Der große Bruch?

Was der 35C3-Hack von Bitcoin-Hardware-Wallets bedeutet

Auf dem 35C3 zeigten drei Hacker, wie sie in verbreitete Hardware-Wallets für Kryptowährungen einbrachen und die Schlüssel stahlen. Doch was bedeutet das für die Praxis?

Von Mirko Dölle

Auch bei Hardware-Wallets für Kryptowährungen gibt es keine absolute Sicherheit. Das zeigten die Hacker Thomas Roth, Dmitry Nedospasov und Josh Datkoauf eindrucksvoll dem 35. Chaos Communication Congress in Leipzig. Sie manipulierten die bekanntesten Hardware-Wallets für Kryptowährungen, Trezor One, Ledger Nano S und Ledger Blue, und hätten auf diese Weise Bitcoins stehlen können. Die Lehre daraus ist jedoch nicht, dass man keine Hardware-Wallets verwenden sollte, sondern, dass man auch mit Hardware-Wallets sorgsam umgehen muss, um sein Vermögen zu schützen.

Der dreiteilige Vortrag zeigte unter anderem, wie leicht sich Siegel von Originalverpackungen entfernen und die Gehäuse der Hardware-Wallets spurlos öffnen lassen. Einmal geöffnet, bauten die Hacker Funkschnittstellen ein, um die Daten abzugreifen, oder manipulierten die Firmware so, dass sie beliebige Transaktionen automatisch signieren lassen konnten. Beim Ledger Blue konnten sie sogar die PIN, die das Gerät vor unberechtigter Benutzung schützt, von einem unmodifizierten Gerät über einige Distanz mitlesen. Möglich wurden diese Angriffe durch offene Debugging- und Programmierschnittstellen sowie durch Schwachstellen im Design der Hardware-Wallets.

Schnitt- und Schwachstellen

Die Hersteller der Hardware-Wallets machten es den Hackern einfach: So ist die Programmierschnittstelle des Mikrocontrollers beim Ledger Nano S auf der Rückseite der Platine herausgeführt und standardmäßig aktiviert, sodass sie letztlich ihre eigene Firmware aufspielen und somit die vollständige Kontrolle über das Hardware-Wallet übernehmen konnten.

Eine weitere Schwachstelle der Hardware-Wallets von Trezor und Ledger ist, dass sie ein sogenanntes Secure Element verwenden, das lediglich eine Datenleitung besitzt: den ST31-Chip. Der ST31 ist an sich erprobt und kommt zum Beispiel bei Pay-TV-Decodern zum Einsatz – doch für ein Hardware-Wallet ist er nur bedingt geeignet, denn er hat keine Anschlüsse für ein Display oder Tasten.

Der ST31 bewahrt den Seed des Wallet auf und signiert die Transaktionen. Für die Ansteuerung ist jedoch ein herkömmlicher Mikrocontroller zuständig. Dieser erzeugt die Transaktion, zeigt die Transaktionsdaten auf dem Display des Hardware-Wallets an, reicht sie an den ST31 weiter und teilt dem ST31 mit, ob der Benutzer die Transaktion autorisiert hat. Dann liefert der ST31 die fertig signierte Transaktion an den Mikrocontroller zurück.

Das Problem ist, dass der ST31 nicht wissen kann, welche Informationen der Mikrocontroller auf dem Display angezeigt und ob der Anwender tatsächlich die Transaktion autorisiert hat – er muss darauf vertrauen, dass der vorgelagerte Mikrocontroller nicht manipuliert wurde. Genau das schafften die Hacker jedoch, sie veröffentlichten sogar eine Firmware mit dem Spiel Snake für den Ledger Nano S.

Problemfall Seed

Beim Trezor One gelang es den Hackern sogar, den Seed auszulesen, der dafür gedacht ist, bei einem Defekt ein Ersatzgerät in Betrieb zu nehmen. Der Seed ist der heilige Gral für jeden Angreifer, denn damit erhält er dauerhaft vollständigen Zugriff auf das Bitcoin-Wallet – und zwar unabhängig vom Hardware-Wallet selbst und auch noch Monate nach dem Angriff. Der weitreichende Zugriff resultiert daraus, dass heute nur noch sogenannte Hierarchical Deterministic Wallets verwendet werden.

Ursprünglich erzeugten Bitcoin-Clients lediglich einen privaten und einen

öffentlichen Schlüssel für ein Wallet. Aus dem öffentlichen Schlüssel wurde dann per Hash-Funktion die Bitcoin-Adresse berechnet. Die Adresse ist also eine verkürzte Darstellung des Schlüssels. Um einen Bitcoin zu überweisen, muss der Absender die Transaktion mit seinem privaten Schlüssel signieren. Die Bitcoin-Nodes überprüfen dann anhand des öffentlichen Schlüssels, ob Transaktion und Signatur korrekt sind und zur Bitcoin-Adresse des Absenders passen – erst dann leiten sie die Transaktion an die Miner weiter.

Ließ man den Bitcoin-Client eine neue Bitcoin-Adresse anlegen, wurde dazu auch ein neues Public-Private-Schlüsselpaar erzeugt. Das Wallet bestand also aus einer Sammlung unzusammenhängender privater und öffentlicher Schlüssel. Wollte man sein Wallet sichern, musste man sämtliche Schlüssel aller benutzten Bitcoin-Adressen kopieren, um später wieder an seine Bitcoins heranzukommen.

Doch das Backup funktionierte nur so lange, bis man eine weitere Bitcoin-Transaktion durchführte: Das Bitcoin-Protokoll sieht vor, dass jede Bitcoin-Adresse nur ein einziges Mal benutzt wird, um Geld auszugeben. Das Wechselgeld, also die Differenz zwischen dem aktuellen Guthaben und dem Überweisungsbetrag, wird auf eine neue Bitcoin-Adresse überwiesen – die neu generierte Schlüssel besitzt. Die alte Adresse, deren Schlüssel man im Backup gesichert hat, ist leer – das Backup damit nutzlos.

Dies führte dazu, dass Anwender dauerhaft immer wieder dieselbe Bitcoin-Adresse benutzten. Auch wenn das von Satoshi Nakamoto so nicht gedacht war, funktioniert dieser Trick in der Praxis bis heute. Allerdings ist das riskant, denn es bietet Angreifern die Möglichkeit, Transaktionen zu duplizieren.

Bei den heute üblichen deterministischen Wallets hingegen erzeugen Bitcoin-Clients, aber auch Hardware-Wallets, einen Seed mit einer Länge von 128 oder 256 Bit. Dieser Seed wird durch eine Zählvariable ergänzt, anschließend ein Hashwert gebildet und erst daraus privater und öffentlicher Schlüssel und somit eine Bitcoin-Adresse generiert. Zwar hat weiterhin jede Bitcoin-Adresse ihren individuellen privaten und öffentlichen Schlüssel, diese muss man jedoch nicht mehr sichern: Es genügt, den Seed zu kennen, denn daraus lassen sich durch Hochzählen nacheinander alle Schlüssel und somit alle zu diesem Wallet gehörigen Bitcoin-Adressen erzeugen.

Dauerhafte Backups

Für Bitcoin-Nutzer war die Einführung der deterministischen Wallets eine große Hilfe. Endlich konnte man ein Backup seines Wallets anlegen, das auch Monate und Jahre später noch funktionierte. Es genügte nun sogar, einmalig unmittelbar nach Erzeugen eines deterministischen Wallets den Seed als eine Reihe 12 oder 24 englischer Wörter (gemäß BIP-39) aufzuschreiben und sicher zu verwahren, um jederzeit wieder an seine Bitcoins heranzukommen.

Doch genau hier liegt auch die Gefahr: Bekommt ein Angreifer den Seed auch nur ein einziges Mal zu sehen, kann er auch Jahre später noch über sämtliche Bitcoins des Wallets verfügen. Es gibt keine Möglichkeit, ihn seinem Zugriff zu entziehen, das Wallet ist für immer verbrannt. Die einzig sichere Lösung ist, die Bitcoins auf eine Adresse eines neuen Wallets mit einem anderen Seed zu transferieren.

Hardware-Wallets haben den Ruf, den Seed sicher zu verwahren und sie niemandem preiszugeben. Das führt zu einer gewissen Sorglosigkeit im Umgang: Weil das Wallet ja sicher ist, ist es nicht mehr so wichtig, es im gleichen Maß zu schützen wie einen auf Papier notierten Seed. Etliche Teilnehmer einer Bitcoin-Konferenz zum Beispiel ließen ihre Hardware-Wallets in ihren Hotelzimmern, berichteten die Hacker auf dem 35C3, anstatt sie mitzunehmen und darauf genauso gut aufzupassen wie auf ihren Geldbeutel.

Keine Panik

Damit ist aber auch klar, dass Privatanwender von den Hacks praktisch nicht betroffen sind. Der Aufwand dafür ist sehr hoch und setzt einen physischen Zugriff auf das Gerät voraus. Wer sein Hardware-Wallet nicht sorglos herumliegen lässt, sondern es wie andere Wertsachen einschließt, kann darauf auch sein Kryptovermögen sicher aufbewahren.

Problematisch sind die Hacks für Kryptobörsen und andere Firmen, deren Mitarbeiter mit Kryptowährungen arbeiten und auch von zu Hause aus oder auf Dienstreisen Zugriff auf ein Wallet benötigen. Sie können nicht kontrollieren, wie gut jeder einzelne Mitarbeiter auf das Hardware-Wallet achtet. So besteht die Gefahr, dass ein Angreifer die Firmware des Hardware-Wallets unbemerkt manipuliert und so schlimmstenfalls Transaktionen an sein Wallet umleitet und diese vollautomatisch autorisiert.

Die Hersteller von Crypto-Wallets sollten den Vortrag vom 35C3 als Lehre für den Entwurf der nächsten Gerätegeneration sehen, denn viele der Angriffsmöglichkeiten sind erst durch Fehler oder Schwachstellen im Design entstanden. So ist der ST31 für Pay-TV-Decoder vielleicht eine gute Wahl, für ein Hardware-Wallet ist er jedoch nicht geeignet, da er immer vom vorgelagerten Mikrocontroller abhängig ist. Er signiert schlicht alles, um das ihn der Mikrocontroller bittet.

Debug- und JTAG-Schnittstellen haben in sicherheitsrelevanten Geräten wie dem Ledger Nano S oder Trezor One generell nichts zu suchen, die entsprechenden Anschlüsse der Prozessoren sind gar nicht erst herauszuführen oder, wann immer möglich, definiert auf Masse zu ziehen, sodass ein Angreifer selbst durch Aufstecken eines Chip-Adapters nicht an sie herankommt. Er müsste dann schon den SMD-Chip aus- und wieder einlöten, um die Firmware manipulieren zu können.

Außerdem scheinen die Ingenieure der Wallet-Hersteller zwar gut mit Digitaltechnik umgehen zu können, ihnen fehlt aber scheinbar der Blickwinkel eines Hochfrequenztechnikers. Anders ist die extrem lange und exponierte Datenleitung zwischen Prozessor und Secure Element im Ledger Blue nicht zu erklären, über die die Hacker die PIN ausspähen konnten. Eine solche Leitung muss so kurz wie möglich sein, um Abstrahlungen, aber auch Einstrahlungen zu verhindern.

Anders als die prinzipbedingten Schwachstellen des verwendeten Secure Element lässt sich das Schaltungsdesign leicht und ohne zusätzliche Produktionskosten verbessern. Dies sollten die Hersteller schleunigst umsetzen. *(mid@ct.de)*

Vortrag „wallet.fail“: ct.de/yqnp