

لینوکس و پل USB به NVMe ریلک RTL9210

خلاصه:

- **علائم:** ریست‌های مکرر USB، خطاهای ورودی/خروجی، یا ناپدید شدن دیسک‌ها در لینوکس.
- **متأثرین:** ریلک RTL9210 (تأیید شده) و RTL9220 (احتمالاً).
- **علت:** بازگشت به ROM داخلی (f0.01) پس از شکست در بررسی مجموع کنترل.
- **تأثیر:** ناپایداری دائمی، هیچ ابزار بازنویسی برای لینوکس در دسترس نیست.
- **راه‌حل:** فقط ابزارهای ویندوزی OEM می‌توانند فریمور را بازیابی کنند - ریلک جایگزین‌های متن‌باز را مسدود می‌کند.

مقدمه

در سال 2025 میلادی، باید کاملاً منطقی باشد که یک Raspberry Pi را از یک SSD متصل شده از طریق USB بوت کنیم. با این حال، به لطف پیچیدگی‌های فریمور ریلک، این هدف منطقی به یک ماجرا تبدیل شده است. پس از ماه‌ها ناپایداری غیرقابل توضیح - ریست‌های تصادفی، ناپدید شدن دیسک‌ها، سیستم‌های فایل خراب - نویسنده تمام راه‌حل‌های معمولی را امتحان کرد: کابل‌های جدید، هاب‌های تغذیه‌شده، به‌روزرسانی‌های کرنل، تنظیمات USB، و بهینه‌سازی فریمور. پیشرفت تنها زمانی حاصل شد که ChatGPT به یک سؤال عجیب در نیمه‌شب پاسخ داد: «آیا ممکن است پل USB به NVMe به یک فریمور قدیمی بازگشته باشد؟»

معرفی

اگر محفظه NVMe مبتنی بر ریلک شما پس از هفته‌ها عملکرد بی‌نقص ناگهان ناپایدار شود - ریست‌های مکرر USB، خطاهای ورودی/خروجی، یا ناپدید شدن دیسک‌ها - شما تنها نیستید. این الگو در چندین برند، از واحدهای بدون نام گرفته تا OEM‌های شناخته‌شده مانند Sabrent و Orico ظاهر شده است. عامل مشترک: **چیپ‌های پل USB به NVMe ریلک RTL9210 و احتمالاً RTL9220.**

در ابتدا همه چیز کار می‌کند. سپس، به ظاهر بدون دلیل، دستگاه تحت بار یا در استفاده طولانی‌مدت، به‌ویژه در سیستم‌های لینوکس یا Raspberry Pi، شروع به قطع ارتباط می‌کند. علت واقعی نه SSD است و نه منبع تغذیه - این خود کنترل‌کننده فریمور است که به‌طور بی‌صدا به کد پشتیبان تعبیه‌شده در ROM بازمی‌گردد، نسخه‌ای که ریلک همچنان به صورت داخلی به عنوان f0.01 عرضه می‌کند.

مکانیزم مخفی - بازگشت فریمور به صورت طراحی شده

چیپ‌های پل ریلک فریمور عملیاتی و داده‌های پیکربندی خود را در یک فلش SPI خارجی ذخیره می‌کنند. هنگام روشن شدن، کنترل‌کننده یک مجموع کنترل ساده را بررسی می‌کند. اگر این مجموع کنترل مطابقت نداشته باشد، از بارگذاری

فریمور خارجی امتناع می‌کند و در عوض از ROM داخلی خود بوت می‌شود.

این فریمور پشتیبان قدیمی و معیوب است. فاقد چندین اصلاح پایداری USB و بهبودهای مدیریت وضعیت لینک است که در نسخه‌های بعدی وجود دارند، و این منجر به دنباله‌ای کلاسیک می‌شود که هر کاربر لینوکس آن را می‌شناسد:

usb 3-2: ریست دستگاه USB پرسرعت شماره 2 با استفاده از xhci-hcd

usb 3-2: خواندن توصیف‌گر دستگاه/64، خطا -71

هشدار EXT4-fs (دستگاه sda2): خطای ورودی/خروجی هنگام نوشتن در اینود ...

مجموع کنترل می‌تواند زمانی نامعتبر شود که داده‌های پیکربندی بازنویسی شوند - به‌عنوان مثال، هنگامی که پل تنظیمات مدیریت انرژی یا UAS خود را به‌روزرسانی می‌کند - و دستگاه در حین نوشتن برق خود را از دست بدهد. بوت بعدی یک مجموع کنترل خراب را می‌بیند و به‌طور دائم به فریمور ROM بازمی‌گردد.

در آن لحظه، محفظه NVMe «با کارایی بالا» شما دقیقاً مانند ارزان‌ترین پوسته بدون نام رفتار می‌کند، زیرا در داخل اکنون همان کد پایه معیوب حک‌شده در سیلیکون را اجرا می‌کند.

تأیید مشکل

شما می‌توانید این وضعیت را به‌راحتی در لینوکس تأیید کنید:

```
lsusb -v | grep -A2 Realtek
```

یک پل ریلtek سالم نسخه فریمور (**bcdDevice**) بالاتر از 1.00 را گزارش می‌دهد. یک پل بازگشتی نشان می‌دهد:

```
bcdDevice f0.01
```

این امضای **f0.01** به این معناست که کنترل‌کننده از ROM بوت می‌شود - و هیچ مقدار از قطع اتصال، فرمت مجدد، یا تنظیم کرنل آن را برطرف نخواهد کرد.

این مکانیزم بازگشت در **RTL9210** تأیید شده است. به نظر می‌رسد **RTL9220** همان معماری طراحی و چیدمان فریمور را به اشتراک می‌گذارد، بنابراین ممکن است رفتار مشابهی را نشان دهد، اما این همچنان محتمل است و نه ثابت‌شده.

چرا نمی‌توانید خودتان آن را تعمیر کنید

در اصل، راه‌حل ساده است: فریمور درست را به SPI بازنویسی کنید. در عمل، ریلtek این کار را غیرممکن می‌کند.

این شرکت یک به‌روزرسان با کد بسته برای ویندوز به OEMها و یکپارچه‌سازان ارائه می‌دهد. به کاربران لینوکس چیزی ارائه نمی‌شود. توسعه‌دهندگان جامعه ابزارهای فلش سازگار (**rtsupdater**, **rtl9210fw**, **rtsupdater-cli**) را مهندسی معکوس کردند که امکان بازیابی کامل فریمور از سیستم‌های لینوکس را فراهم می‌کرد - تا زمانی که ریلtek اخطارهای حذف DMCA را برای سرکوب آنها صادر کرد.

هیچ توجیه معقولی برای مالکیت معنوی برای مسدود کردن چنین ابزارهایی وجود ندارد: آنها میکروکد را افشا نمی‌کنند، فقط توالی به‌روزرسانی را از طریق USB هماهنگ می‌کنند. حذف‌های ریلک برای حفاظت نبودند. آنها ایدئولوژیک بودند.

هزینه یک ایدئولوژی

این درباره ایده‌آلیسم متن‌باز نیست. این درباره خصومت ایدئولوژیک یک فروشنده سخت‌افزار نسبت به سیستم‌های باز است که دستگاه‌هایی را که به‌عنوان سازگار با لینوکس به بازار عرضه می‌شوند، خراب می‌کند.

مقاومت ریلک در برابر مستندسازی و ابزارهای باز دو دهه است که ادامه دارد و شامل Wi-Fi، اترنت، صدا و اکنون کنترل‌کننده‌های ذخیره‌سازی می‌شود. این انزوا ممکن است در جهانی که فقط ویندوز است، مورد توجه قرار نگیرد، اما زمانی که همین چیپ‌ها در محصولات چندپلتفرمی مانند Sabrent EC-SNVE ادغام می‌شوند، که به‌طور آشکار لوگوی لینوکس را روی بسته‌بندی خود نمایش می‌دهد، سمی می‌شود.

ریلک با ممنوع کردن ابزارهای فلش لینوکس و مسدود کردن نگهداری جامعه، به‌طور مؤثر خود-تعمیر را جرم‌انگاری کرده است. عواقب به بیرون گسترش می‌یابد:

- کاربران لینوکس می‌بینند که سخت‌افزار «پشتیبانی‌شده» به ناپایداری فرو می‌رود.
- OEM‌هایی مانند Sabrent و Orico با هزینه‌های غیرضروری RMA و گارانتی مواجه می‌شوند.
- شهرت دیرینه ریلک برای سازگاری ضعیف با لینوکس بار دیگر تقویت می‌شود.

در نهایت، این متن‌باز نیست که دستگاه‌های ریلک را خراب می‌کند - این خصومت ریلک نسبت به متن‌باز است که آنها را خراب می‌کند.

مسیر منطقی به جلو

راه‌حل نیازی به تغییر ایدئولوژیک ندارد، فقط عمل‌گرایی. ریلک می‌تواند:

1. یک به‌روزرسان خط فرمان امضا شده توسط فروشنده برای لینوکس منتشر کند (بدون نیاز به افشای کد منبع).
2. الگوریتم مجموع کنترل را منتشر کند تا یکپارچه‌سازان بتوانند تصاویر فلش را با اطمینان تأیید کنند.
3. یک حالت به سبک DFU را بپذیرد که به‌روزرسانی‌ها را از طریق ذخیره‌سازی انبوه USB، مستقل از سیستم‌عامل، قبول کند.

هر یک از اینها از هزینه‌های گارانتی جلوگیری می‌کند، روابط OEM را محافظت می‌کند و اعتماد به چیپ‌های پل ریلک را در میان کاربران حرفه‌ای لینوکس - از سازندگان ایستگاه‌های کاری گرفته تا توسعه‌دهندگان Raspberry Pi - بازمی‌گرداند.

کاری که می‌توانید انجام دهید

اگر مشکوک هستید که محفظه شما به فریمور ROM بازگشته است:

- با `lsusb -v | grep bcdDevice` بررسی کنید.

- اگر **f0.01** را نشان دهد، مشکل را به OEM خود گزارش دهید.
- گزیده‌ای از **dmesg** را شامل کنید و به این مکانیزم بازگشت مستند اشاره کنید.
- از فروشنده خود بخواهید که مشکل را با ریلتنک مطرح کند، با اشاره به نیاز به یک به‌روزرسان سازگار با لینوکس.

سیاست فریمور ریلتنک تنها باعث ناراحتی علاقه‌مندان نمی‌شود؛ این سیاست خسارات مالی ملموسی برای اکوسیستم خود ایجاد می‌کند. هرچه زودتر این واقعیت در داخل شرکت به رسمیت شناخته شود، کاربران لینوکس و شرکای OEM می‌توانند زودتر از هدر دادن زمان در چرخه‌های RMA قابل اجتناب دست بکشند.

پاسخ‌های تولیدکنندگان

هر دو ریلتنک و Sabrent دعوت شدند تا در مورد مشکل بازگشت فریمور شرح داده شده در بالا اظهارنظر کنند. پاسخ‌های آنها - در صورت دریافت - در اینجا اضافه خواهد شد.

پیوست - شناسایی دستگاه‌های متأثر

کنترل‌کننده	شناسه فروشنده	شناسه محصول	یادداشت‌ها	وضعیت
RTL9210	0x0bda	0x9210	پل USB 3.1 Gen 2 10 Gb/s	تأیید شده رفتار بازگشت
RTL9220	0x0bda	0x9220	پل USB 3.2 Gen 2×2 20 Gb/s	محتمل، معماری مشابه

امضای بازگشت فریمور: **bcdDevice f0.01**

نسخه‌های پایدار شناخته‌شده: **1.23 – 1.31**