ELECTRONIC FRONTIER FOUNDATION

Coders' Rights Project Vulnerability Reporting FAQ

Looking for instructions on reporting security vulnerabilities in EFF software or systems? Click here.

Table Of Contents

Just as important as discovering security flaws is reporting the findings so that users can protect themselves and vendors can repair their products. There are many outlets for publicly reporting vulnerabilities, including mailing lists supported by universities and by the government. Unfortunately, researchers have received legal threats from vendors and government agencies seeking to stop publication of vulnerability information or “proof of concept” code demonstrating the flaw. This FAQ sets forth some ways that security researchers can reduce their legal risk when reporting vulnerabilities.

What Is This FAQ And Who Is It For?

This FAQ is intended for non-lawyers who want some general information about how U.S. laws might affect vulnerability reporting by security researchers. This information is provided as a general guide to the legal issues, but is not legal or technical advice.

The legal questions raised by vulnerability reporting can be complex and the legal risks in any case will depend on the particular facts and on legal doctrines not necessarily mentioned here. This FAQ is meant to familiarize you with some of the principles involved, so that you can have a more effective discussion if and when you engage an attorney to help you with your particular circumstance.

Feel free to contact EFF if you need help finding a lawyer qualified to advise on vulnerability reporting.

What Are the Policy Considerations Affecting Opinions About Vulnerability Reporting? ^

One of the more vigorous public policy debates in the security field involves the publication of information about security vulnerabilities. On the one hand, public disclosure of security information enables informed consumer choice and inspires vendors to be truthful about flaws, repair vulnerabilities and build more secure products. Disclosure and peer review advances the state of the art in security. Researchers can figure out where new technologies need to be developed, and the information can help policymakers understand where problems tend to occur.

On the other hand, vulnerability information can give attackers who were not otherwise sophisticated enough to find the problem on their own the very information they need to exploit a security hole in a computer or system and cause harm.

Vulnerability reporting is part of a broader debate about the potential harms and benefits of publishing information that can be used for dangerous purposes, but software security disclosures are a special case because vulnerability reports may include proof of concept code, a very specific way of explaining a security flaw to other coders and researchers. Proof of concept code can be particularly problematic because it is both descriptive and functional and can be used to or modified to create a program that will use the vulnerability to gain unauthorized access or otherwise interfere with the computer system.

Many security researchers have voluntarily adopted a delayed publication policy often called "responsible disclosure". While the details differ, the term has come to mean that the researcher discloses full information to the vendor, possibly discloses some information – but not proof of concept code – to the public, and refrains from publishing details that would allow an attacker to exploit the security flaw until the vendor issues a patch. In return, the vendor is supposed to expeditiously issue a fix and give credit to the researcher for his or her discovery.

Problems persist. Disclosure or the threat of disclosure often encourages quick patching, but when vendors do not act quickly to issue patches, the researcher may reasonably believe that the responsible thing to do is to disclose the problem so that customers can protect themselves. Vendors may have strong economic incentives to downplay or misrepresent risks, incentives that disclosure counteracts. Vendors may have contractual relationships with security firms that inhibit disclosure of important security information. Criminals disinterested in improving security may refuse to report security information to vendors or the public, so that the flaw will not be fixed and can be secretly exploited for economic or political gain.

This FAQ does not endorse any particular view of when disclosure is responsible. EFF believes that security researchers have a First Amendment right to report their research and that disclosure is highly beneficial. It is a highly subjective question of when and how to hold back details to mitigate the risk that vulnerability information will be misused, so the law should only rarely police disclosures.

That having been said, reporting that conforms with commonly accepted best practices is less likely to draw legal fire. However, disclosures outside of the "responsible disclosure" model may be both responsible and legal. Conversely, responsible disclosure may not protect you from being sued.

What Aspects Of Vulnerability Reports Are Most Legally Risky?

- The more detailed the advisory, the more risky it is. Ask how much the details in an advisory would aid a potential attacker.
- The more functional code an advisory contains, the more risky it is. Ask whether the code in a disclosure can be compiled into a tool that will exploit the vulnerability.

- The more likely the audience is to use the information to break the law, the more risky the publication. Ask whether you are disseminating the information to the general public, to a trusted group, or to a group considered likely to use the information for illegal purposes.
- If the security flaw relates to digital rights management tools or other technological "locks" that control access to copyrighted works (e.g., authentication handshakes, protocol encryption, password authentication, code obfuscation, code signing) controlling access to copyrighted works, the advisory is more risky. Ask (a lawyer) whether the publication might be restricted by the anti-circumvention provisions of the Digital Millennium Copyright Act (DMCA).
- There are no "whistleblower" protections under the applicable laws for security researchers. If the publication violates the law, or is proof of illegal research activities, it is not a defense that the information obtained and reported was important for public safety. The fact that the research was reported in a way that promotes public safety rather than criminal activity, however, may contribute to a finding that the report did not violate the law.

Don't miss our section on *How to Limit Legal Risk*.

What Legal Doctrines Are Most Likely To Affect Vulnerability Reporting?

The following areas of United States law are particularly relevant for security researchers reporting vulnerabilities:

- The First Amendment of the United States Constitution;
- Copyright law;
- Trade secret law;
- Patent law;
- The anti-circumvention provisions of the Digital Millennium Copyright Act (DMCA), codified at 17 U.S.C. section 1201;
- Contract law, if use of the software is subject to an End User License Agreement (EULA), Terms of Service notice (TOS), Terms of Use notice (TOU), Non-Disclosure Agreement (NDA), developer agreement or API agreement;
- Criminal laws including conspiracy and aiding and abetting.

Could International Laws Affect Vulnerability Reporting?

International laws may be more restrictive than United States law because other countries generally lack the speech protections of the U.S. Constitution's First Amendment. For example, the Council of Europe's Cybercrime Treaty requires signatories to impose criminal penalties for the production, sale, import and distribution of a device or program designed or adapted primarily for the purpose of committing unauthorized access or data interceptions. Countries may – but are not required to – exempt tools possessed for the authorized testing or protection of a computer system.

This FAQ does not address international law.

How Might The First Amendment Protect Security Vulnerability Reporting?

Publication of truthful information is protected by the First Amendment. Both source code and object code are also protected speech. Therefore truthful vulnerability information or proof of concept code are constitutionally protected.

This protection, however, is not absolute. Rather, it means that legal restrictions on publishing vulnerability reports must be viewpoint-neutral and narrowly tailored. Practically speaking, this means it is very rare for the publication of non-code information lead to legal liability. For example, a researcher who shares vulnerability information with people he knows will use the information for criminal purposes may be illegal.

In contrast, many regulations of code, including copyright law and regulation of circumvention tools under the DMCA, are acceptable under the First Amendment. Courts have held that the functional aspects of code, which may make it easy for others to perform illegal acts, justify restrictions on the development, use, and distribution of certain kinds of computer programs. That is why publishing code, though highly informative for computer scientists, brings more legal risk.

As a result, the law may punish a researcher for obtaining a copy of software improperly (i.e. through copyright infringment or trade secret misappropriation) or for using illegal research methodologies, but only rarely will even these improper means result in additional punishment for disseminating *non-code* research results. But if a code or non-code publication violates the law—because it reveals protected trade secrets or because it contains copyrighted code, for example—then even legitimate research protocols will not be a defense.

How Might Copyright Law Limit My Ability To Report Vulnerabilities?

Copyright law generally grants a certain set of exclusive rights to copyright owners, including the right to make copies of copyrighted works. Software is one category of works that are protected by copyright. As a result, if you make copies of software, you generally need either permission from the copyright owner or an exception granted by the copyright laws. The most relevant exception is the fair use doctrine, which allows users to make unauthorized copies in certain circumstances. Information about vulnerabilities in a software program do not implicate copyright law. However proof of concept code that copies the original program is infringing unless the copying constitutes a fair use under copyright law.

How Might Trade Secret Law Limit Vulnerability Reporting?

Like copyright infringement, misappropriation of trade secrets can be both a civil and criminal offense. Generally, a trade secret is information that (1) derives independent economic value, actual or potential, from not being generally known to the public or to other persons who can obtain economic value from its disclosure or use; and (2) is the subject of efforts that are reasonable under the circumstances to maintain its secrecy. Misappropriation means a wrongful acquisition, use, or disclosure of a trade secret.

Security research performed through reverse engineering and not subject to any contractual restrictions generally doesn't violate trade secret law because it is a fair and independent means of learning information, not a misappropriation. Once the information is discovered in a fair and independent way, it can be reported without violating trade secret law.

However, reverse engineering that violates an NDA or other contractual obligation not to reverse engineer or disclose[1] may be misappropriation. Reporting those findings would also be prohibited. Breaking a promise made in a negotiated NDA is more likely to result in a trade secret claim than would violating a term in a mass market EULA. (Although that could be a breach of contract).

One undecided legal question is whether a security flaw could fit the definition of trade secret. The flaw is not valuable to the software vendor, but the software program or computer service may be more valuable when the public is ignorant about the security problem. Vendors have argued that vulnerability information is their protected trade secret information. For example, in 2003, Blackboard, a door access control company obtained a temporary restraining order preventing two students from disclosing security flaws in the company’s locks based on a trade secret allegation. The suit eventually ended in a settlement, and it appears that the court never had the benefit of adversarial briefing from both parties on the question of whether the vulnerability information was in fact a trade secret. The Blackboard case did not result in a ruling with any value as precedent, but is one example of how a company seeking to control bad press about its product might try to use trade secret law to suppress vulnerability information.

How Might Patent Law Restrict Vulnerability Reporting?

A patent is a set of exclusive rights granted by to an inventor or his assignee for a fixed period of time in exchange for a disclosure of an invention. To infringe a patent, one must make, use, sell or offer for sale an invention described by the patent's claims without the patent owner's authorization.

In 2007, HID, a manufacturer of access-control devices used a patent infringement threat to force Chris Paget, a researcher with IOActive to pull a conference presentation on security flaws in RFID cards. In demonstrating RFID insecurity, the researchers created a homebrew RFID reader that arguably worked the same way as HID’s patented reader. HID claimed that demonstrating the hack would infringe the patent and the researcher decided not to present his research.

No court ever approved this argument, but if accepted, it has a potentially broad effect on vulnerability reporting.

How Might The Anti-Circumvention

Restrictions Limit Vulnerability Reporting?

The anti-circumvention provisions of the DMCA, 17 U.S.C. 1201, prohibit circumvention of "technological protection measures" that effectively control access to copyrighted works. The law also prohibits trafficking in tools that are primarily designed for circumvention, have only limited commercially significant purpose other than circumvention or are marketed for circumvention. Vulnerability reports that do not include "tools", i.e. code, do not run afoul of section 1201 (but the research activities underlying that report might. See the Reverse Engineering FAQ for more information). Proof of concept or exploit code may be prohibited under section 1201 if it otherwise fits the definition of a circumvention tool.

Section 1201 has been used on several occasions to threaten security researchers, but no court has ever approved of the claim in this context. For examples, see EFF's Unintended Consequences: Seven Years under the DMCA.

While section 1201 can arguably apply to any security researcher, those studying digital rights management (DRM) of music, movies or other creative content are most likely to face section 1201 claims, since Congress intended to protect these copyrighted works when it passed the statute. Researchers looking for vulnerabilities in authentication handshakes, code signing, code obfuscation, and protocol encryption also have to worry about section 1201 because vendors have argued that these also qualify as "technical protection measures" covered by the DMCA.

Congress recognized that the anti-circumvention provisions could interfere with security research, so it included three exceptions that permit reverse engineering, encryption research and security research under very narrow circumstances. A researcher must jump additional hurdles to distribute code (tools) derived from the very limited research activities allowed by the statute. For example:

- A circumvention tool created as a result of permitted reverse engineering may only be distributed for the sole purpose of enabling interoperability of an independently created computer program with other programs, and only to the extent that doing so is non-infringing and does not violate other laws.
- A circumvention tool created as a result of permitted encryption research may only be distributed to someone with whom the researcher is working collaboratively for the purpose of conducting good faith encryption research or for the purpose of having that other person verify

his or her acts of good faith encryption research.

- A circumvention tool produced as a result of permitted security testing may only be distributed or employed for the sole purpose of security testing with the authorization of the owner operator of a computer system or network, provided such technological means does not otherwise violate section 1201 or the Computer Fraud and Abuse Act.

In sum, section 1201 potentially poses a serious restriction to security researchers and vulnerability reporters, particularly in the areas of DRM or other technological protection measure. If your research is in this area, you should consult a lawyer early in the course of your research.

How Could Contract Law Limit Vulnerability Publication?

Most software today comes with EULAs, and EULAs sometimes have "no reverse engineering" clauses. Websites or other internet services also may TOS or TOU that purport to restrict otherwise legal research activities. Researchers and programmers may receive access to code pursuant to an NDA, developer agreement or API agreement that restricts the right to report security flaws. The enforceability of contractual prohibitions on security research or vulnerability reporting is still in flux. Even if the researcher legally obtains a copy of software distributed with a EULA, but avoids the EULA click-through, either through reverse engineering or by obtaining the software pre-installed, we can not say for certain that a court will not enforce the EULA research and reporting restrictions. While it is more likely that a court will enforce a negotiated NDA than a mass market EULA, and less likely that the court will enforce a EULA that the researcher has not agreed to, the law is not clear. Be sure to consult with counsel if the code you have studied is subject to any kind of contractual restriction.

Note that you do not have whistleblower rights that protect your job if you choose to disclose. Your employer may want to maintain a relationship with the vendor of the product you have researched and tell you not to publish. In the absence of an employment agreement or specific whistleblower statute to the contrary (and there are none specifically on point), a private employer generally has the right to fire you for any non-discriminatory reason, including publishing a security vulnerability.

How Could Criminal Laws Like Conspiracy Or Aiding And Abetting Law Restrict Vulnerability Reporting? ^

Conspiracy requires proof of an agreement to commit a crime and an act that advances that objective. If you distribute vulnerability information pursuant to an agreement to illegally access computers, that is a crime.

Vulnerability publication could be aiding and abetting if the publisher distributes the information with the intent to further someone else's illegal activity. Intent is usually inferred from the circumstances surrounding the report. Because of First Amendment concerns, only rarely is criminal intent inferred from a publication to a general audience even if the publisher knows it will be used as part of an illegal act.[2] Publishing to peers, to the government or to a general audience is less likely to be aiding and abetting than is publishing to a single person with a grudge against the company. The more useful the information you publish is for criminal activity the more risk you face of aiding and abetting liability, even if you publish to people with whom you have no prior relationship.

In Sum, What Can I Do To Limit My Legal Risks When I Publish Vulnerability Information? ^

- Don't ask for money in exchange for keeping vulnerability information quiet. Researchers have been accused of extortion after saying they would reveal the vulnerability unless the company wants to pay a finder's fee or enter into a contract to fix the problem. See, *e.g.* GameSpy warns security researcher
- If you are under a non-disclosure agreement, you may not be allowed to publish. Courts are likely to hold researchers to their promises to maintain confidentiality.
- You may publish information to the general public, but do not publish directly to people you know intend to break the law.
- Consider disclosing to the vendor or system administrator first and waiting a reasonable and fair amount of time for a patch before publishing to a wider audience.
- Consider having a lawyer negotiate an agreement with the company under which you will provide details about the vulnerability—thus helping to make the product better—in exchange for the company's agreement not to sue you for the way you discovered the problem.
- Consider the risks and benefits of describing the flaw with proof-of-

concept code, and whether that code could describe the problem without unnecessarily empowering an attacker.

- Consider whether your proof of concept code is written or distributed in a manner that suggests it is "primarily" for the purpose of gaining unauthorized access or unlawful data interception, or marketed for that purpose. Courts look both to the attributes of the tool itself as well as the circumstances surrounding the distribution of that tool to determine whether it would violate such a ban.
- Consider whether to seek advance permission to publish, even if getting it is unlikely.
- Consider how to publish your advisory in a forum and manner that advances the state of knowledge in the field.
- Do not publish in a manner that enables or a forum that encourages copyright infringement, privacy invasions, computer trespass or other offenses.

[1] In *DVD CCA v. Bunner*, the defendant posted a copy of DeCSS acquired by reverse engineering, and the DVD CCA alleged violations of California's Uniform Trade Secrets Act. The trial court found the trade secrets were acquired by reverse engineering in violation of a license agreement and therefore acquired by improper means. *DVD Copy Control Ass'n, Inc. v. Bunner*, 31 Cal.4th 864 (Cal. 2003). While the California Supreme Court's subsequent opinion reached only whether a preliminary injunction violated the first amendment, a concurring opinion rejected the argument that a "no reverse engineering" EULA clause transformed reverse engineering into something other than a "fair and honest" way of acquiring trade secrets. *See id.* at 875, 901 n.5 (Cal. 2003) (Moreno, J., concurring) ("[N]owhere has it been recognized that a party wishing to protect proprietary information may employ a consumer form contract to, in effect, change the statutory definition of "improper means" under trade secret law to include reverse engineering, so that an alleged trade secret holder may bring an action even against a nonparty to that contract.")

[2] *United States v. Buttorff*, 572 F.3d 619 (8th. Cir. 1978) (information aiding tax protestors); *United States v. Barnett*, 667 F.3d 835 (9th Cir. 1982) (instructions for making PCP).

ELECTRONIC FRONTIER FOUNDATION
eff.org