Dirk Fox

Zero Day Exploits

Exploit

Unter einem „Exploit“ wird ein funktionsfähiger Programmcode in einer beliebigen Programmiersprache verstanden, mit dem eine gefundene Schwachstelle eines verbreiteten Programms oder Betriebssystems ausgenutzt werden kann.

Ein Exploit demonstriert als „Proof of Concept“, dass und wie ein theoretisch beschriebener Angriff praktisch funktioniert. In Hackerkreisen gilt ein Exploit zugleich als Beleg dafür, dass die Schwachstelle tatsächlich gefunden und überprüft wurde – und stellt gewissermaßen den „Skalp“ am Gürtel eines Hackers dar. Je größer Anzahl, Vielfalt und Komplexität der entwickelten Exploits, desto größer die Anerkennung in internen Kreisen.

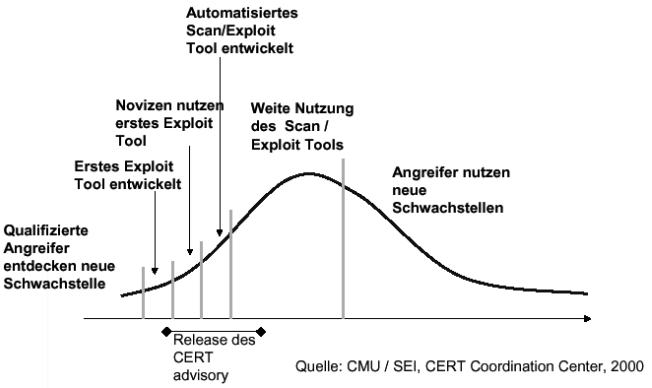
Da Angreifersoftware heute in zunehmendem Umfang in Gestalt von Open Source Projekten weiterentwickelt wird, stellt die Veröffentlichung von Exploits eine wachsende Gefahr dar. Denn oft werden Exploits innerhalb von wenigen Minuten bis Stunden in verbreitete Hacking-Tools integriert und stehen dann vom wenig kompetenten „Amateur-Hacker“ bis zum professionellen Wirtschaftsspion jedermann für automatisierte Attacken zur Verfügung (siehe Abbildung).

Patch

Der Hersteller einer Software muss hingegen – nachdem er von der Entdeckung einer kritischen Sicherheitslücke erfahren hat – zunächst den Fehler eingrenzen, korrigieren und anschließend einen Software-„Patch“ (Flicken) entwickeln, der die Lücke schließt, sowie das „geflickte“ Programm abschließend testen. Der Patch wiederum muss an alle betroffenen Kunden verteilt, von diesen ihrerseits getestet und dann unternehmensweit installiert („ausgerollt“) werden.

Dieser gesamte Prozess kostet Zeit – erfahrungsgemäß oft mehrere Tage bis Wochen. Während dieses Zeitraums („Window of Exposure“) sind die betroffenen Systeme möglichen Angriffen ausgesetzt und können höchstens durch zusätzliche und speziell konfigurierte Sicherheitskomponenten (neben Firewalls und Virenscannern) wirksam vor Angriffen geschützt werden. Sind Anwendungen betroffen, auf die ein Angreifer direkten Zugriff hat, wie beispielsweise öffentliche Web-Applikationen, kann es erforderlich sein, die Anwendung bis zur Verfügbarkeit des Patches zu sperren.

Abbildung |



Quelle: CMU / SEI, CERT Coordination Center, 2000

„0day“

In der Hacker-Szene gab und gibt es immer wieder Diskussionen um die Vertretbarkeit der unmittelbaren Veröffentlichung von Exploits. Zahlreiche Hacker senden, wenn sie auf eine besonders kritische Sicherheitslücke stoßen, ihre Erkenntnisse zunächst an die betroffenen Hersteller, bevor sie – nach einer „Anstandsfrist“ von einigen Tagen, manchmal auch erst nach Bereitstellung eines Patches durch den Hersteller – Details ihrer Erkenntnisse und ein Exploit veröffentlichen.

Allerdings wird diese Vorgehensweise nicht allgemein geteilt. Einige Hacker ziehen die Veröffentlichung von „Zero Day Exploits“ vor, also die Publikation entdeckter Sicherheitslücken ohne zeitlichen Vorlauf für die Hersteller. Aus Verärgerung über die oft unkooperative Haltung betroffener Unternehmen und um den Druck auf solche Hersteller zu erhöhen, die wiederholt kritisch fehlerhafte Software verbreiten, wurden auch schon „0day“-Exploits „mit Ansage“ publiziert – zum Beispiel beim „Month of Browser Bug“, dem „Month of PHP-Bug“ oder dem „Month of Apple Bug“.

Fazit

Zwar ist die Publikation von Zero Day Exploits für Hersteller und Nutzer betroffener Software gleichermaßen unangenehm und kann auch ein kritisches Sicherheitsrisiko darstellen. Dennoch sind selbst Zero Day Exploits deutlich ungefährlicher als entdeckte, aber unveröffentlichte Sicherheitslücken, die von Angreifern unbemerkt ausgenutzt werden können.

Das von solchen Exploits eine wachsende Gefahr ausgeht, zeigen sowohl die erhebliche Zunahme neuartiger Schadprogramme, die unveröffentlichte Schwachstellen nutzen, als auch die wachsende Popularität von Versteigerungsplattformen, auf denen Exploits exklusiv erworben werden können.

Langfristig hilft gegen (Zero Day) Exploits nur die Entwicklung „sicherer Software“ [1], also von Programmen, die bei Auslieferung keine Sicherheitslücken mehr enthalten. Meist jedoch verfügen Softwareentwickler weder über die dafür erforderliche Qualifikation noch über ausreichende Disziplin. Denn trotz aller unleugbaren Fortschritte in der Softwareentwicklung steigt die Zahl entdeckter kritischer Sicherheitslücken – auch aufgrund der wachsenden Komplexität heutiger Programme – nach wie vor Jahr für Jahr [2].

Referenzen

[1] Datenschutz und Datensicherheit (DuD): *Schwerpunkt „Sichere Softwareentwicklung“*, Heft 12/2007.

[2] CERT Coordination Center: *CERT Statistics (Historical)*, Carnegie Mellon University, 2009 http://www.cert.org/stats/