خاموش نشدنی؛

Implant.ARM.iLOBleed.a

نخستین روت‌کیت کشف شده در سفت افزار iLO سرورهای HP

دی ماه ۱۴۰۰

۱ فهرست مطالب

۲ چکیده فنی

سرورهای HP دارای یک ماژول مدیریتی به نام iLO مخفف Integrated Lights-Out هستند که به محض اتصال کابل برق روشن شده و یک سیستم عامل کامل اختصاصی را بارگذاری می‌کند. این ماژول با خاموش کردن سرور باز هم به کار خود ادامه می‌دهد و دارای دسترسی کاملی به کل سفت‌افزار، سخت‌افزار، نرم‌افزار و سیستم‌عامل سرور است و علاوه بر مدیریت سخت‌افزار سرور، به ادمین اجازه می‌دهد از راه دور سرور را روشن و خاموش نموده، به کنسول آن دسترسی داشته و حتی سیستم عامل روی آن نصب نماید.

دسترسی فوق‌العاده بالای این ماژول (بالاتر از هر سطح دسترسی در سیستم عامل)، احاطه کامل آن به سخت‌افزار، دور بودن از چشم ادمین و عمومی نبودن دانش و ابزارهای لازم برای بررسی و محافظت از آن، و ثابت بودن و عدم تغییر آن حتی با تغییر سیستم عامل، و به خصوص همواره روشن بودن دائمی خصوصیاتی هستند که این ماژول را به مثابه بهشتی برای نفوذگران و گروه‌های APT جهت پنهان نمودن بدافزار می‌گرداند.

ما در این گزارش به تحلیل یک نهان‌افزار[1] کشف شده در محیط عملیاتی[2] می‌پردازیم که خود را درون iLO پنهان می‌کند، با آپگرید سفت‌افزار قابل حذف نیست، و می‌تواند مدتها از چشم‌ها پنهان بماند. این بدافزار مدتی است در حال استفاده توسط نفوذگران می‌باشد و ما در حال بررسی عملکرد آنها بوده‌ایم. تا جایی که ما اطلاع داریم، این اولین گزارش از کشف بدافزاری واقعی در این سفت‌افزار در دنیا می‌باشد.

از آنجایی که پرداختن به تحلیل این بدافزار، نیازمند دانشی از معماری سفت‌افزار HP iLO می‌باشد، در این سند ابتدا اندکی درباره معماری HP و نقاط آسیب‌پذیر آن پرداخته شده است. سپس در بخش بعدی به تحلیل بدافزار کشف شده و ماژول‌های مختلف آن می‌پردازیم. در نهایت و در بخش آخر، به راهکارهای بررسی آلودگی و محافظت در برابر آن خواهیم پرداخت.

در کنار این گزارش، ابزارهایی جهت دامپ گیری و بررسی آلودگی سرورها توسعه داده شده است که در آینده نزدیک در اختیار عموم متخصصین قرار می‌گیرد. امیدواریم این گزارش نقطه شروعی برای توجه عمومی بیشتر به سفت‌افزارها و ایجاد راهکارهای محافظت از آنها باشد.

۲,۱ نکات کلیدی گزارش

- پنل مدیریتی iLO سرورهای HP محل امنی برای بدافزارها به شمار می‌رود که پس از آلودگی قابل تشخیص یا پاکسازی به روش‌های متعارف نیست.
- دسترسی به iLO و آلوده کردن آن، غیر از اینکه از طریق پورت شبکه iLO امکان‌پذیر است، از طریق داشتن دسترسی مدیر سیستم یا روت به سیستم عامل هاست نیز امکان‌پذیر است. یعنی در صورتیکه نفوذگر به کاربر admin ویندوز

[1] Rootkit
[2] In the wild

یا root سیستم عامل نصب شده روی سرور دسترسی داشته باشد، می‌تواند با iLO ارتباط برقرار کرده و در صورت آسیب‌پذیر بودن آن را آلوده نماید.

- تحقیقات انجام شده در طول سالیان گذشته، آسیب‌پذیری‌های متعددی را در HP iLO نشان داده‌اند که موجب ارائه وصله و تغییرات معماری از طرف سازنده شده است.
- در نسخه‌های iLO4 و قبلتر که در سرورهای G9 و ماقبل استفاده می‌شوند، مکانیزم بوت امن با داشتن کلید ریشه معتمد در سخت‌افزار وجود نداشته و سفت‌افزار این نسخه‌ها توسط بدافزار قابل تغییر و آلوده شدن می‌باشد.
- حتی در صورت به‌روزرسانی iLO به آخرین نسخه مربوطه که آسیب‌پذیری شناخته شده‌ای نداشته باشد، باز هم امکان دانگرید[1] آن به نسخه‌های پایین‌تر آسیب‌پذیر وجود دارد. این موضوع در کلیه سرورهای HP وجود داشته و فقط در سری G10 به شرط فعال نمودن یک تنظیم غیرپیش‌فرض قابل جلوگیری می‌باشد. در سایر سرورها، امکان جلوگیری از دانگرید و آلوده شدن سرور وجود ندارد.
- با توجه به موارد فوق، حتی قطع کردن کابل شبکه iLO یا آپگرید نمودن آن به آخرین نسخه جهت جلوگیری از آلودگی بدافزاری کافی نمی‌باشد.

- از سال ۲۰۲۰ گروه تحلیل بدافزار شرکت امن‌پرداز نهان‌افزاری را کشف کرده‌اند که یک ماژول بدافزاری با نام Implant.ARM.iLOBleed.a را به سفت‌افزار iLO اضافه می‌کند و تعدادی از ماژول‌های اصلی سفت‌افزار را نیز تغییر می‌دهد. به این ترتیب امکان به‌روزرسانی سفت‌افزار از بین می‌رود و دسترسی‌هایی به سخت‌افزار سرور ایجاد می‌شود که یکی از نتایج آن حذف کامل اطلاعات موجود روی دیسک‌های سرور است.

- ابزار بررسی صحت HP iLO توسط امن‌پرداز به زودی به صورت عمومی منتشر شده و در دسترس عموم قرار می‌گیرد.

۲,۲ علائم آلودگی

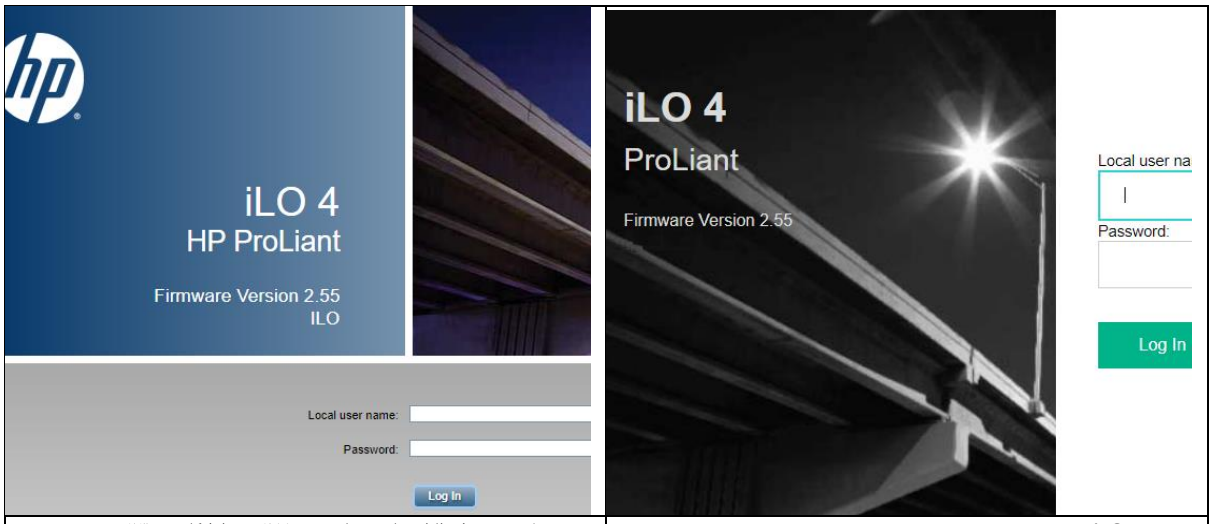
اگر چه معمولا در علائم آلودگی مواردی مانند هش و ... بیان می‌شود، در دسترس نبودن ابزار dump سفت‌افزار iLO انتشار این نوع علائم آلودگی را بی‌فایده می‌کند. به علاوه در صورت دسترسی به این ابزار، استفاده از راهبرد لیست سفید یعنی مقایسه دامپ تهیه شده با مجموعه‌ی محدود سفت‌افزارهای اصیل HP کاری ساده‌تر و موثرتر می‌باشد.

اما اگر می‌خواهید بدانید که سرور شما آلوده شده است یا خیر، یک راه تشخیص ساده برای آلودگی وجود دارد:

همانگونه که بیان شد، این بدافزار جهت حفظ استتار و ماندگاری خود بعد از آپگرید سفت‌افزار، این عملیات را شبیه‌سازی می‌کند. اگر چه بدافزار تلاش می‌کند که با برداشتن شماره نسخه سفت‌افزار آپگرید شده و نمایش آن در محل‌های مختلف منجمله صفحه اصلی لاگین iLO، روند آپگرید را موفق جلوه بدهد، اما یک نکته وجود دارد: شرکت HP در رابط کاربری iLO تغییرات چشمگیری داده است. بنابراین تشخیص یک سفت‌افزار نادرست به سادگی با چشم ممکن است.

[1] Downgrade

در شکل ۱ می‌توانید مقایسه دو صفحه لاگین واقعی و جعلی (آلوده شده توسط بدافزار) را مشاهده کنید. هر دو صفحه اعلام می‌کنند که iLO نسخه ۲.۵۵ هستند اما صفحه لاگین آلوده، در واقع صفحه لاگین مربوط به iLO قدیمی ۲.۳۰ می‌باشد و فقط شماره نسخه آن توسط بدافزار جعل شده است.



صفحه جعل شده توسط بدافزار (نمایش شماره نسخه ۲.۵۵ روی رابط کاربری ۲.۳۰) | صفحه اصلی iLO نسخه ۲.۵۵

شکل ۱ – مقایسه صفحه لاگین جعلی (آلوده به بدافزار) با لاگین واقعی iLO

البته مانند هر علامت آلودگی و IOC دیگری، می‌شود انتظار داشت که بدافزارنویسان راهی برای تغییر این علامت و پنهان کردن آن بیابند. اما تا آن زمان، این روش ساده و موثری برای تشخیص بدافزار بدون نیاز به هیچ گونه ابزاری می‌باشد.

۳ فناوری HP iLO

شرکت HP فناوری iLO[1] را برای مدیریت این سیستم‌ها در اختیار کاربرانی که به عنوان مدیر سیستم فعالیت می‌کنند ارائه کرده است. این فناوری به مدیران سیستم اجازه می‌دهد تا به صورت ریموت و از راه دور با استفاده از یک واسط شبکه، به مشخصه‌های گوناگونی از سیستم تحت مدیریت خود دسترسی داشته باشند از جمله:

- مدیریت توان مصرفی
- دسترسی به کنسول سیستم از راه دور
- نصب CD/DVD Image از راه دور
- رصد و کنترل بسیاری از شاخص‌های سخت‌افزاری و نرم‌افزاری سیستم از راه دور

شرکت HP برای نسل‌های گوناگون سروهای خود در سال‌های اخیر نسخه‌های متنوعی از سفت‌افزار iLO را ارائه کرده است که جدول ۱ این نسخه‌ها را نمایش می‌دهد.

جدول ۱- نسخه‌های سفت‌افزار iLO برای نسل‌های گوناگون سرورهای HP

سری سفت‌افزار iLO	نسل سرور HP ProLiant	آخرین نسخه سفت‌افزار ارائه شده
iLO	نسل ۲ و ۳ و ۴ و نسل ۶ (زیر سری ۳۰۰)	۱.۹۶- آوریل ۲۰۱۴
iLO 2	نسل ۵ و۶ (بالای سری ۳۰۰)	۲.۳۳- مارس ۲۰۱۸
iLO 3	نسل ۷	۱.۹۳- آگوست ۲۰۲۰
iLO 4	نسل ۸ و ۹	۲.۷۵- آگوست ۲۰۲۰
iLO 5	نسل ۱۰	۲.۳۰- سپتامبر ۲۰۲۰

به دلیل حوزه اختیارات و عملکرد حساس این واسط سفت‌افزاری در تمام خانواده‌های سرور، سناریوهای گوناگونی برای حمله به این واسط قابل تصور است. از جمله این حملات، دست‌یابی به رمز عبور واسط مدیریتی، کشف و به‌کارگیری آسیب‌پذیری‌های امنیتی و همچنین فلش کردن سفت‌افزار آلوده به جای سفت‌افزار اصلی روی سرور است.

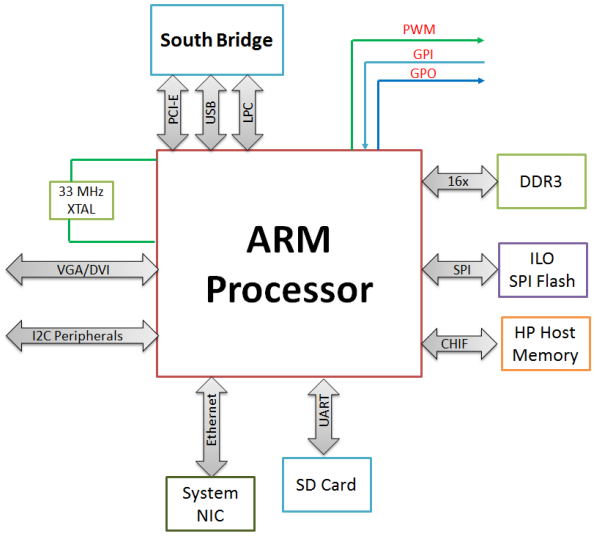
در سال‌های اخیر مجموعه‌ای از مطالعات با هدف شناخت آسیب‌پذیری‌های امنیتی واسط مدیریتی iLO سرورهای HP انجام گرفته است [۱][۲][۳]. این مطالعات در نهایت منجر به کشف تعدادی آسیب‌پذیری با درجه خطر بالا و متوسط شده است. متأسفانه انتشار عمومی این آسیب‌پذیری‌ها و نمونه‌های کد برای اثبات این آسیب‌پذیری‌ها سبب شده است تا در چند سال اخیر، افراد یا گروه‌های مهاجم با استفاده از این آسیب‌پذیری‌ها، به زیرساخت شبکه برخی از سازمان‌ها که از سرورهای HP به منظور مدیریت شبکه‌های سازمانی خود استفاده می‌کنند حملاتی انجام دهند.

[1] Integrated Lights-Out

۳،۱ معماری سخت‌افزاری iLO

از دیدگاه سخت‌افزاری، iLO به‌صورت مجتمع در کنار بورد اصلی سیستم تعبیه شده است و شامل موارد زیر است [۲]:

- پردازنده ARM با معماری GLP/Sabine
- حافظه فلش برای ذخیره‌سازی سفت‌افزار
- حافظه RAM اختصاصی
- واسط شبکه اختصاصی
- مجموعه‌ای از درگاه‌های سخت‌افزاری برای ارتباط با بخش‌های کنترلی



شکل ۲- شماتیک سخت‌افزاری iLO

شکل ۲ شماتیک سخت‌افزاری iLO را نمایش می‌دهد. همانطور که در این شکل قابل مشاهده است، پردازنده ARM به واسط PCI-Express به پل جنوبی و از طریق آن به پردازنده اصلی سرور متصل می‌شود. همچنین iLO به صورت مستقیم می‌تواند با CMOS نیز ارتباط برقرار کند. این نوع از ارتباط برای تنظیم متغیرهایی چون Boot Order است که از طریق واسط مدیریتی iLO این امر را امکان‌پذیر می‌کند.

۳،۱،۱ پردازنده iLO

پردازنده مورد استفاده در iLO از سری پردازنده‌های نسل هفتم و هشتم ARM هستند. این پردازنده‌ها در کنار قدرت پردازش خوبی که دارند از سری تراشه‌های بسیار کم مصرف محسوب می‌شوند. این امر کمک می‌کند تا در حالتی که سرور در حالت انتظار قرار دارد بدون مصرف زیاد انرژی و جریان بیش از اندازه بتواند واسط مدیریتی را در اختیار مدیر شبکه قرار دهد.

۳,۱,۲ حافظه‌های جانبی

پردازنده iLO می‌تواند با حافظه‌های متفاوتی در ارتباط باشد. به طور خاص، دو مورد از این تراشه‌ها در این بخش مورد بررسی قرار گرفته‌اند. نخست تراشه اصلی این سیستم است که بر روی آن سفت‌افزار iLO قرار دارد و هنگام بالا آمدن iLO، این سفت‌افزار بارگذاری می‌شود.

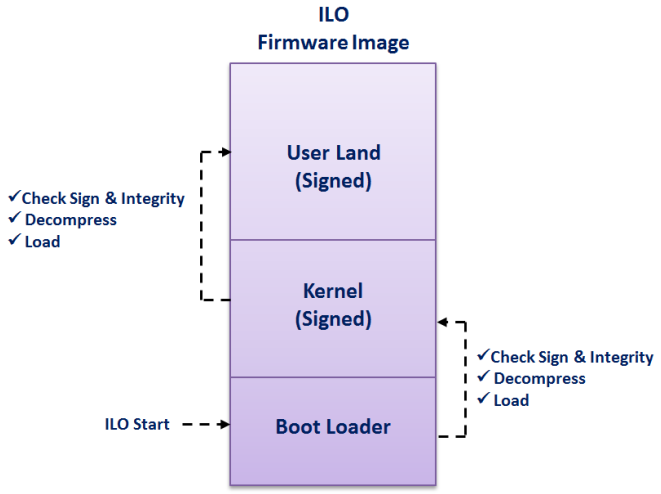
مورد دوم حافظه تراشه‌ای است که از آن با عنوان iLO NAND Flash یاد می‌شود که به عنوان حافظه جانبی سیستم در اختیار سیستم‌عامل iLO قرار دارد. پس از بارگذاری، سیستم‌عامل iLO با کمک این حافظه می‌تواند به ذخیره فایل‌هایی از قبیل تاریخچه رخدادهای سیستم بپردازد. همچنین این حافظه محل ذخیره برنامه‌هایی است که توسط سیستم عامل iLO اجرا می‌شوند.

۳,۱,۳ ارتباط iLO با سرور

فناوری iLO به‌عنوان یک واحد مدیریت و کنترل سرور، به تمامی اجزاء سخت‌افزاری سرور از قبیل حافظه، پردازنده، درگاه‌های ورودی و خروجی و دیسک سخت دسترسی مستقیم دارد. همچنین پردازنده اصلی سرور نیز iLO را به‌عنوان یک ماژول PCI شناسایی می‌کند و می‌توان با آن ارتباط برقرار کند.

۳,۲ ساختار سفت‌افزار iLO

سفت‌افزار iLO به صورت یک فایل باینری درون حافظه فلش SPI (اغلب با اندازه ۱۶ مگابایت) ذخیره شده است. همان‌طور که در شکل ۳ دیده می‌شود، این سفت‌افزار از ۳ بخش اصلی Boot loader، هسته سیستم عامل (Kernel) و بخش ماژول‌ها تشکیل شده است. از این ۳ بخش، تنها بخش Boot loader به صورت رمزگذاری نشده است و دو بخش دیگر رمزگذاری شده و فشرده (با الگوریتم LZMA) و دارای امضاء امنیتی هستند. تمامی محتوای سفت‌افزار iLO به صورت کدهای کامپایل‌شده C با معماری ARM است.



شکل ۳- ساختار درونی سفت‌افزار iLO

از دیدگاه نرم‌افزاری، iLO سرویس‌های گوناگونی مانند Web Server و SSH Server را برای مدیران سرورها ارائه می‌کند. در واقع iLO به منزله یک سیستم‌عامل کامل است که به محض اتصال سیستم به برق، حتی در شرایطی که هاست سرور خاموش است بالا می‌آید و سرویس‌های خود را در دسترس قرار می‌دهد.

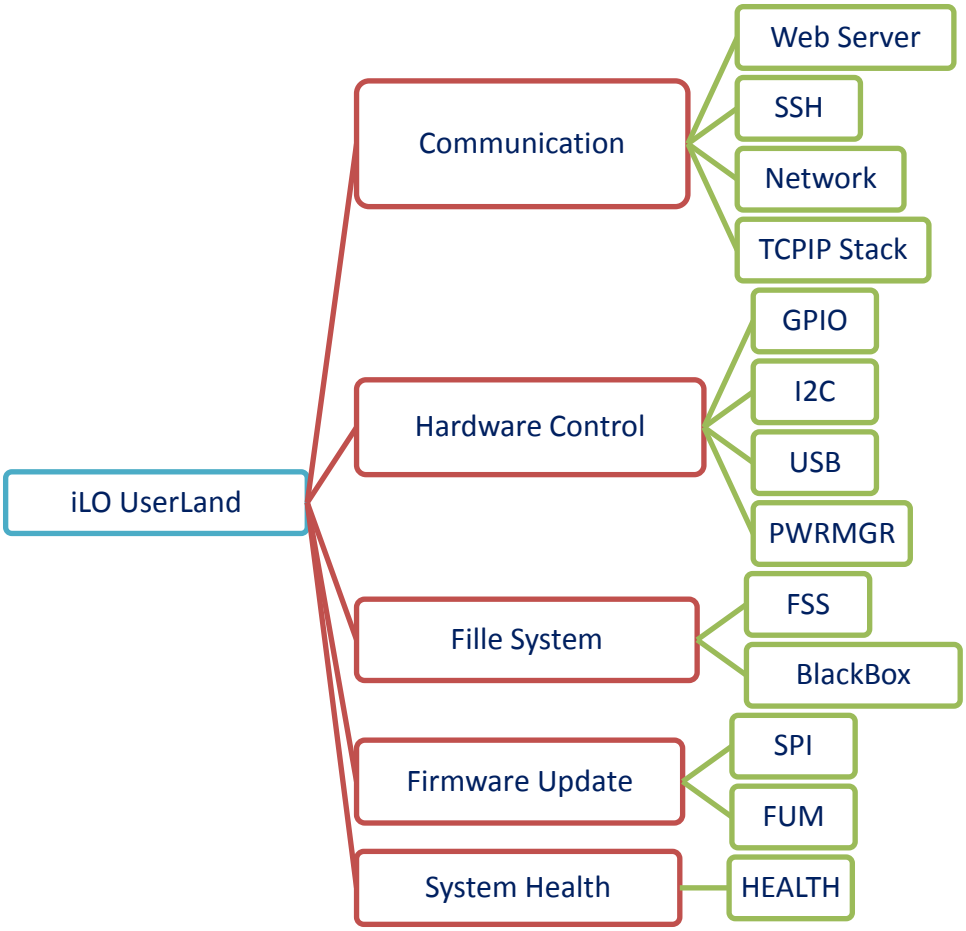
در زمان بوت شدن iLO، قبل از اجرای هر بخش (هسته سیستم‌عامل و ماژول‌های اجرایی)، بخش قبل باید امضای آن را صحت‌سنجی نماید تا اجازه اجرا به آن داده شود. به این ترتیب، بخش Boot Loader مسئول صحت‌سنجی امضاء، استخراج از وضعیت فشرده و بارگذاری بخش Kernel خواهد بود. همچنین بخش Kernel مسئول صحت‌سنجی امضاء، استخراج از وضعیت فشرده و بارگذاری ماژول‌های اجرایی خواهد بود.

سیستم‌عامل مورد استفاده در بخش سفت‌افزار iLO یک سیستم‌عامل بی‌درنگ به نام Integrity است که توسط شرکت Green Hills Software[1] توسعه داده شده است و مسئول اجرای وظایف موجود در ناحیه کاربر[2] است. ناحیه کاربر در واقع یک فایل باینری ELF با معماری ARM است که توسط شرکت HP توسعه یافته و پک شده است. این فایل دارای ماژول‌های متنوعی است هر یک دارای وظیفه خاصی هستند. هر وظیفه به‌صورت یک پردازه با فضای حافظه مجازی اختصاصی و مجموعه‌ای از ریسمان‌های[3] پردازشی است که در ناحیه کاربر اجرا می‌شود. در بخش‌های بعد این مستند عملکرد، برخی از مهمترین ماژول‌های iLO تشریح خواهد شد.

۳,۳ ماژول‌های iLO

شکل ۴ تعدادی از ماژول‌های User Land در سفت‌افزار iLO 4 را نشان می‌دهد. در ادامه این بخش، عملکرد تعدادی از مهم‌ترین ماژول‌ها تشریح می‌شوند.

[1] http://www.ghs.com/products/rtos/integrity.html
[2] User Land
[3] Threads



شکل ۴- تعدادی از ماژول‌های User Land در سفت‌افزار iLO 4

۳،۳،۱ ماژول Web Server: ارائه واسط مدیریتی وب

ماژول Web Server وظیفه ارائه واسط مدیریتی iLO در قالب سرویس وب را بر عهده دارد. این ماژول دارای بخش‌هایی نظیر واسط وب، یک واسط برنامه‌نویسی[1] XML، یک واسط برنامه‌نویسی Redfish، و یک کنسول کنترل از راه دور است.

اتصال به ماژول Web Server می‌تواند از نوع HTTP و یا HTTPS باشد. این ماژول دارای چهار ریسمان پردازشی است که هر ریسمان مسئول اداره و پاسخگویی به یکی از اتصالاتی است که با ماژول ایجاد می‌شوند. هر درخواست اتصال به‌صورت خط به خط مورد پردازش قرار می‌گیرد و محتوای آن تجزیه و تحلیل می‌شود و در صورت صحت احراز هویت و سطح دسترسی آن، پاسخ مورد نظر ارائه می‌شود. اگرچه تقریبا دسترسی به تمامی صفحات وب نیاز به احراز هویت کاربر دارد، اما **برخی از داده‌ها در قالب XML بدون طی کردن فرآیند احراز هویت قابل دسترسی هستند.**

[1] API

۳،۳،۲ ماژول CHIF: ارتباط با پردازنده سرور

ماژول CHIF[1] یکی از ماژول‌های موجود در iLO است که وظیفه ارتباط با بخش‌های پردازنده و حافظه سرور و انتقال پیام میان iLO و سیستم عامل میزبان را بر عهده دارد. به طور خلاصه می‌توان وظایف این ماژول را به شرح زیر لیست کرد [۳]:

- انتظار برای دریافت پیام از سمت سرور
- ارسال پیام دریافتی به واحد پردازش پیام (Command Handler) متناسب با پیام
- ارسال پیام‌های خاص به سایر ماژول‌های مرتبط با پیام

عملیات انتقال و پردازش پیام‌ها در این ماژول به‌طور پیش‌فرض هیچ‌گونه فرآیند احراز هویتی را طی نمی‌کند.

۳،۳،۳ ماژول FUM: به‌روزرسانی سفت‌افزار iLO

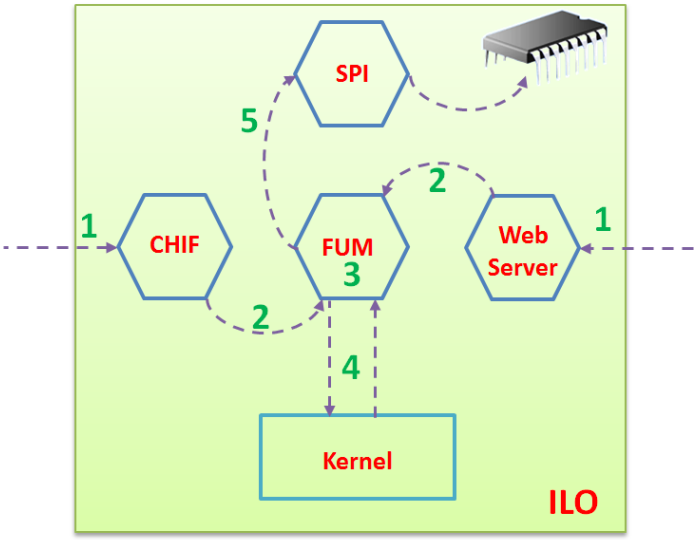
وظیفه اصلی ماژول FUM[2] به‌روزرسانی سفت‌افزار iLO است. این وظیفه از سه طریق امکان‌پذیر است:

- از طریق واسط مدیریتی HP Intelligent Provisioning
- با استفاده از سیستم‌عامل میزبان نصب شده روی سرور و با دسترسی به واسط PCI-E
- از راه دور و از طریق واسط مدیریتی واسط وب iLO

ماژول FUM عملیات به‌روزرسانی سفت‌افزار را در ۵ مرحله انجام می‌دهد که شامل مراحل زیر است [۳]:

۱- فایل سفت‌افزار جدید از طریق هاست سرور و یا ماژول Web Server دریافت می‌شود.
۲- فایل سفت‌افزار جدید به ماژول FUM ارسال می‌شود.
۳- ماژول FUM فایل سفت‌افزار جدید را از جهت صحت امضاء و جامعیت بررسی و اعتبار سنجی می‌کند.
۴- ماژول FUM از هسته سیستم‌عامل نیز می‌خواهد تا صحت فایل سفت‌افزار جدید را اعتبار سنجی کند.
۵- ماژول FUM از ماژول SPI درخواست می‌کند تا فایل سفت‌افزار جدید را روی حافظه فلش SPI برنامه‌ریزی نماید.

[1] Channel Interface
[2] Firmware Update Module



شکل ۵- فرآیند به‌روز رسانی سفت‌افزار iLO توسط ماژول FUM

در این فرآیند، **صحت‌سنجی جامعیت سفت‌افزار از طریق بررسی صحت امضاء دیجتال فایل سفت‌افزار انجام می‌گیرد** تا ماژولی به سفت‌افزار اصیل که توسط شرکت HP ارائه شده است افزوده یا کاسته نشود. نکته مهم این است که **در فرآیند به‌روز رسانی سفت‌افزار، امکان به‌روزرسانی به نسخه پایین‌تر سفت‌افزار وجود دارد.**

۳,۳,۴ ماژول SPI: دسترسی به حافظه فلش

وظیفه اصلی این ماژول، ارتباط با حافظه فلش SPI حاوی سفت‌افزار iLO و سفت‌افزار BIOS سرور است. از طریق این ماژول، عملیات خواندن، نوشتن و پاک کردن حافظه‌های فلش حاوی سفت‌افزار انجام می‌شود. همان‌طور که قبلاً نیز اشاره شد، بخش‌نهایی عملیات به‌روز رسانی سفت‌افزار iLO از طریق این ماژول انجام می‌شود.

۳,۳,۵ ماژول ConAppCli: سرویس کنسول

این ماژول وظیفه ارائه سرویس واسط کاربری خط فرمان دریافت فرمان‌های کاربر و مدیر سرور را بر عهده دارد. این فرمان‌ها می‌توانند از نوع روشن/خاموش کردن iLO، مدیریت کاربران، مدیریت توان مصرفی، مشاهده رخدادهای سیستم و .. باشد.

۳,۳,۶ ماژول SSH: خط فرمان از راه دور

سفت‌افزار iLO علاوه بر واسط وب، با استفاده از ماژول SSH سرویس shell رمز شده را در اختیار کاربر قرار می‌دهد و کاربر می‌تواند از طریق پورت ۲۲ با سفت‌افزار ارتباط برقرار نماید و مجموعه‌ای از فرمان‌ها را اجرا نماید.

۳,۳,۷ ماژول Health: سرویس رصد صحت اجزاء سیستم و ثبت رخدادها

وظیفه این ماژول، بررسی دوره‌ای وضعیت مشخصات سیستم و ثبت رخدادهای سرور است. مشخصات سیستم شامل دمای کاری، سرعت فن‌ها، وضعیت منبع تغذیه، وضعت حافظه‌های سیستم، شبکه، پردازنده‌ها و ... است.

۳,۳,۸ ماژول BlackBox: جعبه سیاه سیستم

این ماژول به منزله جعبه‌سیاه سرور عمل می‌کند و بسیاری از اطلاعات و رخدادهای حساس و حیاتی سیستم که توسط ماژول Health ثبت می‌شوند، توسط این ماژول به‌صورت روزانه در قالب یک فایل فشرده ذخیره می‌شوند.

۳,۳,۹ سایر ماژول‌ها

علاوه بر ماژول‌های ذکر شده در بخش‌های قبل، سفت‌افزار iLO دارای ماژول‌های متنوع دیگری است که هریک در بخش UserLand دارای وظیفه مشخصی هستند. ماژول‌هایی مانند SNMP و SNTP و SVCSiLO وظایف مدیریت سیستم و شبکه را بر عهده دارند و ماژول‌هایی مانند USB، GPIO، I2C دسترسی کنترلی iLO به بخش‌های سخت‌افزاری سرور را فراهم می‌کنند.

۴ تحلیل نهان‌افزار Implant.ARM.iLOBleed.a

در این بخش به تحلیل فنی نهان‌افزار کشف شده در سفت‌افزار HP iLO می‌پردازیم.

زمانیکه تیم تحلیل امن‌پرداز این بدافزار را کشف نمود، مهاجمان تصمیم به تخریب اطلاعات دیسک‌های سرور و از بین بردن کامل آن گرفته بودند. جالب اینجاست که مهاجمین به یکبار تخریب اکتفا نکرده، و بدافزار را برای اجرای مکرر تخریب در بازه‌های زمانی تنظیم کرده بودند. به این ترتیب بعد از هر بار راه‌اندازی سرور با سیستم عامل جدید، پس از مدتی مجددا کل هاردها تخریب می‌گردید. اما این بدافزار برخلاف سایر بدافزارهای Wiper که صرفا به تخریب اطلاعات می‌پردازند، به قصد یکبار اجرا و ضربه زدن ایجاد نشده است.

یکی از قابلیت‌های مهم این بدافزار، دستکاری روال آپگرید سفت‌افزار iLO می‌باشد، به این نحو که در صورتی که مدیر شبکه تلاش کند سفت‌افزار iLO را به به نسخه جدیدی ارتقا دهد، بدافزار ضمن جلوگیری از انجام روال آپگرید، تغییر نسخه را شبیه‌سازی می‌کند. به این منظور علاوه بر آنکه روال آپگرید در ظاهر اجرا شده و پیام انجام موفقیت‌آمیز آن نمایش داده می‌شود، حتی شماره نسخه سفت‌افزار نیز در ظاهر کنسول وب و سایر مکان‌ها افزایش پیدا می‌کند، در حالیکه در واقع هیچگونه آپگریدی انجام نشده است.

همین مساله به تنهایی نشان می‌دهد که هدف از تولید آن، ایجاد یک بدافزار با حداکثر مانایی و مخفی ماندن از کلیه مکانیزم‌های شناخته شده امنیتی می‌باشد. بدافزاری که با پنهان شدن در یکی از پرقدرت‌ترین منابع پردازشی که همواره نیز روشن است، قابلیت انجام هر دستوری از طرف مهاجمین را داشته باشد، بدون اینکه هرگز کشف گردد.

طبیعتا هزینه صرف شده برای چنین حمله‌ای آن را در دسته APTها قرار می‌دهد. اما استفاده از چنین بدافزار قدرتمند و پرهزینه‌ای برای کاری مانند تخریب اطلاعات که احتمال کشف بدافزار را افزایش می‌دهد، به نظر یک اشتباه فاحش از سمت این گروه می‌باشد.

سایر نکات فنی درباره این بدافزار را در بخش‌های آتی مطالعه می‌فرمایید.

۴،۱ ابزار تهیه رونوشت از سفت‌افزار

اولین گام برای بررسی آلودگی سفت‌افزار، تهیه رونوشت یا اصطلاحا دامپ[1] کامل از سفت‌افزار جهت بررسی می‌باشد.

متاسفانه شرکت HP ابزار یا روشی را جهت بررسی و خواندن سفت‌افزار iLO ارائه نکرده است. به همین منظور تهیه یک ابزار جهت دامپ سفت‌افزار در دستور کار قرار گرفت که نهایتا به صورت ابزار Padvish iLO Scanner در دو نسخه تکامل پیدا کرد:

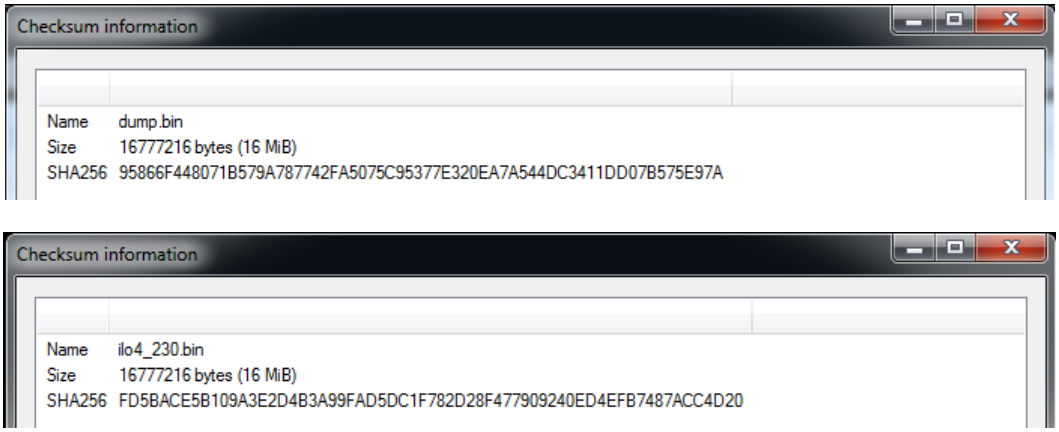
[1] Dump

۱. **پویش از داخل سیستم عامل میزبان:** همانطور که بیان شد، سخت‌افزار iLO به عنوان یک کارت PCI-Express از طریق پردازنده اصلی و سیستم عامل نصب شده روی سیستم قابل دسترسی می‌باشد. البته شرکت HP نیز ابزاری با نام flash_ilo جهت نوشتن و به‌روزرسانی سفت‌افزار به نسخه‌های جدید برای سیستم‌عامل‌های مختلف ارائه داده است، اما ابزار مذکور تنها امکان نوشتن سفت‌افزار را فراهم کرده و اجازه خواندن سفت‌افزار قبلی را نمی‌دهد. به همین منظور و بر پایه دانش به دست آمده در حوزه iLO، ابزاری جهت خواندن سفت‌افزار و تهیه دامپ از آن توسعه دادیم.

۲. **پویش از طریق پورت iLO:** از آنجاکه پویش از طریق سیستم عامل میزبان ممکن است همواره میسر نبوده و انجام آن روی سرورهای عملیاتی و یا در حجم انبوه برای مدیران شبکه دشوار باشد، روش دیگری جهت پویش سفت‌افزار مدنظر قرار گرفت. این نسخه از پویشگر از طریق استفاده از یکی از آسیب‌پذیری‌های شناخته شده روی iLO و با اجرای کد از راه دور، امکان تهیه دامپ را فراهم می‌کند. به علت استفاده از آسیب‌پذیری، دریافت دامپ در این نسخه فقط از سفت‌افزار سرورهای HP iLO4 با بازه نسخه سفت‌افزار ۲.۳۰ تا ۲.۵۰ وجود دارد.

۴,۲ تحلیل سفت‌افزار آلوده

پس از تهیه رونوشت از سفت‌افزار یک سرور، باید آن را با نسخه‌های سفت‌افزار اصلی مقایسه نمود. بدافزار Implant.ARM.iLOBleed.a بر اساس نسخه ۲.۳۰ سفت‌افزار iLO توسعه داده شده است. بر همین اساس تفاوت بین این نسخه آلوده را با نسخه اصلی در (شکل ۶) مشاهده می‌کنید.



شکل ۶- اختلاف امضاء سفت‌افزار: (بالا) رونوشت سیستم؛ (پایین) نسخه اصلی ارائه شده توسط شرکت HP

در یک بررسی دقیقتر، اجزاء هر دو سفت‌افزار در سطح فایل‌سیستم و ماژول‌ها نیز مقایسه شدند که نتایج این مقایسه در جدول ۲ قابل مشاهده است.

جدول ۲- مقایسه امضاهای ماژول‌های سفت‌افزار سیستم مورد حمله با نسخه اصلی

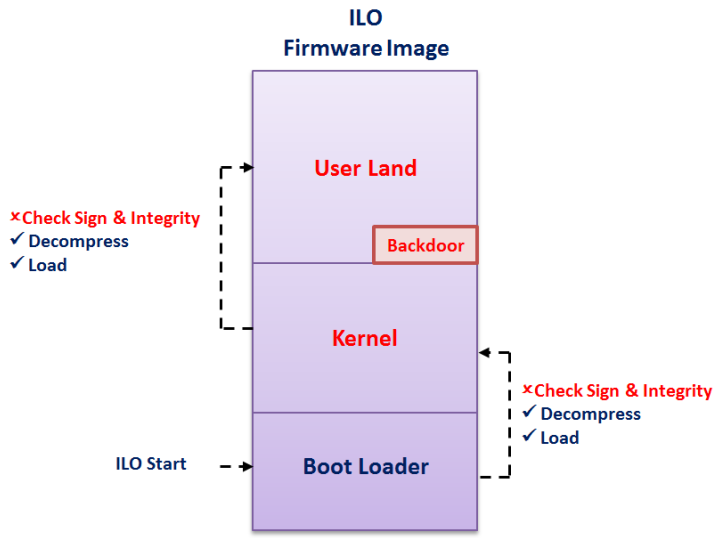
Module Name		MD5 (Original)	MD5 (Infected)	Difference (Bytes)
hpimage.bin		98af47cb8cacb25abd333d8a1a752c6b	4f8417af3a6f75780e09c5792397a05f	15.625MB
-	hpimage.hdr	8433650ef98fd8790877e6616c02b66c	8433650ef98fd8790877e6616c02b66c	0
	bootloader.hdr	ae22d82a3e954ecf911b834463dbfbbe	ae22d82a3e954ecf911b834463dbfbbe	0
	bootloader.bin	20ff78c6604563c27b6f9c75775c9306	1fdb4270665177ecb1c9708039bab934	5 B
	kernel_main.hdr	e1b1244fead44f73efb7b559e9d719c9	7df3b258ca3c12f0f8de77469456e25d	2 B
	kernel_main.bin	bacc259ea63785607faf2dab6939a2db	9ab97c5b03664da18ab1f775dc11c200	12 B
	kernel_recovery.hdr	e1b1244fead44f73efb7b559e9d719c9	7df3b258ca3c12f0f8de77469456e25d	2 B
	kernel_recovery.bin	bacc259ea63785607faf2dab6939a2db	9ab97c5b03664da18ab1f775dc11c200	12 B
	ELF.hdr	7db6ebd698fa4862cfde68a546e9a75b	64d0143d638885745b241796268eb0b2	2 B
	ELF.bin	d16fee481f78ad0275dd29ed271582aa	bdeeab3994ec5d0b93d961148a6b712d	15.625MB

طبق اطلاعات این جدول، از کل ماژول‌های تشکیل دهنده hpimage.bin، تنها دو بخش سرآیند hpimage و سرآیند bootloader یکسان هستند. در سایر بخش‌ها تفاوت امضاء نشان‌دهنده وجود اختلاف میان دو فایل در این بخش‌ها است. همچنین می‌توان مشاهده کرد که بخش عمده تغییرات مربوط به ماژول ELF.bin می‌باشد، در حالیکه سایر ماژول‌ها فقط ۲ تا ۱۲ بایت تغییر داشته‌اند.

۴،۲،۱ مانایی بدافزار در بوت

یکی از دغدغه‌های توسعه‌دهندگان هر نوع بدافزار، آلوده ماندن سیستم پس از قرار گرفتن بدافزار در سیستم است.

همانطور که در بخش ساختار سفت‌افزار iLO بیان شد، در فرآیند راه‌اندازی iLO ماژول bootloader وظیفه اعتبارسنجی امضای هسته سیستم عامل را برعهده داشته، و هسته سیستم نیز وظیفه اعتبارسنجی امضای ماژول‌ها را بر عهده دارد. بنابراین، در صورتی که فرد مهاجم بخواهد در سفت‌افزار iLO درب پشتی ایجاد کند، علاوه بر قرار دادن درب‌پشتی که اصولا در فایل ELF.bin انجام می‌گیرد، لازم است مکانیزم اعتبارسنجی هسته سیستم عامل، و به تبع آن مکانیزم اعتبار سنجی bootloader را از کار بیاندازد.



شکل ۷-دور زدن فرآیند اعتبار سنجی امضاء هسته سیستم‌عامل و بخش ماژول‌های iLO

شکل ۷ روند دور زدن فرآیند اعتبار سنجی امضاء هسته سیستم‌عامل و بخش ماژول‌های iLO را به‌طور خلاصه نشان می‌دهد [۲]. پس از استخراج سه بخش اصلی bootloader.bin، kernel.bin و ELF.bin از طریق مهندسی معکوس، آدرس توابعی که در دو بخش bootloader و kernel عملیات بررسی و اعتبار سنجی امضاء را انجام می‌دهند جستجو و با دستور NOP جایگزین می‌شوند. در مرحله آخر فایل‌های تغییر یافته در کنار یکدیگر تجمیع شوند تا یک فایل کامل با فرمت HP Image را تشکیل دهند و فایل ایجاد شده در حافظه فلش SPI نوشته می‌شود. پس از یک راه‌اندازی مجدد، می توان مشاهده نمود که سفت‌افزار آلوده به درب پشتی بدون مشکل بارگزاری می‌شود.

۴,۲,۲ بررسی سطح ماژول

۴,۲,۲,۱ بخش Boot Loader

طبق اطلاعات مندرج در جدول ۲ اختلاف میان دو بخش Boot Loader ۵ بایت است که تحلیل جزئی‌تر آن نشان می‌دهد که این اختلاف ناشی از تغییر تابع مسئول اعتبارسنجی امضاء بخش kernel و غیرفعال (NOP) کردن این فرآیند است (شکل ۸).

```
000038B4 64 FF FF EB    BL    sub_364C
000038B8 00 00 50 E3    CMP   R0, #0
000038BC 00 00 A0 E1    NOP
000038C0 56 0F 8F E2    ADR   R0, aSignatureValid ; "Signature valid\r\n"
000038C4 74 FF FF EB    BL    sub_369C
```

```
000038B4 64 FF FF EB    BL    sub_364C
000038B8 00 00 50 E3    CMP   R0, #0
000038BC 40 00 00 1A    BNE   loc_39C4
000038C0 56 0F 8F E2    ADR   R0, aSignatureValid ; "Signature valid\r\n"
000038C4 74 FF FF EB    BL    sub_369C
```

شکل ۸- غیرفعال کردن بخش اعتبارسنجی امضاء امضاء در بخش Bootloader

۴,۲,۲,۲ بخش Kernel

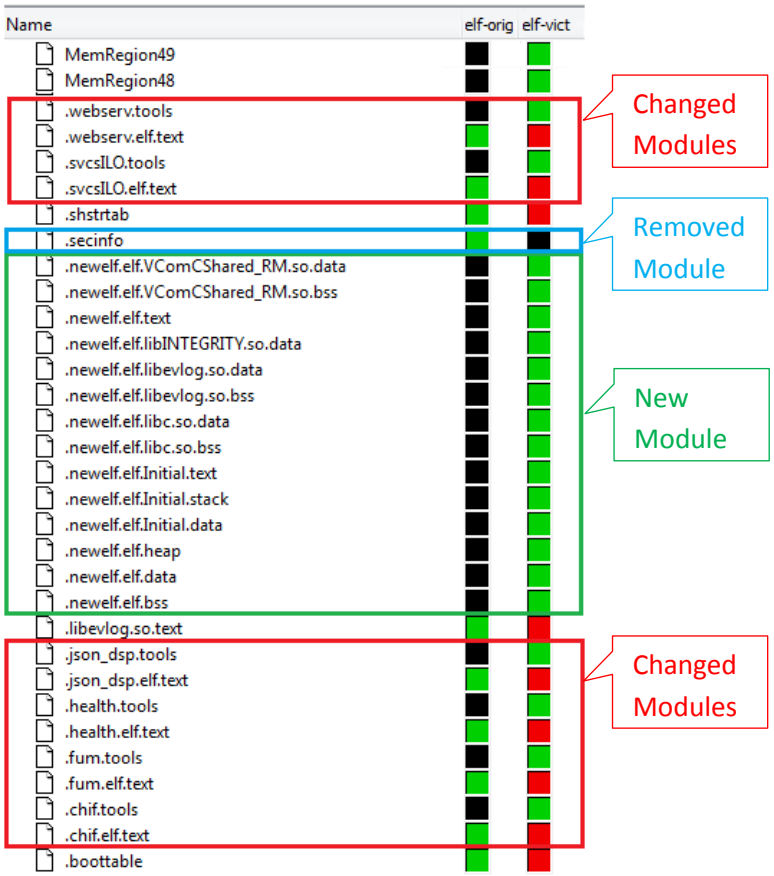
طبق اطلاعات مندرج در جدول ۲ اختلاف میان دو بخش کرنل ۱۲ بایت است که تحلیل جزئی‌تر آن نشان می‌دهد که این اختلاف ناشی از تغییر تابع مسئول اعتبارسنجی امضاء بخش userland (از طریق coprocessor) و غیرفعال (NOP) کردن این فرآیند در ۳ مکان است (شکل ۹).

```
ROM:000ADF14 08 50 82 E8    STMIA   R2, {R3,R12,LR}          ROM:000ADF14 08 50 82 E8    STMIA   R2, {R3,R12,LR}
ROM:000ADF18 00 30 A0 E3    MOV     R3, #0                   ROM:000ADF18 00 30 A0 E3    MOV     R3, #0
ROM:000ADF1C 10 3F 0D EE    MCR     p15, 0, R3,c13,c0, 0     ROM:000ADF1C 00 00 A0 E1    NOP
ROM:000ADF20 00 00 A0 E1    NOP                              ROM:000ADF20 00 00 A0 E1    NOP
ROM:000ADF24 00 00 A0 E1    NOP                              ROM:000ADF24 00 00 A0 E1    NOP
ROM:000ADF28 30 E0 90 E5    LDR     LR, [R0,#0x30]           ROM:000ADF28 30 E0 90 E5    LDR     LR, [R0,#0x30]
```

شکل ۹- غیرفعال شدن بخش اعتبارسنجی امضاء بخش userland در بخش Kernel

۴,۲,۲,۳ بخش UserLand

استخراج بخش User Land سفت‌افزار iLO و مقایسه محتوای آن با نسخه ۲.۳۰ اصلی ارائه شده توسط شرکت HP نشان‌دهنده حذف یک فایل (sectionInfo) و افزوده شدن یک ماژول جدید (با نام newELF به صورت ۱۷ بخش مجزا) به بخش User Land است (شکل ۱۰). علاوه بر اضافه شدن ماژول‌های جدید بدافزار، تعدادی از ماژول‌های موجود در نسخه اصلی نیز تغییر کرده‌اند.



شکل ۱۰- فایل‌های تغییریافته در سفت‌افزار سیستم آلوده

در این بخش مقایسه‌ای بین ماژول‌های نسخه اصلی و ماژول‌های تغییریافته در این بدافزار انجام گرفته است که نتایج آن در جدول ۳ به صورت کامل قابل مشاهده است.

جدول ۳- مقایسه نسخه iLO دارای NewELF با نسخه اصلی

File Name	Offset	Function	Old Op	New Op
Chif.ELF.text	0x280	sub_10070	MOV	BL
Chif.ELF.text	0x9c28	sub_19BD0	ADD	BL
Webserver.ELF.text	0x218	sub_10210	SUB	BL
Webserver.ELF.text	0x2af80	sub_3AEC4	BL	BL
Webserver.ELF.text	0x2b024	sub_3AEC4	BL	BL
Health.ELF.text	0x5840	sub_152AC	MOV	BL
Health.ELF.text	0x37b30	sub_47B1C	MOV	BL
Fum.ELF.text	0x4324	sub_13B68	BLNE	BL
Fum.ELF.text	0x43b8	sub_13B68	MOV	BL
Fum.ELF.text	0x4400	sub_13B68	BL	BL
Fum.ELF.text	0x4738	sub_14720	MOV	BL

Json_dsp.ELF.text	0x9ac	sub_109A4	SUB	BL
Json_dsp.ELF.text	0xa564	sub_19F48	BL	BL
Json_dsp.ELF.text	0x6324c	sub_73190	BL	BL
Json_dsp.ELF.text	0x632f0	sub_73190	BL	BL
SvcsiLO.ELF.text	0x1578	sub_114EC	MOV	BL
SvcsiLO.ELF.text	0x4310	sub_14300	BL	BL
SvcsiLO.ELF.text	0xf388	sub_1F378	MOV	BL

۴،۲،۳ بررسی فایل‌های ایجاد شده توسط بدافزار

بدافزار Implant.ARM.iLOBleed.a در داخل NAND Flash که فضای کاری و حافظه جانبی قابل خواندن نوشتن iLO است، ۳ فایل ایجاد می‌کند. به نظر می‌رسد که مسیر و حتی نام این فایل‌ها از طریق یک Configurator قابل تنظیم بوده است. در نسخه بدافزار موجود، این سه فایل با نام‌های lifesignal.bin، schedule.bin و fakefwdata.bin وجود دارند (جدول ۴).

جدول ۴- سه فایل استفاده شده در بدافزار

نام فایل	مسیر ذخیره	ماژول‌های مرتبط
lifesignal.bin	i:\vol0\logs\lifesignal.bin	Chif.tool – NewELF.ELF
schedule.bin	i:\vol0\logs\schedule.bin	NewELF
fakefwdata.bin	i:\vol0\logs\fakefwdata.bin	Fum.tool – NewELF.ELF

در صورتی که مدیر سیستم تلاش کند نسخه سفت‌افزار سرور خود را ارتقا دهد، بدافزار ضمن جلوگیری و شبیه‌سازی عملیات ارتقای سفت‌افزار جهت فریب مدیر سیستم، در فایل fakefwdata.bin اطلاعات این عملیات را درج می‌کند.

بررسی فایل schedule.bin در کنار سایر بخش‌های بدافزار نشان می‌دهد که درون این فایل دو عدد ۴ بایتی قرار دارد که بدافزار از محتوای آن برای برنامه ریزی حذف محتوای دیسک‌های سخت سرور استفاده می‌کند. عدد ۴ بایتی اول بیانگر شماره‌ای است که به یک مقدار اولیه مقداردهی می‌شود و سپس با هر بار اجرای فرآیند حذف اطلاعات دیسک، یک واحد از آن کم می‌شود تا به صفر برسد. عدد دوم نیز بیانگر تاریخ اجرای فرآیند پاک‌سازی دیسک است (شکل ۱۱).

```
</>hpiLO-> please hexdump i:/vol0/logs/schedule.bin

open(i:/vol0/logs/schedule.bin) = 5
 00 00 00 00 [redacted] b5 5e  .......^

[root@localhost ~]# date --date='@1588[redacted]
[redacted] 2020
[root@localhost ~]#
```

شکل ۱۱- محتویات فایل schedule.bin

۴,۲,۴ تحلیل ماژول newELF

این ماژول یک ELF کامل است که به انتهای Boottable اضافه شده و تعداد پردازه‌های آن را نیز به اندازه یک واحد افزایش داده است. این ELF همانند سایر ELFهای این سفت‌افزار، از چند بخش اساسی تشکیل شده است. اولین بخش آن NewELF.ELF.Initial.text است که برخلاف سایر ELFها خالی از محتوا نیست و در داخل آن کدهایی وجود دارد. با بررسی دقیقتر می‌توان دریافت که این بخش شباهت بسیار زیادی به ConAppCLI.ELF.text دارد که یکی از بخش‌های اصلی سفت‌افزار iLO است. در جدول ۵ مقایسه‌ای کلی بین این دو ماژول صورت گرفته است که تفاوت آن‌ها را نشان می‌دهد. این شباهت‌ها نشان می‌دهد که ساختار پایه بدافزار Implant.ARM.iLOBleed.a مبتنی بر توابع ماژول ConAppCLi است.

جدول ۵- مقایسه بین ConAppCLI.ELF.text و NewELF.ELF.Initial.text

Offset	Function	ConAppCli.ELF.text Opode	NewELF.ELF.Initial.text Opcode
0x74	sub_10070	STMFD	**STMFD**
0x7c	sub_10070	SUB	**SUB**
0x80	sub_10070	LDR	**LDMDB**
0x91c78	End of Old File	-	**Add Zero Buffer**

بخش دیگر و اساسی بدافزار، بخش NewELF.ELF.text است. شکل ۱۲ تابع اصلی اجرا کننده بدافزار را نشان می‌دهد. یکی از وظایف اصلی این تابع، مقداردهی یک ساختار داده است که پارامترهای اصلی اجرای عملیات بدافزار را تعیین می‌کند و بخشی از آن در شکل ۱۳ نشان داده شده است. در ابتدای این تابع آدرس فایل schedule.bin در یک متغیر کپی شده و اشاره‌گر این آدرس در آدرس 0x0c ساختار داده کپی می‌شود.

```
1  int MainOpration()
2  {
3    int v0; // r5
4    int v2; // [sp+0h] [bp-38h] BYREF
5    int schedule_bin_path_ptr[4]; // [sp+4h] [bp-34h] BYREF
6    char v4[12]; // [sp+14h] [bp-24h] BYREF
7    char v5; // [sp+20h] [bp-18h] BYREF
8
9    strcpy(schedule_bin_path_ptr, (int)aIVol0LogsSched_0, (int)&v2);// i://vol0//logs//schedule.bin
10   memcpy(v4, schedule_bin_path_ptr);
11   memfree(schedule_bin_path_ptr);
12   v0 = initOperationConfigFiles((int)v4);
13   if ( !MEMORY[0x5A38818](240) )              // scvcIlo: Check Config files exsitence
14   {
15     if ( startPeriodicOperation((int)v4) )
16       v0 = 19;
17     else
18       v0 = 0;
19   }
20   memfree(&v5);
21   return v0;
22 }
```

شکل ۱۲- تابع اصلی اجراکننده بدافزار

در ادامه، در تابع دیگری با نام initOperationConfigFiles، وضعیت فایل‌های مورد نیاز بدافزار بررسی می‌شود. در این تابع در ابتدا فایل lifesignal.bin ساخته می‌شود و سپس بنا به شرایط عملیات، در مورد فایل schedule.bin تصمیم‌گیری می‌شود؛ به این صورت که اگر فایل schedule.bin وجود داشته باشد و ساختار معتبری داشته باشد، محتوای آن در آدرس‌های 0 تا 0x07 ساختار داده کپی می‌شود و در فایل lifesignal.bin مقدار 0x3 نوشته می‌شود. در صورت عدم وجود این فایل در آدرس مربوطه، این فایل ایجاد می‌شود و با مقادیر اولیه پر می‌شود. پس از آن نیز ساختار داده به صورت کامل پر می‌شود. در شرایطی که فایل schedule.bin دارای ساختار نامعتبری باشد، در فایل lifesignal.bin مقدار 0x9 نوشته می‌شود.

```
struct operation_parameters {
    0    : int counter;
    0x04 : int startDate;
    0x08 : char commandBit;
    ...
    0x0C : char *schedule_bin_path;
    ...
    0x1C : int operationDate;
    ...
    0x24 : int maxOperationCount;
}
```

شکل ۱۳- ساختار داده پارامترهای عملیاتی بدافزار

همان‌طور که قبلا تشریح شد، فایل schedule.bin از دو عدد ۴ بایتی تشکیل شده است. در ابتدای عملیات، در تابع initOperationConfigFiles عدد شمارنده به مقدار بیشینه ممکن این شمارنده (آدرس 0x24 ساختار داده) و عدد تاریخ نیز بر روی ساعت و روز مورد نظر اجرای فرمان تنظیم می‌شود. در یک نمونه تحلیل شده از این بدافزار، مقدار بیشنیه تکرار عملیات روی 0x2 و در نمونه دیگر روی 0x3e8 (معادل ۱۰۰۰ مرتبه) تنظیم شده است.

پس از بررسی مناسب بودن شرایط اجرای عملیات تخریب، عملیات در تابع startPeriodicOperation آغاز می‌شود (شکل ۱۴). وظیفه این تابع در مرحله اول، ایجاد و پرکردن آرایه‌ای از ساختار داده عملیات با طول بیشینه مقدار اجرای عملیات است. هریک از این ساختار داده‌ها با مقادیر گوناگون زمان اجرای عملیات مقداردهی می‌شود. این مقداردهی به این صورت است که بدافزار مضارب گوناگون دوره تناوب اجرای عملیات (مثلا ۱۲ ساعت) را به یک زمان انتظار اولیه (مثلاً ۳۶ ساعت) اضافه می‌کند و این مقدار را در آدرس 0x1C هریک از ساختار داده‌ها ثبت می‌کند. در این صورت، عملیات پس از طی زمان انتظار (۳۶ ساعت) از زمان ثبت شده در فایل schedule.bin آغاز می‌شود و در هر دوره تناوب تکرار می‌شود. در نهایت تابع startWipeOperation فراخوانی می‌شود که عملیات تخریب دیسک را انجام می‌دهد.

```
int __fastcall startPeriodicOperation(int a1)
{
  int *v2; // r0
  int v3; // r1
  int *v4; // r8
  int *v5; // r12
  int *v6; // r7
  int *i; // lr
  int v8; // r8
  int *v9; // r0
  int *v10; // r3
  int v11; // t1
  int *v12; // t1
  int v13; // r5
  int *v15[2]; // [sp+4h] [bp-54h] BYREF
  int v16; // [sp+Ch] [bp-4Ch]
  int v17[3]; // [sp+14h] [bp-44h] BYREF
  char v18[16]; // [sp+20h] [bp-38h] BYREF
  int v19; // [sp+30h] [bp-28h]

  v16 = 0;
  v15[1] = &dword_20;
  v2 = (int *)sub_3F2D7E4(128);
  v3 = 0;
  v4 = v2;
  v5 = v2;                              int
  v6 = &dword_1158CC;
  v15[0] = v2;
  for ( i = 129600; ; i = v12 )                 // Perform operation every 12 hours
  {
    v8 = (char *)v4 - (char *)v5;
    v17[0] = (int)i;
    sub_3F32D2C(v15, v3 + 1, v17);
    v3 = v16;
    v5 = v15[0];
    if ( v16 - 1 > (unsigned int)(v8 >> 2) )
    {
      v9 = (int *)((char *)v15[0] + v8);
      v10 = &v15[0][v16 - 1];
      do
      {
        v11 = *--v10;
        v10[1] = v11;
      }
      while ( v9 != v10 );
    }
    else
    {
      v9 = (int *)((char *)v15[0] + v8);
    }
    *v9 = v17[0];
    v4 = v9 + 1;
    if ( v6 == &dword_116834[13] )
      break;
    v12 = (int *)v6[1];
    ++v6;
  }
  initOperationStruct((int)v17, a1, v15);
  free(v15[0]);
  v15[0] = dword_11586C;
  v13 = startWipeOperation((int)v17, (int (__fastcall ***)(_DWORD))v15);
  free(v19);
  memfree(v18);
  return v13;
}
```

شکل ۱۴– بدنه تابع startPeriodicOperation

تابع startWipeOperation در یک حلقه عملیات تخریب را انجام می‌دهد این تابع در شکل ۱۵ نشان داده شده است. در ابتدای این حلقه، تابعی به نام GetAndValidateOperationParameters صحت پارامترهای عملیات را بررسی می‌کند و

تعداد عملیات باقی مانده را محاسبه می‌کند. در حقیقت تعداد عملیات انجام شده در متغیری به نام SuccessfulOperationCount استخراج می‌شود و بررسی می‌شود که عدد بدست آمده از بیشینه مقدار شمارنده بیشتر نباشد.

```
int __fastcall startWipeOperation(int OperationParametersStructPointer, int (__fastcall ***WipeDisk)(_DWORD))
{
  int v3; // r7
  int v5; // r0
  _BOOL1 v6; // zf
  unsigned int SuccessfulOperationCount; // r0
  int OperationStartTime[5]; // [sp+4h] [bp-14h] BYREF

  if ( !WipeDisk )
    return 19;
  v3 = GetOperationStartTime(OperationParametersStructPointer, OperationStartTime);
  if ( v3 )
  {
    MEMORY[0x5A38128](240);                    // Operation Failed! Reset to default settings
    return v3;
  }
  while ( 1 )
  {
    SuccesfullOperationCount = GetAndValidateOperationParameters(OperationParametersStructPointer);
    if ( SuccessfulOperationCount >= *(_DWORD *)(OperationParametersStructPointer + 36) )
      break;
    WaitForNextOperationTarget(OperationParametersStructPointer, OperationStartTime, SuccessfulOperationCount ;
    v5 = (**WipeDisk)(WipeDisk);
    v6 = v5 == 24;
    if ( v5 != 24 )
      v6 = v5 == 0;
    if ( v6 )
      DecrementOperationCount(OperationParametersStructPointer);
  }
  return v3;
}
```

شکل ۱۵- بدنه تابع startWipeOperation

تابع بعدی تابع WaitForNextOperationTarget است که محتوای آن در شکل ۱۶ به نمایش درآمده است. وظیفه این تابع، ایجاد انتظار در حلقه اصلی تا زمان رسیدن به زمان اجرای عملیات بعد است. در زمان مقرر، این تابع از حلقه خارج خود شده و حلقه اصلی عملیات به کار خود ادامه می‌دهد.

```
1  int __fastcall WaitForNextOperationTarget(int a1, _DWORD *a2, int a3)
2  {
3    int v5; // r5
4    int result; // r0
5
6    v5 = 4 * a3;
7    while ( 1 )
8    {
9      result = GetCurrentTime() - *a2;
10     if ( result >= *(_DWORD *)(*(_DWORD *)(a1 + 28) + v5) )
11       break;
12     MEMORY[0x55C52F4](1);                          // Dealy_us
13   }
14   return result;
15 }
```

شکل ۱۶- بدنه تابع WaitForNextOperationTarget

در قسمت بعدی حلقه اصلی عملیات wipeDisk انجام می‌شود و دیسک سرور به صورت کامل پاکسازی می‌شود و اطلاعات آن حذف می‌شود. پس از اتمام عملیات تخریب، فایل schedule.bin توسط تابع DecrementOperationCount به‌روزرسانی شده و یک عدد از شمارنده آن کم می‌شود. بدنه تابع DecrementOperationCount در شکل ۱۷ نشان داده شده است.

```
1  int __fastcall DecrementOperationCount(int a1)
2  {
3    if ( !*(_BYTE *)(a1 + 8) )
4      return 16;
5    if ( !*(_DWORD *)a1 )
6      return 17;
7    --*(_DWORD *)a1;
8    return sub_3F29078();                    // Write New Counter to schedule.bin
9  }
```

شکل ۱۷- بدنه تابع DecrementOperationCount

۴,۲,۵ تحلیل ماژول‌های ابزاری

ماژول‌های ابزاری در این بدافزار به این دلیل در سفت‌افزار گنجانده شده‌اند که اختلالاتی را در کارکرد آن برنامه‌ها ایجاد کنند. البته گاهی نیز این امر به صورت اختلال دیده نمی‌شود بلکه عملی را خارج از عملکرد واقعی آن برنامه انجام می‌دهند. این بدافزار در حالت کلی ۶ ماژول ابزاری دارد که در زیر لیست شده‌اند:

جدول ۶- ماژول‌های ابزاری بدافزار

نام ماژول	شرح
.chif.tools	تغییر کانال تبادل پیام میان iLO و سرور
.fum.tools	دور زدن فرآیند به‌روز رسانی سفت‌افزار به منظور ابقاء سفت‌افزار آلوده
.webserver.tools	تغییر واسط وب مدیریتی برای نمایش اطلاعات نامعتبر نسخه سفت‌افزار iLO
.health.tools	تغییر ماژول ثبت رخدادهای سرور برای غیرفعال‌سازی ثبت رخدادهای مربوط به عملکرد بدافزار
.svcsiLO.tools	تغییر بخش چند نخی هسته سیستم‌عامل iLO
.json_dsp.tools	نامشخص

۵ جمع‌بندی مطالب

امنیت سفت‌افزار در سال‌های اخیر به عنوان یک موضوع مهم در امنیت فناوری اطلاعات مطرح می‌شود که در عمل توجه کافی به آن نمی‌شود. به دلیل امکانات و سطح دسترسی بالایی که ابزار مدیریتی HP iLO در اختیار دارد، نیازمند روش‌های محافظتی ویژه‌ای می‌باشد. متاسفانه نبود ابزارها و اطلاعات کافی و اختصاصی بودن محیط iLO باعث می‌شود بسیاری از محققین امنیت در بررسی این سیستم‌ها دست بسته باشند. بدتر از آن، با وجود اینکه پژوهش‌های منتشر شده توسط محققین امنیت در گذشته امکان‌پذیری قرارگیری بدافزار فرضی در این سفت‌افزار را بررسی کرده بود [۲]، همچنان راهکاری جهت کشف آلودگی و رفع آن در صورت اتفاق به صورت عمومی ارائه نشده است.

نکته مهم دیگر وجود راه‌های دسترسی و آلوده نمودن iLO هم از طریق شبکه و هم از طریق سیستم عامل میزبان می‌باشد. این مساله به این معناست که حتی با قطع کامل کابل شبکه iLO، باز هم امکان آلوده شدن و قرارگیری بدافزار در آن وجود دارد. جالب اینجاست که هیچ راهکاری برای خاموش نمودن یا غیرفعال نمودن کامل iLO در مواردی که نیازی به آن نیست نیز دیده نشده است.

این موضوعات ضرورت انجام اقدامات امنیتی پیشگیرانه جهت ارتقاء سطح امنیت سفت‌افزار مانند به‌روزرسانی به آخرین نسخه ارائه‌شده توسط سازنده، تغییر دوره‌ای رمزعبور کاربران و جداسازی شبکه iLO از شبکه عملیاتی و در نهایت پایش دوره‌ای وضعیت سفت‌افزار از منظر پارامترهای امنیتی و آلودگی‌های احتمالی بیش از پیش اهمیت دارد.

۵,۱ راهکارهای پیشنهادی حفاظت iLO

- ✓ عدم اتصال واسط شبکه iLO به شبکه عملیاتی و اختصاص یک شبکه کاملا مجزا
- ✓ به‌روز رسانی دوره‌ای نسخه سفت‌افزار iLO به آخرین نسخه رسمی ارائه شده توسط شرکت HP
- ✓ غیر فعال نمودن امکان دانگرید و انجام تنظیمات امنیتی iLO روی سرورهای نسل دهم HP
- ✓ استفاده از تجهیزات امنیتی دفاع در عمق جهت کاهش ریسک و کشف نفوذها قبل از رسیدن به iLO
- ✓ استفاده دوره‌ای از ابزار پویشگر iLO پادویش به منظور کشف آسیب‌پذیری‌ها، بدافزارها و درب‌های پشتی احتمالی در نسخه کنونی سفت‌افزار iLO سرور

۶ مراجع

[۱] F. Périgaud, A. Gazet, and J. Czarny, “*Subverting your server through its BMC*: the HPE iLO4 case,” Recon, 2018.

[۲] F. Périgaud, A. Gazet, and J. Czarny, “*Backdooring your server through its BMC: the HPE* iLO4 case,” SSTIC, 2018.

[۳] F. Périgaud, A. Gazet, and J. Czarny, “*Turning your BMC into a revolving door*,” ZeroNights, 2018.