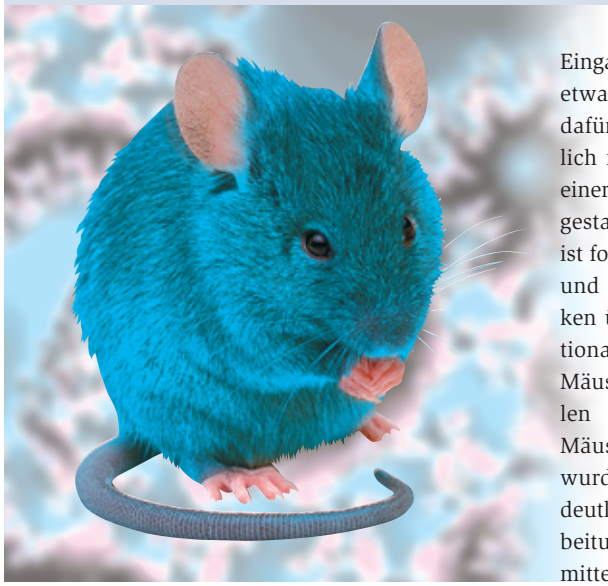
Bluetooth-HID im Praxistest

Blaue Maus

Herkömmliche drahtlose Tastaturen und Mäuse, ob per Infrarot oder Funk angebunden, sind störanfällig und leicht auszuspionieren. Mit dem erweiterten Bluetooth-Stack von Blue-Z lassen sich jetzt auch Bluetooth-Tastaturen und -Mäuse unter Linux einsetzen, die Datenspione vor größere Probleme stellen. Nils Faerber



Drahtlos angebundene Tastaturen und Mäuse sind ein nicht zu unterschätzendes Sicherheitsrisiko – nicht nur die Benutzer geben ihr Passwort über die Tastatur ein, auch der Administrator. Infrarot-Tastaturen sind in der Reichweite zwar beschränkt, aber sehr leicht abzuhören. Technisch etwas aufwändiger ist das Abhören von Funktastaturen, dafür ist ihr Signal aber selbst aus großer Entfernung noch zu empfangen.

Kanalwechsel

Mit Bluetooth-Tastaturen ist es ohne spezielle Geräte kaum noch möglich, die Verbindung zu verfolgen, denn Bluetooth-Geräte wechseln 1600-mal pro Sekunde den Kanal. Die Bluetooth Special Interest Group (Bluetooth-SIG, [http://www.bluetooth.com]) hat zudem bei der Entwicklung des HID-Standards (Human Interface Devices) eine Verschlüsselung zwischen dem Host und dem jeweiligen Gerät vorgesehen.

Eingabegeräte werden unter Linux noch etwas stiefmütterlich behandelt. Grund dafür ist, dass der klassische PC lediglich mit genau einer Tastatur an genau einer speziellen Schnittstelle (PS/2) ausgestattet war. Die Tastaturunterstützung ist folglich recht hart im Kernel integriert und niemand machte sich groß Gedanken über eine Abstraktion dieser Funktionalität. Anders liegt der Fall bei den Mäusen, neben der traditionellen seriellen Maus gab es Bus-Mäuse, PS/2-Mäuse und viele andere Typen. Hier wurde die Schnittstelle von Anfang an deutlich flexibler gehalten und die Verarbeitung des Mausprotokolls konsequent mittels XFree86 und »gpm« in den Userspace verlegt.

Die Tastaturbehandlung verblieb aber im Kernel und sorgte mit dem Aufkommen der USB-Tastaturen erstmals für Probleme – auf einmal gab es mehrere Tastaturschnittstellen. Die Lösung war der Input Core, eine Kernel-Abstraktion für Eingabegeräte. Für USB-Geräte klappte das auch recht gut. Man hatte nun einen generischen Kernel-Input-Treiber, auf den andere Kerneltreiber aufsetzten. Ein USB-Tastaturtreiber schreibt so über den Aufruf von Input-Core-Funktionen Tastatur-Events in die Kernel Keyboard Input Queue.

Bluetooth-HID im Userspace

Die Architektur des Input Core führt bei Bluetooth-HID-Geräten jedoch zu einem grundsätzlichen Problem: Wie bei allen Bluetooth-Geräten wird bei den HIDs eine Verbindung vom Rechner zum Bluetooth-Gerät hin aufgebaut. Anschließend kommuniziert das HID-Gerät mit dem Host über ein bestimmtes Protokoll des Bluetooth-L2CAP-Layers. Eine so komplexe Kommunikation im Kernel-Kontext zu behandeln – die Komplexität ist etwa vergleichbar mit einer PPP-Verbindung – ist wenig sinnvoll. Vielmehr wäre die sauberste Lösung, einen Userspace-Prozess zu verwenden und lediglich das Ergebnis, die Input Events, an den Kernel zu übergeben.

Event-Schnittstelle

Eine solche Schnittstelle zur Übergabe von Input Events an den Kernel gibt es bisher nicht. Die Entwickler des Linux-Bluetooth-Stack [1] haben daher den »User level driver support« zum Input Core hinzugefügt. Der User Level Driver (kurz Uinput) stellt dem Userspace über ein Character Device mit Major-Nummer 10 und Minor-Nummer 223 (im Devfs »/dev/misc/uinput«) den Zugriff auf den Input Core zur Verfügung. Mit diesem Device ist es möglich, aus dem Userspace heraus neue virtuelle Input Devices zu registrieren und entsprechende Events zu erzeugen.

Für die Bluetooth-HID-Geräte erledigt das der neue Daemon »bthid«, er ist die Schnittstelle zwischen dem Kernelinterface »uinput« und dem Bluetooth-HID-Protokoll. Der Daemon öffnet die Bluetooth-Verbindung zur Tastatur oder Maus, empfängt den Datenstrom und generiert die entsprechenden Input Events für das Linux Input Core Subsystem.

Voraussetzungen

Als Voraussetzung muss der Linux-Bluetooth-Stack von Blue-Z installiert und funktionsfähig sein. Für den Uinput-Treiber ist das Kernelpatch von Marcel Holt-

manns Website [2] erforderlich, was wiederum erfordert die Kernelmodule neu zu übersetzen. Das Patch gab es bei Redaktionsschluss für die Kernel 2.4.23 und 2.4.24. Nach dem Laden von »uinput.o« mittels »modprobe« erscheint auf Systemen mit Devfs unter »misc« automatisch das Character Device »uinput«; wer Devfs nicht verwendet, muss das Device von Hand anlegen:

```
mkdir -p /dev/misc
mknod /dev/misc/uinput c 10 223
```

Von der Blue-Z-Homepage [3] sind noch die Quellen der »libbluetooth2« und der Blue-Z-Utilities 2 direkt aus dem CVS herunterzuladen, namentlich sind es die beiden CVS-Module »libs2« und »utils2«. Die Befehle

```
./bootstrap
./configure
make
make install
```

(zunächst im Verzeichnis »libs2« und dann in »utils2« eingegeben) übersetzen und installieren die Quellen automatisch. Aus den Blue-Z-Utilities 2 sind die Programme »hid2hci« und »bthid« besonders interessant, sie sind für Bluetooth-HID-Geräte zuständig.

Schlüssel zur Welt

Zurzeit ist es noch so, dass eine Bluetooth-Tastatur oder -Maus Zusatzgeräte zum PC sind, es in der Regel also zumindest noch eine PS/2-Tastatur gibt. Aber man sollte den Rechner auch ausschließlich mit der Bluetooth-Tastatur bedienen können, auch zur Bootzeit.
Um nicht jedem PC-Bios einen Bluetooth-Stack verpassen zu müssen, hilft ein Trick: Praktisch jedes moderne Bios lässt sich bereits mit USB-Tastatur bedienen. Mit einer Zusatzsoftware für die Firmware der Bluetooth-USB-Dongles arbeitet das Dongle als so genannter HID-Proxy und meldet die Bluetooth-HID-Geräte beim System als USB-Geräte an. Die meisten derzeit angebotenen Komplettsets – bestehend aus Bluetooth-Tastatur und Dongle – beherrschen den HID-Proxy-Modus.

Um das Dongle und die Bluetooth-Tastatur wieder als Bluetooth-HID-Geräte verwenden zu können, ist zunächst das Dongle in den Bluetooth-USB-HCI-Modus (Host Controller Interface) umzuschalten. Das erledigt »hid2hci«; Root ruft es ohne weitere Kommandozeilenoptionen auf, es meldet bei erfolgreicher Umschaltung etwa Folgendes:

```
Switching device 0a12:1000 to HCI mode ➥
was successful
```

Derzeit unterstützt »hid2hci« viele Dongles mit CSR-Cipsatz (Cambridge Silicon Radio) sowie einige Logitech-Adapter. Ist der HCI-Daemon gestartet, sollte nun ein neues Bluetooth-Device verfügbar sein und das Blue-Z-Tool »hciconfig« beim Aufruf ohne Parameter etwa Folgendes ausgeben:

```
hci0:   Type: USB
  BD Address: 01:05:62:81:E2:CF ACL MTU: ➥
192:8  SCO MTU: 64:8
  UP RUNNING PSCAN ISCAN
  RX bytes:107 acl:0 sco:0 events:14 ➥
errors:0
  TX bytes:299 acl:0 sco:0 commands:13 ➥
errors:0
```

Marktübersicht: Bluetooth-Tastaturen und -Mäuse

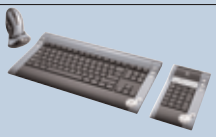
	Hersteller	Modell	Lieferumfang	Bezugsquelle	Preis
	Apple	Wireless Keyboard	Tastatur	Apple Store [http://www.apple.de]	80 Euro
	Apple	Wireless Mouse	Maus	Apple Store [http://www.apple.de]	80 Euro
	Elito-Epox	BT-Barton	Tastatur, Maus, USB-Dongle	Flexist [http://www.flexist.de]	100 Euro
	Logitech	Cordles MX Desktop Bluetooth	Tastatur, Maus mit Bluetooth-Adapter	Flexist [http://www.flexist.de]	145 Euro
	Logitech	Di-Novo Bluetooth	Tastatur, Maus mit Bluetooth-Adapter	Alternate [http://www.alternate.de]	250 Euro
	Logitech	MX900	Maus	Flexist [http://www.flexist.de]	95 Euro
	Microsoft	Wireless Intellimouse Explorer Bluetooth	Maus	Alternate [http://www.alternate.de]	75 Euro
	Microsoft	Wireless Desktop Optical Bluetooth	Tastatur, Maus mit Bluetooth-Adapter	Alternate [http://www.alternate.de]	140 Euro

Abbildung 1: Das Bluetooth-Kit BT-Barton von Elito-Epox besteht aus Tastatur, Maus und USB-Dongle. Beim Systemstart emuliert das Dongle eine USB-Tastatur, das Programm »hid2hci« schaltet später um in den HCI-Modus.

Für den Test stand dem Autor das Bluetooth-HID-Kit Barton von Elito-Epox zur Verfügung (Abbildung 1). Das Set enthält die Tastatur BT-KB01B, die Maus BT-MS02B und das USB-Dongle BT-DG03BF. Das Dongle ist mit einem der erwähnten CSR-Chipsätze ausgestattet, weshalb es unmittelbar nach dem Anschluss als USB-Tastatur und -Maus erkannt wird. Nach dem Umschalten in den HCI-Modus mittels »hid2hci« verhält es sich aber wie ein ganz normales Bluetooth-USB-Dongle.

Sichtbar und unsichtbar

Tastatur und Maus sind normalerweise für andere Bluetooth-Geräte nicht erkennbar. Zum Einbinden der beiden Geräte muss daher ein Knopf auf der Unterseite des jeweiligen Geräts gedrückt werden, um es sichtbar (discoverable) zu machen. Der Befehl »hcitool scan« zeigt die Bluetooth Device Address sowie bei der Elito-Epox-Tastatur auch das Tastaturmodell an:

```
Scanning ...
  01:05:62:81:DC:DD    BT-KB01B 80DBDC
```

Die Maus meldet sich jedoch nicht mit Namen, weshalb man die Bluetooth-Geräte am besten nacheinander sichtbar macht – das erleichtert die Zuordnung:

```
Scanning ...
  01:05:62:81:DB:73        n/a
  01:05:62:81:DC:DD        BT-KB01B 80DBDC
```

Im nächsten Schritt wird der Bluetooth-HID-Daemon mit dem Befehl »bthid -d« gestartet. Der Prozess legt sich selbst in den Hintergrund und wickelt von dort die Kommunikation mit den Bluetooth-HID-Geräten ab. Die Anmeldung von Bluetooth-Tastatur und Maus läuft so ab, dass »bthid« die Bluetooth Device Address (BD-Address) des betreffenden Geräts erhält.

Die folgenden beiden Befehle registrieren zuerst die Tastatur und anschließend die Maus beim USB-Dongle:

```
bthid -c 01:05:62:81:DC:DD
bthid -c 01:05:62:81:DB:73
```

Der Daemon »bthid« schreibt allerdings keine Meldungen auf die Standardausgabe, sondern ins Systemlog, hier ein Auszug:

```
bthid[15517]: Bluetooth HID service ➥
started
bthid[15517]: Connected: Epox ➥
HIDEngine Keyboard (ffff:0000)
bthid[15846]: Connected: CSR HIDEngine ➥
Three Button Mouse (ffff:0000)
```

Die Verbindung zu Tastatur und Maus ist zustande gekommen, beide Geräte wurden korrekt erkannt.

Alle zugleich

Der Kernel wirft trotz Input Core die Eingaben aller Tastaturen in einen Topf, es ist also nicht möglich, die Eingaben über mehrere Tastaturen zu unterscheiden – egal ob Bluetooth, USB oder PS/2. Deshalb ist es auch nicht möglich, an einem Rechner mehrere Arbeitsplätze zu schaffen, indem man einfach mehrere Monitore und Tastaturen anschließt. Anders bei Mäusen: Hier bekommt jede Maus ein eigenes Device, im Devfs sind das »/dev/input/mouse*X*«. In der Konfigurationsdatei von XFree86 steht dann einfach das Mouse Device.

Die Mouse Events lassen sich aber ebenfalls zusammenfassen wie bei Tastaturen. Kernel 2.4 bietet dazu das Device »/dev/input/mice« an. Diese Möglichkeit ist besonders für Notebookbesitzer interessant, die normalerweise mit einer externen Maus arbeiten: Sie tragen »/dev/input/mice« in die XFree86-Konfigurationsdatei ein und können anschließend sowohl das interne Trackpad als auch die externe Maus benutzen. So funktioniert die Maus unter X auch dann, wenn die externe Maus nicht angeschlossen ist.

Ecken und Kanten

Bluetooth-HID hat unter Linux noch seine Ecken und Kanten, beim Blue-Z-HID ist noch mit einigen Änderungen zu rechnen. So ist der Daemon »bthid« noch unvollständig, einige Funktionen, etwa die Suche nach HID-Geräten oder die Anzeige der aktuellen Verbindungen, funktionieren derzeit nicht. Auch die Umschaltung des Dongle aus dem USB-HID-Proxy-Modus in den HCI-Modus läuft noch nicht mit allen Chipsätzen – hier ist Unterstützung der Hersteller dringend gefragt.

Ein weiteres Problem ist die Sicherheit. Bei Redaktionsschluss war es noch nicht möglich, ein Pairing zwischen Tastatur und Host durchzuführen – ohne wird aber kein Verbindungsschlüssel erzeugt und die Verbindung bleibt unverschlüsselt. Nun ist es bei 1600 Kanalwechseln pro Sekunde sehr schwer, den Datenverkehr zwischen zwei Bluetooth-Geräten zu belauschen, doch möglich ist es. Wenn die Gerüchte stimmen, gibt es Firmware-Versionen für die CSR-basierten Chipsätze, die tatsächlich das Abhören von Bluetooth Verbindungen zulassen – normale Bluetooth-Dongles können dies definitiv nicht.

Zusammen mit dem neuen Blue-Z-Maintainer Marcel Holtmann arbeitet der Autor dieses Artikels zurzeit an einer Erweiterung von »hcitool«, um Pairing und Verschlüsselung mit Bluetooth-Geräten zu ermöglichen. Eventuell gibt es also bei Erscheinen dieses Artikels eine neue Version der Blue-Z-Utilities. *(mdö)* ■

Infos

[1] Linux-Bluetooth-Stack von der Blue-Z-Homepage: [http://www.bluez.org]
[2] Marcel Holtmanns Blue-Z-Kernelpatches: [http://www.holtmann.org/linux/kernel/]
[3] Blue-Z-CVS-Zugriff und -Archiv: [http://www.bluez.org/cvs.html]