Inseguridades de IoT

Por Gunnar Wolf

Gunnar Wolf es administrador de sistemas para el Instituto de Investigaciones Económicas de la UNAM y desarrollador del proyecto Debian GNU/Linux.
http://gwolf.org

Comienzo con la escritura de esta columna el 22 de octubre del 2016. Para quienes trabajamos en el campo de la seguridad informática el día de ayer fue particularmente intenso: Por un lado, nos enfrentamos con un tremendo ataque de denegación de servicio distribuido (DDoS) que dejó fuera de la red al proveedor de resolución de nombres Dyn DNS — con ello, a sitios del calibre de Twitter, Amazon, Whatsapp o Reddit. Y, por el otro, muchos administradores de sistema estuvimos atareados parchando nuestros sistemas por culpa de la "vaca sucia", conocida formalmente como CVE-2016-5195, una vulnerabilidad de escalación de privilegios en el núcleo de Linux.

Si bien no hay más relación entre estos dos hechos que la casualidad de su fecha de publicación, es muy probable que su impacto a futuro se mantenga entrelazado.

Se ha hablado del internet de las cosas en muchos contextos, mayormente positivos, especulando con anticipación acerca de la forma que tomará el futuro de nuestro campo (tanto hacia adentro, a cómo los profesionales lo afrontamos; como hacia afuera, cómo será el impacto hacia los usuarios y la sociedad en general). Sin ir más lejos, el número 51 de Software Gurú fue dedicado íntegro a este tema.

Como en todo tema que especula acerca del futuro, siempre hay algún pesimista que le ve el lado tenebroso. En este caso, traigo a referencia un par de artículos recientemente escritos por Matthew Garrett:

En el primero de estos artículos [1], Matthew describe cómo –por mera curiosidad por ver lo que había conectado a un cable Ethernet en un hotel– pudo asumir el control de las lámparas y cortinas de todos los cuartos del hotel. El segundo [2] describe algunos ataques que lanzó contra focos inteligentes que compró para fines de mera investigación... pero que podrían volver locos a muchos usuarios menos involucrados en el tema que él.

¿A QUÉ VIENE ESTO AL CASO?

El principal factor característico del ataque del 21 de octubre fue que provino de una botnet (red de equipos vulnerados, controlados por un mismo atacante) generada mediante el programa atacante Mirai, palabra que significa futuro en japonés. Mirai es un programa que busca y explota vulnerabilidades conocidas en las configuraciones por default de diversos sistemas del rango del Internet de las cosas — principalmente, cuentas abiertas por omisión. De hecho, el código fuente de Mirai está disponible en github [3].

Cabe mencionar que Mirai (cuyo código se hizo público varios días antes del ataque en cuestión), fue empleado también en el ataque al sitio Web del investigador en temas de seguridad informática Paul Krebs [4]. En ambos casos, la cantidad de tráfico generado rondaba los 600Gbps.

La forma en que Mirai obtiene control de los dispositivos es bastante sencilla y poco sofisticada. No se centra en aprovechar vulnerabilidades de validación de entradas, como la mayor parte de los gusanos; tampoco aprovecha mega-agujeros comunes, para los cuales se ha convertido en moda una publicación con todo y nombre y logotipo (piensen en Heartbleed para OpenSSL, Badlock para Windows y Samba, y el ya mencionado Dirty CoW para Linux). No, lo que hace Mirai es mucho menos sofisticado, ya que el grueso del ataque se centra en intentar autenticarse como administrador usando contraseñas comunes o predeterminadas (ej. admin/admin). La figura 1 muestra un pedazo del código de bot/scanner.c donde se aprecia esto.

```
add_auth_entry("\x50\x4D\x40\x56", "\x43\x46\x4F\x4B\x4C", 8);                  // root      admin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C", 7);             // admin     admin
add_auth_entry("\x50\x4D\x40\x56", "\x1A\x1A\x1A\x1A\x1A\x1A", 6);             // root      888888
add_auth_entry("\x50\x4D\x40\x56", "\x54\x4F\x4A\x46\x4B\x52\x41", 5);         // root      xmhdipc
add_auth_entry("\x50\x4D\x40\x56", "\x4B\x47\x44\x43\x57\x4E\x56", 5);         // root      default
add_auth_entry("\x50\x4D\x40\x56", "\x4B\x57\x43\x4C\x56\x47\x41\x4A", 5);     // root      juantech
add_auth_entry("\x50\x4D\x40\x56", "\x13\x10\x11\x16\x17\x14", 5);             // root      123456
add_auth_entry("\x50\x4D\x40\x56", "\x17\x16\x11\x10\x13", 5);                 // root      54321
add_auth_entry("\x51\x57\x52\x52\x4B\x58\x56", "\x51\x57\x52\x52\x4B\x58\x56", 5); // support   support
add_auth_entry("\x50\x4D\x40\x56", "", 4);                                     // root      (none)
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51\x55\x4D\x50\x46", 4); // admin     password
add_auth_entry("\x50\x4D\x40\x56", "\x58\x4D\x4B\x56", 4);                     // root      root
add_auth_entry("\x50\x4D\x40\x56", "\x13\x10\x11\x16\x17", 4);                 // root      12345
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3);                     // user      user
add_auth_entry("\x43\x46\x4F\x4B\x4C", "", 3);                                 // admin     (none)
add_auth_entry("\x50\x4D\x40\x56", "\x52\x43\x51\x51", 3);                     // root      pass
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C\x13\x10\x11\x16", 3); // admin     admin1234
add_auth_entry("\x50\x4D\x40\x56", "\x13\x13\x13\x13", 3);                     // root      1111
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x51\x4F\x41\x43\x46\x4F\x4B\x4C", 3); // admin     smcadmin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x13\x13\x13", 2);                 // admin     1111
add_auth_entry("\x50\x4D\x40\x56", "\x14\x14\x14\x14\x14\x14", 2);             // root      666666
add_auth_entry("\x50\x4D\x40\x56", "\x52\x43\x51\x51\x55\x4D\x50\x46", 2);     // root      password
add_auth_entry("\x50\x4D\x40\x56", "\x13\x10\x11\x16", 2);                     // root      1234
```

Figura 1. Ejemplo de contraseñas que intenta usar Mirai.

Y es aquí donde realizamos la conexión con el maravilloso mundo del IoT...

Al iniciar el diplomado de Linux en sistemas embebidos de la Facultad de Ingeniería en la UNAM, uno de los primeros temas que abordamos es, naturalmente, qué es un sistema embebido. Y, sin entrar en detalles y disquisiciones, uno de los principales criterios es que es un sistema en el que, independientemente del tamaño físico, la potencia

de cálculo u otros factores; la principal función (esto es, la principal característica que se presenta al usuario) no es la de una computadora de propósito general y en su uso diario podemos incluso olvidarnos de la computadora que incluye. Posiblemente el usuario deba pasar por una fase de configuración, pero el dispositivo en su uso diario pasa inadvertido como parte de la funcionalidad provista por un aparato completamente distinto.

De este modo, por ejemplo, podríamos clasificar de sistema embebido a:

- La compleja red de sensores y actuadores que hay dentro de un automóvil vendido en los últimos veinte años.

- Se impone el ejemplo eterno de un refrigerador inteligente que facilite nuestra vida prediciendo lo que requerimos comprar o tirar (si bien la cantidad de estos dispositivos se mantiene en lo risible). Los refrigeradores inteligentes que he visto, sin embargo, se acercan al límite de esta definición, pues incorporan una pantalla con una instalación de un entorno Android completo.

- Las cámaras de vigilancia y sistemas de portero automático controlables desde un teléfono celular (incluso si éste no está en la casa en cuestión).
  Los sensores biométricos y asistentes de salud vestibles.

- Los teléfonos VoIP, que ya se han convertido en la norma en entornos corporativos.

- Los ya mencionados focos inteligentes y demás soluciones de automatización y centralización de mando en un edificio.

Y, claro, un universo de dispositivos que va creciendo imparablemente, y ya superan ampliamente a las computadoras de uso frontal. En esta lista no entrarían los dispositivos de cómputo móvil, como las tabletas y los celulares, pues para éstos, su principal función (muy por encima de la que hasta hace diez años era la primaria, mantener una conversación de voz entre dos participantes) es indefectiblemente proveer una interfaz de cómputo.

Una diferencia fundamental, de cierto modo implícita en la definición, radica en lo relativo a la flexibilidad del sistema: dado que no se trata de dispositivos de uso genérico, el conjunto de programas que ejecutan podría mantenerse estable a largo plazo. Los diseñadores del software que va a manejarlo buscan ofrecer un entorno estable y duradero…y no siempre consideran brindar funcionalidad para la actualización, ya sea por el espacio adicional que esto requeriría o por considerarlo como un factor que complica la vida del público objetivo.

Esto, claro está, no es nuevo. El mismo problema se presentaba hace diez o veinte años — Pero si los programas que controlan el funcionamiento de mi cámara digital (producida en 2005), proyector de video (del 2008) o impresora (una reliquia del 2001, todavía perfectamente funcional) tienen vulnerabilidades que los hacen susceptibles a que un atacante se apodere de su funcionamiento…¿Cuál es el riesgo real? Dado que son equipos que no cuentan con ningún tipo de conexión a red, tan bajo que resulta despreciable.

Los dispositivos IoT dependen de una conexión a Internet, no para su función principal, pero sí para funcionalidad altamente deseable para el usuario. Cómo actualizar a estos dispositivos ante una vulnerabilidad es, por decir lo menos, un problema complejo — Si se hace de forma automática, los problemas en que incurriría su compañía en caso de que una actualización no funcionara correctamente no serían asunto nada menor.

Y…toca volver a las contraseñas. A pesar de que el usuario que busca expresamente equipos IoT probablemente pase al menos algunos minutos familiarizándose con su configuración, cambiando contraseñas y ajustando funcionalidad, la experiencia demuestra que una amplia proporción de usuarios simplemente enchufan el dispositivo y se olvidan de él o lo dejan con las configuraciones predeterminadas en aras de no tener que buscar los manuales a futuro.

Como profesionales del desarrollo de software, muy probablemente nos tocará diseñar sistemas que caigan en la definición del Internet de las Cosas. No podemos olvidar nuestra responsabilidad ante el mundo. Desplegar masivamente un sistema con contraseñas fijas (o incluso generadas algorítmicamente alrededor de un valor visible desde fuera, como la dirección MAC) es una invitación a que nuestros equipos sean víctimas de mal uso. No proveer interfaces confiables y bien probadas para la actualización es un pecado igualmente grave.

La usabilidad de Internet a futuro depende de cada uno de nosotros.

Referencias

[1] M. Garrett. "I stayed in a hotel with Android light switches and it was just as bad as you'd imagine". https://mjg59.dreamwidth.org/40505.html

[2] M. Garrett. "I've bought some more awful IoT stuff". https://mjg59.dreamwidth.org/43486.html

[3] https://github.com/jgamblin/Mirai-Source-Code/

[4] "Source for IoT botnet mirai released". Krebs on Security. https://krebsonsecurity.com/2016/10/source-code-for-iot-botnet-mirai-released/