Embargo On, Embargo Off

Art Manion
CERT Coordination Center
2019-03-14
Carnegie Mellon University
Software Engineering Institute

- Federally Funded Research and Development Center (FFRDC)

CERT Coordination Center (CERT/CC)

amanion@cert.org  @zmanion
CA-88:01

CERT Advisory
December 1988
ftpd vulnerability

** The sendmail portion of this advisory is superseded by CA-95:05. **

There have been several problems or attacks which have occurred in the past few weeks. In order to help secure your systems we have gathered the following suggestions:

1) Check that you are using version 5.59 of sendmail with the debug option DISABLED. To verify the version try the following commands. Use the telnet program to connect to your mail server. Telnet to your hostname or localhost with 25 following the host. The sendmail program will print a banner which will have the version number in it. You need to be running version 5.59. Version 5.61 will be released on Monday 12/12/1988. Any version less than 5.59 is a security problem.
Marvell Avastar wireless SoCs have multiple vulnerabilities

Vulnerability Note VU#730261

Original Release Date: 2019-02-05 | Last Revised: 2019-03-11

Overview

Some Marvell Avastar wireless system on chip (SoC) models have multiple vulnerabilities, including a block pool overflow during Wi-Fi network scan.
Vulnerabilities

- **Heartbleed**
  - OpenSSL, CVE-2014-0160
- **That Struts one that got Equifax**
  - CVE-2017-5638
  - 143M records
  - $439M direct costs (- $125M insurance)
- **Eternal Blue**
  - Windows SMB, CVE-2017-0144
  - Patient care impacted by WannaCry
- **Spectre**
  - And family
Vulnerabilities

- Varying definitions and counts
  - Bug, security bug, vulnerability
  - Hard to measure across different software ecosystems
- Changing technology
  - Connected things, some of which are safety-critical
  - Containers, microservices, SDx, autonomy
- Changing consequences
  - Life and limb
Vulnerabilities

• Varying definitions and counts
  • Bug, security bug, vulnerability
  • Hard to measure across different software ecosystems

• Changing technology
  • Connected things, some of which are safety-critical
  • Containers, microservices, SDx, autonomy

• Changing consequences
  • Life and limb
Coordinated Vulnerability Disclosure

Find vulnerability

Report vulnerability

Develop fix

Publish vulnerability and fix

Deploy fix

Embargo on

Embargo off
# CVD Considerations

<table>
<thead>
<tr>
<th>Question</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why?</td>
<td>Develop and test fixes? Deploy fixes at scale? Defend at scale?</td>
</tr>
<tr>
<td>When?</td>
<td>How long are others willing to hold embargos?</td>
</tr>
<tr>
<td>Under what sharing rules?</td>
<td>Ability to inform others Keeping embargos Willingness of others to follow rules</td>
</tr>
</tbody>
</table>
Risk Timelines

- Fixes available → Vulnerability public → Exploit available → Attacks
- Vulnerability public → Fixes available → Exploit available → Attacks
- Vulnerability public → Exploit available → Attacks → Fixes available
- Exploit available → Vulnerability public → Attacks → Fixes available
- Exploit available → Attacks → Vulnerability public → Fixes available
Disclosure Increases (Short-Term?) Risk

(a) Attacks exploiting zero-day vulnerabilities before and after the disclosure (time = t₀).

(b) Malware variants exploiting zero-day vulnerabilities before and after disclosure (time = t₀).

Figure 6: Impact of vulnerability disclosures on the volume of attacks. We utilize logarithmic scales to illustrate an increase of several orders of magnitude after disclosure.

Minimize Vulnerability and Disclosure Risk

Optimize embargo period for the least social cost (risk)

Figure 4: Social Cost as Function of $T$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>patch release time set by the vendor</td>
</tr>
<tr>
<td>$T$</td>
<td>protected period set by the social planner</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>proportion of customer loss internalized by the vendor</td>
</tr>
<tr>
<td>$\tau^s$</td>
<td>socially optimal patch release time (if the social planner could release patch)</td>
</tr>
<tr>
<td>$\tau^\infty$</td>
<td>patch release time by the vendor under secrecy policy</td>
</tr>
<tr>
<td>$T^k$</td>
<td>kink point in vendor’s reaction function w.r.t to $T$</td>
</tr>
</tbody>
</table>

When to Disclose

Rule for releasing fixes for embargoed bugs

From: Dominique Martinet <asmadeus () codewreck org>
Date: Fri, 17 Aug 2018 13:45:16 +0200

Hi,

I tried asking this question in private and was told there is no clear rule (and opinions vary) on the subject of releasing fixes for bugs still under embargo; and to ask the list, so here we go:

When should vendors publish fixes for bugs that are under embargo?

My opinion is that the point of security embargoes, and linux-distro in particular, is to give vendors time to prepare a fix so that fixes can be released almost immediately after the issue is made public.

• Linux kernel
  • “As a basic default policy, we expect report date to disclosure date to be on the order of 7 days.”
  • Recently changed to 14 calendar days after fix is ready
• distros and linux-distros mailing lists
  • “Please note that the maximum acceptable embargo period for issues disclosed to these lists is 14 days… In fact, embargo periods shorter than 7 days are preferable.”
    https://oss-security.openwall.org/wiki/mailing-lists/distros
  • Average ~6 days
    https://oss-security.openwall.org/wiki/mailing-lists/distros/stats
• Survey of public disclosure policies
  • 86 policies, ~2017
  • 35 specify embargo period
  • Range: 0-182 days
  • Median: 42 days
  • Mean: ~58 days
Why did OpenBSD silently release a patch before the embargo?

OpenBSD announced an errata on 30 August 2017 that silently prevented our key reinstallation attacks. More specifically, patches were released for both OpenBSD 6.0 and OpenBSD 6.1.

We notified OpenBSD of the vulnerability on 15 July 2017, before CERT/CC was involved in the coordination. Quite quickly, Theo de Raadt replied and critiqued the tentative disclosure deadline: “In the open source world, if a person writes a diff and has to sit on it for a month, that is very discouraging”. Note that I wrote and included a suggested diff for OpenBSD already, and that at the time the tentative disclosure deadline was around the end of August. As a compromise, I allowed them to silently patch the vulnerability. In hindsight this was a bad decision, since others might rediscover the vulnerability by inspecting their silent patch. To avoid this problem in the future, OpenBSD will now receive vulnerability notifications closer to the end of an embargo.

https://www.krackattacks.com/#openbsd
Takeaways

- Lots of bugs, security bugs, vulnerabilities
  - Relatively few contribute to disproportionate amount of risk
- Focus on response process: CVD
  - But also create fewer vulnerabilities and find them faster please
- Multi-party CVD gets complicated
- Embargo periods vary widely, there is not one correct answer
- Inclusion in the embargo is another important consideration
- Be aware of different timelines needed by different suppliers
- Set basic expectations, build common practices and norms