

Programmierfehler/ Bugs

Zusammenfassung

Derzeit laufen Tausende Softwareprogramme auf Millionen von Rechnern in aller Welt. Der Code von Softwareprogrammen oder Betriebssystemen kann so genannte Bugs enthalten. Wenn die Software-Hersteller dieses Problem auf die leichte

Schulter nehmen, ist damit zu rechnen, dass diese Sicherheitslücken weiterhin in zunehmendem Maße ausgenutzt und die Schäden von Tag zu Tag größer werden. Dieses Dokument erläutert, was Bugs sind und auf welche Weise sie die EDV-Systeme gefährden.

Inhalt

- 1 Was ist das? →
- 2 Folgt die Programmierung Sicherheitsstandards ? →
- 3 Warum sollte man sich schützen ? →
- 4 Vorbeugende Maßnahmen →



1 Was ist das ?

Ein Bug (die ursprüngliche Bedeutung dieses englischen Worts ist: "Insekt") ist eine unbeabsichtigte Schwachstelle von Software oder Hardware, die das reibungslose Funktionieren des EDV-Systems gefährdet. Unter den Begriff "Hardware" fallen auch Mikrocodes und sonstige Codes, die in die Schaltkreise und Chips integriert sind.

Unter "Debugging" versteht man die Suche, Korrektur und Beseitigung von

1.1 Allgemeine Informationen

1.1.1. HERKUNFT DES BEGRIFFS

Der erste Computer, der in den USA Ende des Zweiten Weltkriegs zum Einsatz kam, sah noch ganz anders aus als unsere heutigen Rechner. Damals waren die Rechner wesentlich größer und entwickelten viel Wärme. Sie waren daher ein Lieblingsaufenthalt von Insekten (englisch "bug"). Es heißt, dass eine der ersten Pannen eines solchen Rechners von einem Insekt verursacht wurde, das über die Schaltkreise im Innern des ersten Computers krabbelte.

In Frankreich verwendet man dafür den Ausdruck bogue (männlich), der die Schale der Kastanie bezeichnet, die wegen ihrer Stacheln nicht gut rollen kann. Der Ausdruck bogue steht für eine Sache, die nicht reibungslos läuft, nicht wie vorgesehen funktioniert und bei der man zum Kern vordringen muss, um die Ursache festzustellen.

1.1.2. STATISTIKEN

- ➔ Im Jahr 1988 wurden laut dem US-amerikanischen CERT/CC (Computer Emergency and Response Team/Coordination Center) lediglich 8 Bugs verzeichnet.
- ➔ Im Jahr 1998, 10 Jahre später, wurden 3.784 gezählt.
- ➔ Im Jahr 2002 waren bei der gleichen Organisation über 82.000 aufgelistet.

Dieses Wachstum ist natürlich im Kontext der zunehmenden Anzahl der Rechner und Softwareprogramme, aber auch mit der Öffnung der EDV-Welt über einen freien Zugang per Internet zu betrachten.

1.1.3. DER JAHR-2000-BUG

Der Bug, der das größte Medienecho auslöste, war zweifellos der so genannte "Jahr-2000-Fehler". Die meisten Codes, die in den Achtzigerjahren entwickelt oder integriert wurden, speicherten nämlich die Jahreszahl nur unter den beiden Endziffern ab (das Jahr mit dem Code 80 bedeutete also für den Computer 1980). Somit bestand beim Übergang zum Jahr 2000 die Gefahr, dass das EDV-System dies als das Jahr 1900 interpretieren und einen Sprung um 99 Jahre in die Vergangenheit simulieren könnte, da das Jahr 2000 im Computer unter dem Code 00 gespeichert ist, der ebenso das Jahr 1900 bedeuten kann.

Programmierfehler/ Bugs

Dank der Vorsorge der Hersteller in Verbindung mit der guten Verwaltung der meisten EDV-Systeme hielten sich die Auswirkungen in Grenzen, aber es besteht kein Zweifel, dass ohne diese Vorsorgemaßnahmen die Funktionstüchtigkeit der EDV-Systeme beeinträchtigt worden wäre.

1.2 Sehr häufige Programmierfehler

Viele Bugs sind die Ursache der meisten Hacker-Attacken. Wer sich diese Schwachstellen zunutze macht, kann erhebliche Schäden in einem EDV-System anrichten. Bedauerlicherweise muss man feststellen, dass es sich bei diesen Schwachstellen ausschließlich um Fehler bei der Konzeption oder Programmierung handelt.

Die bekanntesten und am häufigsten ausgenutzten Sicherheitslücken sind folgende:

1.2.1. «BUFFER OVERFLOW» ODER «HEAP OVERFLOW»

Dieser Bug wird zweifellos am häufigsten ausgenutzt. Unter gewissen Umständen ermöglicht er von den Programmierern nicht vorgesehene Vorgehensweisen, mit denen sich Hacker Sonderrechte und Privilegien auf dem anfälligen System verschaffen können. Eigentlich handelt es sich dabei um eine Umgehung des dazugehörigen Zwischenspeichers, die von der Software bearbeitet und ausgeführt wird.

BEISPIEL : Ein Benutzer soll 10 Zeichen in ein Eingabefeld auf einer Web-Seite eingeben. Eine Person mit unlauteren Absichten gibt die 10 Zeichen ein, hängt ihnen aber eine weitere Zeichenfolge an. Wenn die Software schlecht konzipiert ist, besteht die Gefahr, dass diese Zeichenfolge unter Umständen ausgeführt wird.

Nach Angaben des CERT infizierte der "Slammer Worm" am 25. Januar 2003 90% der möglichen Ziele innerhalb von 10 Minuten. Dieser Angriff hatte eine Schwachstelle des Typs "Heap Overflow" in Verbindung mit den Diensten von Microsoft Windows SQL2000 Server Resolution Services (SQL = Structured Query Language) ausgenutzt. Da der Bug hinter vielen anderen Softwareprogrammen "versteckt" war, waren sich viele EDV-Verantwortliche gar nicht bewusst, welcher Gefahr sie ausgesetzt waren.

1.2.2. «FORMAT STRING»

Diese Sicherheitslücke war jahrelang in den meisten Rechnern und Softwareprogrammen zu finden. Allerdings wurde man erst Mitte 2000 auf sie aufmerksam.

Sie ist das Ergebnis der Anwendung ungeeigneter Funktionen für die Bearbeitung von Zeichenfolgen. Diese Schwachstelle trat bei der Einführung der Softwareprogramme Microsoft Windows NT4 und 2000 zutage. Denn diese beiden Softwareprogramme verarbeiteten die Daten nicht korrekt und ermöglichten sogar die Einrichtung von "Geheimtüren" (Backdoors).

1.3 Programmierfehler im Internet (web bugs)

Der Begriff Bug wird üblicherweise auch für gewisse Mechanismen unterschiedlicher Art in Verbindung mit Webseiten verwendet. Dazu ist jedoch anzumerken, dass der Begriff Bug hier nicht im Sinne seiner ursprünglichen Definition verwendet wird. Tatsächlich werden die unter diese Kategorie fallenden Mechanismen absichtlich eingerichtet und sind keineswegs zufällige Fehler, sondern erfüllen eine zu einem ganz bestimmten Zweck entwickelte Funktion. Diese Begriffe werden nachfolgend erläutert.

1.3.1. DEFINITION VON "WEB"-BUGS

Es handelt sich dabei um verborgene Mechanismen auf Webseiten, mit denen Informationen über den Benutzer (IP-Adresse, Version des Browsers usw.) und die von ihm zuvor besuchten Web-Seiten beschafft werden sollen. Diese Informationen werden dann an Dritte weitergegeben, entweder zu reinen Spionagezwecken oder zur Ermittlung von Zielgruppen für kommerzielle Zwecke.

⚠ Eine weite verbreitete Vorgehensweise

Laut einer von "Intelytics" vorgelegten Studie enthielten 16 Millionen von 51 Millionen untersuchten Web-Seiten, d. h. 31,4%, einen Spionagemechanismus.

Diese Mechanismen sind in der Regel hinter harmlos aussehenden Bildern, kleinen EXE-Dateien, Scripts, "Cookies" und Anwendungen verborgen. Man findet solche Bugs auf Web-Seiten, in E-Mails, Newsgroups usw. ...

2

Folgt die Programmierung Sicherheitsstandards ?

2.1 Programmiersprachen und Fehler

Die meisten heute verwendeten Programmier-Sprachen sind inzwischen äußerst leistungsfähig und ermöglichen die Einrichtung komplexer interaktiver Anwendungen auf einem öffentlichen Netzwerk.

Diese Programmier-Sprachen überprüfen die Syntax der Sprache, eine Kontrolle auf Fehler bezüglich der Behandlung von "Buffer Overflows" findet bisweilen jedoch nicht statt.

Bei der einfachen Anwendung dieser Sprachen wird die Sicherheit überhaupt nicht berücksichtigt, und es ist unbedingt eine "Abschirmung" dieser Anwendungen vorzusehen. Wenn man Programmiersprachen wie C, C++, Java, Perl oder andere Visual Basic-Sprachen verwendet, muss die Kodierung unbedingt einer Testphase, der so genannten Debugging-Testphase, unterzogen werden, sodass alle Schwachstellen beseitigt werden.

2.2 «open source» und Fehler

Unter Informatikern läuft eine endlose Debatte über die Sicherheit der so genannten "offenen" Softwareprogramme, die Zugang zu ihren Quellcodes ermöglichen, im Vergleich zu "geschlossenen" Programmen, die diesen Zugang nicht gestatten.

Die Befürworter einer geschlossenen Lösung begründen dies damit, dass es schwieriger sei, die Schwächen des Gegners auszunutzen, wenn man diesen nicht kennt. Die Befürworter der offenen Lösung argumentieren dagegen, dass der Programmierer mit jeder Programmzeile seinen guten Ruf aufs Spiel setzt.

Die Statistik zeigt, dass in Wirklichkeit beide Seiten Sicherheitslücken aufweisen, und dass vom Standpunkt der Sicherheit nichts für eine "Open Source"-Lösung spricht.

2.3 Die Verantwortlichen und Programmierfehler

Zurzeit gibt es keine Möglichkeit, den Hersteller einer Software-Lösung für die Ausnutzung von Bugs in seinem Produkt haftbar zu machen, und erst seit Anfang dieses Jahrzehnts sind Strafen für Personen vorgesehen, die diese Sicherheitslücken zu unlauteren Zwecken nutzen.

Eine mögliche Lösung bestünde darin, die Hersteller von Lösungen zur Einhaltung von Sicherheitsstandards zu verpflichten, bevor ihnen die Vermarktung ihrer Lösungen gestattet wird. Die Einhaltung dieser Standards könnte durch unabhängige Labors geprüft werden.

Es wäre jedenfalls interessant, die Haltung bedeutender Marktbeteiligter zu diesem Sicherheitsaspekt zu untersuchen.

2.3.1. MICROSOFT

Viele glauben, dass die Produkte von Microsoft besonders viele Bugs enthalten. Man muss die Zahlen jedoch im Verhältnis zur Anzahl der Implementierungen betrachten. So gesehen ist Microsoft das Opfer seines Erfolges.

Zu Anfang dieses Jahrzehnts erklärte Bill Gates, die Sicherheit der Softwareprogramme sei in den nächsten fünf Jahren oberstes Gebot. Es ist derzeit noch etwas zu früh, um die Ergebnisse beurteilen zu können. Jedenfalls ist festzustellen, dass gewisse Softwareprogramme nicht zum vorgesehenen Zeitpunkt auf den Markt kamen, da sie Sicherheitsprüfungen unterzogen wurden, und dass die meisten Code-Zeilen von Microsoft Windows neu geschrieben wurden, um den neuen Anforderungen zu genügen.

2.3.2. ORACLE

Im Frühjahr 2003 ermittelte das US-amerikanische CERT/CC 37 Sicherheitslücken im Datenbank-System von Oracle. Die meisten dieser Sicherheitslücken fanden sich in der Kodierung des Produkts. Damit wurde das Verkaufsargument der höheren Zuverlässigkeit, mit dem der Hersteller bis dahin geworben hatte, widerlegt.

2.3.3. LINUX

Wie bereits erwähnt, schützt das Arbeiten mit "offener" Software nicht unbedingt vor Bugs. Die "Open Source"-Gemeinschaft gesteht selbst ein, dass die zunehmende Beliebtheit von Linux zur Vermehrung von Bugs führt, weil dessen Softwareprogramme zunehmend auf das Interesse von Hackern stoßen, die noch unbekannte Bugs aufspüren wollen.

Außerdem dürfte dies zur Einrichtung ungeschützter Softwareprogramme führen, weil das gesteigerte Interesse an diesen Softwareprogrammen nicht notwendigerweise mit ihrer technischen Beherrschung seitens der Benutzer einhergeht.

3

Warum sollte man sich schützen ?

Man sollte sich schützen, weil Bugs eine erhebliche Gefahr für alle Benutzer von EDV-Systemen darstellen. Sie können zu erheblichen Verlusten führen:

Direkte finanzielle Verluste:

Direkte finanzielle Verluste:

- Zerstörung kritischer Daten
- Vollständige oder teilweise Außerbetriebsetzung oder Nichtverfügbarkeit des EDV-Systems
- ...

Reputations-Verlust

- Infragestellung der Glaubwürdigkeit im Fall einer Verbreitung vertraulicher Informationen
- ...

Zeitverlust

- Aufwand für die Wiederherstellung der zerstörten Daten
- ...

4

Vorbeugende Maßnahmen

Der Kampf gegen solche Sicherheitslücken findet leider nicht auf der Ebene der Benutzer, sondern auf der Ebene der Hersteller von Software oder EDV-Systemen statt. Unter diesen Umständen können die Benutzer lediglich reaktive, nicht aber vorbeugende Schutzmaßnahmen ergreifen.

4.1 Die Aktualisierung des Programme und Systeme

Im Internet gibt es mehrere Web-Seiten, z. B. von SANS (SysAdmin, Audit, Network, Security institute), CERT und viele andere, die in regelmäßigen Abständen eine Liste der wichtigsten Sicherheitslücken und Maßnahmen zum "Abdichten" der "Leaks" veröffentlichen. Die Verwalter von EDV-Systemen müssen über die Sicherheitslücken auf dem Laufenden sein und so rasch wie möglich die entsprechenden Patches installieren. Dabei ist es jedoch ratsam, die Patches zunächst auf einer Testplattform zu prüfen, damit es nicht zu Kompatibilitätsproblemen mit der in der Produktion eingesetzten Software kommt.

4.2 Die Validierung von Systemen

Verschiedene EDV-Beraterfirmen bieten so genannte "Attack and Penetration Tests" an, bei denen die gesamte Infrastruktur geprüft wird, indem man sich zum einen in die Rolle eines externen Hackers, zum anderen in die Rolle eines Standard-Benutzers des EDV-Systems versetzt. Die gleichen Firmen bieten im Anschluss an einen derartigen Auftrag einen technologischen Überwachungsdienst zur weiteren Gewährleistung der Sicherheit an.

Bei den "Web"-Bugs liegt das Problem auf einer anderen Ebene. Hier kommt es darauf an, einen Kompromiss zwischen dem eingegangenen Risiko und der Einschränkung der Funktionen zu finden. Die meisten möglichen Sicherheitsmaßnahmen sind nämlich mit dem Verlust von Funktionen im Internet verbunden.