For example, some software bugs, such as input validation errors, might allow an attacker to take control over a piece of software. Such behavior is clearly an error. Once reported, the software vendor can easily repair the software to eliminate the vulnerability and "push" the correction out.[214] Other vulnerabilities are less clearly the result of specific bugs, however. In some cases, a vulnerability

---

[212] *See* Brooks, *supra* note 100.

[213] It is difficult to estimate precisely the cost of developing a particular vulnerability, but existing markets can serve as a guide here, as discussed in Section IV.

[214] Many, if not most, companies provide automatic security updates that are simply updated via the Internet.

results from overly powerful software features that might be behaving perfectly correctly as far as the software specification is concerned, but that allow an attacker to exploit them in unanticipated ways. For example, many email systems allow software to be sent as an "attachment" that is executed on the recipient's computer when the user clicks on it. If an attacker emails a user mailware and the user is persuaded however unwisely, to open it, the user's computer becomes compromised. Although it served as a vector for the malware, the email system software, strictly speaking, has behaved "correctly" here. The line between a "bug" and a "feature" is often quite thin.

Then there is the question of when a potential vulnerability that has been discovered becomes "reportable". Many vulnerabilities result from subtle interactions in a particular implementation,[215] and not every software bug results in an actual exploitable vulnerability. If the government is obligated to report exploitable vulnerabilities, when must it do so? A viable rule of thumb might be that once the government has developed an exploit tool, the underlying vulnerability has been confirmed to be exploitable and should promptly be reported. Note that this way of implementing "always report" gives law-enforcement investigators some lead time in using the exploit tool. This approach provides appropriate leeway for law enforcement to do its job (and not, for example, the job of quality assurance testers at a software company).

To whom should a vulnerability report be made? In many cases, there is an obvious point of contact: a software vendor that sells and maintains a product in question, or, in the case of open-source software, the community team maintaining it. In other cases, however, the answer is less clear. Not all software is actively maintained; there may be "orphan" software without an active vendor or owner to report to. And not all vulnerabilities result from bugs in specific software products. For example, standard communications protocols are occasionally found to have vulnerabilities,[216] and a given protocol may be used in many different products and systems. Here, the vulnerability would need to be reported not to a particular vendor, but to the standards body responsible for the protocol. Many standards bodies operate entirely in the open, which can make "quietly" reporting a vulnerability—or hiding the fact that it has been reported by a law enforcement agency—problematic.

---

[215] Quite some time ago, one of the authors of this paper discovered that someone working on an important project was one of three people who were arrested in a hacking incident. (He eventually pled no contest. One of the other two was convicted; the third was acquitted.) An audit of the code base was performed. The team found one clear security hole, but log files showed it was an inadvertent hole coded, ironically, by one of the other auditors. The other problem found was more subtle. There were two independent bugs, for one of which the comments didn't agree with the code. Either bug alone was harmless; both together, combined with a common configuration mistake, added up to a remote exploit. There was a plausible innocent explanation for why the comments and the code didn't match. It remains unclear if this was a deliberate back door or a coincidence.

[216] For example, several vulnerabilities have been found that allow attacks against systems using the Secure Socket Layer (SSL) protocol, a widely used standard employed by many applications, including Web browsing, printing, and email, for encrypting Internet connections.

Finally, there is the question of who in the government would be covered by the reporting policy. In this paper, we are concerned specifically with a law enforcement vulnerability lab. Would every US government employee be covered by the policy? Or only those developing law enforcement surveillance tools? The vast majority of government employees—even those who encounter security vulnerabilities—aren't directly involved in developing wiretapping tools. For example, there are presumably system administrators in the Veterans Administration who occasionally discover security vulnerabilities in the course of their work. Would they become legally obliged to report? We propose that the reporting obligation be linked to the use of vulnerabilities for law enforcement purposes. An ordinary system administrator who discovered a hole perhaps should report it; the legal requirement, though, would apply to those who employ such holes to conduct communications intercepts.

VII. Policy and Legislative Issues

When should reporting occur, at the time of discovery or purchase of the vulnerability, or at the time of working exploit? Might there be exceptions to the reporting rule in the case of an extremely important target, and how that might work? In this section, we attempt to answer these questions as well as discuss the role of oversight.

A. Enforcing Reporting

We advocate that vulnerabilities law enforcement seeks to exploit to be reported by default. There are a number of ways to implement and enforce such a policy.

The simplest would be for an executive branch policy that mandates reporting under certain circumstances. Such a policy would come from the administration, likely through the Department of Justice. However, a policy-only approach has inherent weaknesses. First, the policy would be formulated, implemented, and enforced by the very agency with the most interest in creating exceptions to the rule, and that most "pays the cost" of neutralizing the tools it develops and uses. Such conflicts of interest rarely end up with the strongest possible protections for the public.

Therefore, a legislative approach may be more appropriate. Perhaps as part of the appropriation that funds the exploit discovery effort, Congress could mandate that any vulnerabilities it discovers be reported. As noted above, such legislation would need to be carefully drafted to capture a range of different circumstances.

In many situations, the best solution is for the judge authorizing the use of the vulnerability to insert a reporting requirement into the warrant or order. This provision could include a return date by which the requesting agency must certify

that the vendor had received appropriate notification. Apart from providing an enforcement mechanism, this approach allows for careful consideration of specific circumstances, including exceptional circumstances that might merit a delay.[217]

Finally, one might imagine that the courts would recognize an obligation for the government to report vulnerabilities, and create a tort cause of action for those harmed by a criminal exploitation of a vulnerability known to the government but not reported. This would be perhaps the most radical approach to ensuring government reporting, but it seems most unlikely. There is, currently, no obligation on anyone to report vulnerabilities; for a court to suddenly discover one seems improbable.[218] Thus for early government reporting of vulnerabilities discovered under this program, a legislative mandate that the government report any zero-day vulnerabilities it seeks to exploit seems the best approach.[219]

B. Exceptions to the Reporting Rule

Although we have recommended that law enforcement report vulnerabilities upon discovery (or purchase), there may be exceptional cases when immediate reporting is not appropriate. Immediate reporting of the vulnerability might lead to patching and prevent achieving a wiretap. Might there be circumstances in which not reporting is appropriate?

Consider the closely related established practice of emergency wiretaps. Title III includes an exception allowing wiretaps to be used in emergency situations without a warrant so long as a wiretap order is obtained within forty-eight hours.[220] The law states that an emergency situation exists when there is immediate danger of death or serious bodily injury, conspiratorial activities threatening national security, or conspiratorial activities characteristic of organized crime,[221] but practice is that warrantless wiretapping by law enforcement[222] is permitted only when there is an immediate threat to life such as kidnapping and hostage-taking situations.[223]

---

[217] Exceptional circumstances are discussed in the following section.

[218] Due in part to disclaimers in End User License Agreements (EULAs), there is in general no liability even for vendors or developers of insecure software; *see, e.g.,* Michael D. Scott, "Tort Liability for Vendors of Insecure Software: Has the Time Finally Come?", 67 Md. L. Rev. 425. (2008); however, the issue is a frequent topic of academic discussion and the situation could conceivably change. In some situations, a site operator can be held negligent, *i.e.,*, *In Re Heartland Payment Systems*, 851 F.Supp.2d 1040 (United States District Court, S.D. Texas, Houston Division.2012).

[219] We do not discuss or suggest remedies if the government fails to report vulnerabilities, as urged in this paper. A radical legislative approach would permit damages for those harmed by the exploitation of a zero-day vulnerability that was known to the government but that the government had not reported. A more moderate approach would legislate the government's reporting obligation but disallow private recovery of damages if it fails to do so.

[220] 18 U.S.C. § 2518(7).

[221] 18 U.S.C. § 2518(7).

[222] Note that we are discussing warrantless wiretaps for criminal investigations under Title III, not the legalities of the Bush administration's "terrorist surveillance" warrantless wiretapping program.

[223] For a detailed discussion, see US ATTORNEYS MANUAL, 9-7.112 Emergency Interception, http://www.justice.gov/usao/eousa/foia_reading_room/usam/index.html.

Emergency wiretapping is not done lightly, and requires approval of no rank lower than an Associate Attorney General. Once the emergency wiretap is approved—approved, not installed—law enforcement has forty-eight hours to obtain a wiretap order.[224]

Consider now the subject of a wiretap warrant, one for whom normal methods of interception are unlikely to succeed. Using a wiretap warrant, law enforcement downloads software to the target's machine that reports back what programs and operating system are being run on the device. The target is running an unusual set of programs, e.g., using the OpenBSD operating system with the Lynx web browser.[225] Law enforcement lacks suitable tools for this particular set up. To exercise the actual wiretap, law enforcement must find a vulnerability, and operationalize it. As we discussed earlier, doing so will take between two to seven days. If the vulnerability is immediately reported as soon as it is acquired, law enforcement runs the risk that the target's device may be patched before the operationalized exploit can be used.

We can infer from the FBI's use of CIPAV that there is currently no legal or policy requirement that law enforcement report vulnerabilities. So we recommend a compromise. For public safety, the law should require that law enforcement report vulnerabilities to the vendor once they have been acquired or otherwise discovered. But there should also be an emergency exception similar to that of Title III. We recommend that in an emergency situation, law enforcement should have a forty-eight hour window in which it could petition for a release from reporting the vulnerability until it had successfully installed a wiretap.

We expect that such a provision would be only very rarely invoked. First, most vulnerabilities will have been discovered and reported by law enforcement, and the tools that exploit them built and put in the arsenal for future use, well before there is any case that might use them. For such tools, there is no emergency—or even any case —to weigh against reporting at the time the vulnerability would be reported. Any cases in which a vulnerability is used would come up long after the vulnerability has already been reported.

But there may be exceptional circumstances in which this pattern—vulnerabilities discovered and tools developed well in advance of the cases where they are used—is not followed. For example, we can imagine a very high-value organized crime case in which a target might be using a particular and well-hardened, non-standard platform for which no exploit tools are available in the "standard" arsenal. Law enforcement might devote targeted resources toward discovering vulnerabilities

---

[224] 18 U.S.C. § 2518(7)

[225] OpenBSD is an open-source operating system based on Unix; Lynx is a web browser. (Because Lynx does not support graphics, it cannot have web bugs, embedded objects that track usage, making it particularly privacy protective.) Both systems, which relatively old by industry standards, continue to be developed, but neither has large market share.

and developing tools for the specific devices used by the particular target. In such (likely very rare) cases, the case and target would might known at the time some vulnerability is discovered by law enforcement, and they might place a high priority on preserving their ability to exploit it during the case.

The criteria for exemption must be as stringent as the Title III exemption. If emergency wiretaps are permitted only when there is imminent danger of death — e.g., a kidnapping or hostage-taking situation—then the situation for emergency use of a vulnerability without reporting must be equally dire. Note that even terrorist investigations do not generally employ emergency wiretap provisions; neither should they employ an emergency exemption to vulnerability reporting.

The other issue in emergency use is that the vulnerability must be such that there is a low risk of serious harm resulting from its exploitation by others against innocent persons. As we have discussed, estimating such risk is quite difficult. Given the importance of preventing crime, the decision not to report must not be made lightly. Indeed, the "default" presumption must be that a vulnerability should be reported, with exceptions made only for unusual and compelling reasons. The petition not to report must include not only an argument for the importance of the interception but also an analysis of the harm likely should the vulnerability be discovered and exploited by others during the period that law enforcement is operationalizing the tool. In weighing whether to delay reporting a vulnerability, the court should consider how likely it is that the vulnerability, having been discovered, can actually be exploited, and the damage that may result from such exploitation.

C. Providing Oversight

There is the danger that an operationalized exploit may proliferate past its intended target. Stuxnet[226] provides an interesting case in point. Although aimed at Iran, the malware spread to computers in other countries, including India and Indonesia.[227] It is unclear from the public record how this happened. It may have been due to a flaw in the code, as Sanger contends;[228] alternatively, it may have been foreseeable but unavoidable collateral damage from the means chosen to launch the attack against Iran. Either option, though, represents a process that may be acceptable for a military or intelligence operation but is unacceptable for law enforcement. Only the legally authorized target should be put at risk from the malware used.

---

[226] *See Stuxnet, supra* footnote 17.

[227] David Sanger, CONFRONT AND CONCEAL: OBAMA'S SECRET WARS AND THE SURPRISING USE OF AMERICAN POWER, Crown Publishers, 2012, at 203-205.

[228] *Id*. Sanger's conclusion is somewhat controversial; *see* Steven Cherry, "Stuxnet: Leaks or Lies?", *IEEE Spectrum* podcast, September 4, 2012, available at http://spectrum.ieee.org/podcast/computing/embedded-systems/stuxnet-leaks-or-lies.

Given the policy issues raised by the use of vulnerabilities, it would be appropriate to have public accountability on the deployment of this technique. We have in mind annual reports on vulnerability use similar to the AO's Wiretap Reports, presenting such data as how many vulnerabilities were used by law enforcement were used in a given year, whether by federal or state and local. Was the vulnerability subsequently patched by the vendor, and how quickly after being reported? Was the vulnerability used by others? Did the operationalized vulnerability spread past its intended target? Was the vulnerability exploited outside law enforcement during the period that law enforcement was aware of the problem but had not yet told the vendor? What damages occurred from its exploitation? Making such information open to public analysis should aid in decisions about the right balances being struck between efficacy and public safety.[229]

D. Regulating Vulnerabilities and Exploitation Tools

As we have mentioned, even without considering its use by law enforcement, information about software vulnerabilities is inherently "dual use"—useful for both offense and defense. Related to the issue of reporting and proliferation is the question of how the law should treat information about vulnerabilities and the development of software tools that exploit them by non-law enforcement persons. Should information about vulnerabilities, and tools that exploit them, be restricted by law? How do existing statutes treat such information and tools?

The issue of how to handle such dual-use technologies is not new. The computer security community has grappled for years with the problem of discouraging illicit exploitation of newly discovered vulnerabilities by criminals while at the same time allowing legitimate users and researchers to learn about the latest threats, in part to develop effective defenses.[230] It is all but impossible to prevent information about vulnerabilities or software exploits that use them from getting in to the hands of criminals without hampering efforts at defense. On the one hand—perhaps most straightforwardly—information about zero-day vulnerabilities is coveted by criminals who seek unauthorized and illicit access to the computers of others. But the same zero-day information is also used, and sought out by, legitimate security

---

[229] The same is true regarding data from the Administrative Office of the US Courts, WIRETAP REPORT. For example, one of the authors of the present paper used the WIRETAP REPORT data to show that FBI claims about the importance of wiretaps in solving kidnappings was incorrect. Between 1969 and 1994 that wiretaps were used in only two to three kidnappings a year (out of 450 kidnappings annually) (Whitfield Diffie and Susan Landau, PRIVACY ON THE LINE: THE POLITICS OF WIRETAPPING AND ENCRYPTION, MIT Press, 2007, at 211).

[230] The question of the ethics of publishing vulnerability information far antedates computers. In 1857, Alfred Hobbs, in *Rudimentary Treatise on the Construction of Door Locks,* wrote "A commercial, and in some respects a social, doubt has been started within the last year or two, whether or not it is right to discuss so openly the security or insecurity of locks. Many well-meaning persons suppose that the discussion respecting the means for baffling the supposed safety of locks offers a premium for dishonesty, by showing others how to be dishonest. This is a fallacy. Rogues are very keen in their profession, and already know much more than we can teach them respecting their several kinds of roguery."

researchers and computer scientists who are engaged in building defenses against attack and in analyzing the security of new and existing systems and software. Even software tools that exploit vulnerabilities are inherently dual use. They can be used by criminals on the one hand, but are also useful to defenders and researchers. Computer and network system administrators routinely use tools that attempt to exploit vulnerabilities to test the security of their own systems and to verify that their defenses are effective. Researchers who discover new security vulnerabilities or attack methods often develop "proof of concept" attack software to test and demonstrate the methods they are studying. It is not unusual for software that demonstrates a new attack method to be published and otherwise made freely available by academics and other researchers. Such software is quite mainstream in the computer science research community.[231]

The software used by malicious, criminal attackers to exploit vulnerabilities can thus be very difficult to meaningfully distinguish from mainstream, legitimate security research and testing tools. It is a matter of context and intent rather than attack capabilities *per se*, and current law appears to reflect this.

Current wiretap law does not generally regulate inherently dual-use technology. The provision of Title III concerned with wiretapping equipment, 18 USC § 2512, generally prohibits possession and trafficking in devices that are "primarily useful" for "surreptitious interception"[232] of communications, which does not appear to

---

[231] Many security software packages that might appear to be criminal attack tools are actually designed for legitimate research and testing. For example, the *Metasploit* package [http://metasploit.com] is a regularly updated library of software that attempts to exploit known vulnerabilities in various operating systems and applications. Although it may appear at first glance to be aimed at criminals, it is actually intended for (and widely used by) system administrators and professional "penetration testers" to identify weaknesses that should be repaired in their systems.

[232] 18 USC § 2512 (1) provides criminal penalties for any person not otherwise authorized who:
(a) sends through the mail, or sends or carries in interstate or foreign commerce, any electronic, mechanical, or other device, knowing or having reason to know that the design of such device renders it primarily useful for the purpose of the surreptitious interception of wire, oral, or electronic communications;
(b) manufactures, assembles, possesses, or sells any electronic, mechanical, or other device, knowing or having reason to know that the design of such device renders it primarily useful for the purpose of the surreptitious interception of wire, oral, or electronic communications, and that such device or any component thereof has been or will be sent through the mail or transported in interstate or foreign commerce; or
(c) places in any newspaper, magazine, handbill, or other publication or disseminates by electronic means any advertisement of—
(i) any electronic, mechanical, or other device knowing or having reason to know that the design of such device renders it primarily useful for the purpose of the surreptitious interception of wire, oral, or electronic communications; or
(ii) any other electronic, mechanical, or other device, where such advertisement promotes the use of such device for the purpose of the surreptitious interception of wire, oral, or electronic communications, knowing the content of the advertisement and knowing or having reason to know that such advertisement will be sent through the mail or transported in interstate or foreign commerce,

apply to a wide range of current software exploit tools developed and used by researchers. We believe this is as it should be. The security research community depends on the open availability of software tools that can test and analyze software vulnerabilities. Prohibiting such software generally would have a seriously deleterious effect on progress in understanding how to build more secure systems, and on the ability for users to determine whether their systems are vulnerable to known attacks. In addition, we note that given that majority of vulnerability markets are outside the U.S., and that national-security agencies are heavy purchasers of these vulnerabilities,[233] regulating them is not a plausible option.

The specialized tools developed by law enforcement to collect and exfiltrate evidence from targets' computers, however, might fall more comfortably under the scope of 2512 as it is currently written. These tools would not be developed to aid research or test systems, but rather to accomplish a law-enforcement interception goal. They would have narrowly focused features designed to make their installation surreptitious and their ongoing operation difficult to detect. They would also have features designed to identify and collect specific data, and would have no alternative use outside the surreptitious interception application for which they were developed. Such tools, unlike those used by researchers, could more easily meet 2512's test of being "primarily useful" for "surreptitious interception".

[233] Greenberg, *supra* note 165.

VIII. Conclusions

Changes in telecommunications technologies led to the 1994 passage of CALEA. However, CALEA created problems because of software complexity and the fact that it introduces a security vulnerability. Due to further—and quite extraordinary—changes in the communications technologies since CALEA's passage, the law-enforcement wiretapping capabilities the law engendered are now in danger of failing; law enforcement now seeks to expand the CALEA regime to IP-based communications. As we have discussed, the changes in communications technologies since 1994 not only undermine the present version of CALEA, they make extending the CALEA model to modern communications systems highly problematic, creating serious security risks.

Nonetheless there needs to be a way for law enforcement to execute authorized wiretaps. The solution is remarkably simple. Instead of introducing *new* vulnerabilities to communications networks and applications, in the cases where wiretapping is difficult to achieve by other means, law enforcement should use of vulnerabilities already present in the target's communications device to wiretap. The use of vulnerabilities to accomplish legally authorized wiretapping creates uncomfortable issues. Yet we believe the *technique is preferable for conducting wiretaps against targets when enabling other methods of wiretapping, such as by deliberately building vulnerabilities into the network or device, would result in less security*.

We propose specific policies to limit the potential damage. First, we recommend that in order to prevent rediscovery of the vulnerability and hence proliferation of the exploit, technical defenses should be implemented. Second, we recommend that, with rare exceptions, *law enforcement should report vulnerabilities on discovery or purchase.* This means our proposal may actually have the benefit of *increasing* security generally. Finally, because the exploit may allow far greater penetrations of the target device than would be permitted by a mere wiretap, we urge guidelines to ensure that law enforcement bar use of any other information found on the computer during the exploit (unless permitted by an additional warrant).

There is a critical difference in the societal dangers entailed in the use of targeted vulnerabilities compared with the installation of global wiretapping capabilities in the infrastructure. If abused, targeted vulnerability exploitation, like wiretapping in general, has the potential to do serious harm to those subjected to it. But it is significantly more difficult – more labor intensive, more expensive, and more logistically complex – to conduct targeted exploitation operations against all members of a large population. In other words, although vulnerability exploitation is very likely to be effective against any given target, it is difficult to abuse at large scale or in an automated fashion against *everyone*. Thus our solution provides

better security than extending the model of CALEA to IP-based would.

Vulnerability exploitation has more than a whiff of dirty play about it; who wants law enforcement to be developing and using malware to break into users' machines? We agree that this proposal is disturbing. But as long as wiretaps remain an authorized investigatory tool, law enforcement will press for ways to accomplish electronic surveillance even in the face of communications technologies that make it very difficult. We are at a crossroads where the choices are to reduce everyone's security or to enable law enforcement to do its job through a method that appears questionable but that does not actually make us less secure. In this debate, our proposal provides a clear win for both innovation and security.