HELPDESK

Buffer Overflows und Co.

Jede Woche beantworten Sicherheitsexperten Leserfragen und geben Ratschläge, wie sich die Sicherheit in einem Unternehmen erhöhen lässt.

ILLUSTRATION: CW/THÜ

Frage: Wie schützt man Programme vor Buffer Overflows und Format-String-Attacken?

Buffer Overflows und Format String Bugs sind Sicherheitslücken, welche aus dem Umgang mit Eingaben resultieren. Beim klassischen Buffer Overflow nimmt eine Funktion eine Zeichenkette als Argument entgegen, ohne deren Länge zu prüfen. Damit der Prozessor nach dem Abarbeiten einer Funktion weiss, an welcher Adresse er fortzufahren hat, schiebt er diese als erstes auf den Stack. Wird in der verwundbaren Funktion die überlange Eingabe ebenfalls auf den Stack kopiert (z.B. durch die Funktion «strcpy» in eine lokale Variable), wird dadurch die zuvor abgelegte Rücksprungadresse überschrieben: Der Prozessor sucht seine Befehle nach Verlassen der verwundbaren Funktion am falschen Ort und das Programm stürzt in der Regel mit einem Segmentierungsfehler ab. Der Kunstgriff beim Buffer Overflow ist, die Rücksprungadresse so zu überschreiben, dass sie auf die Adresse des Stacks zeigt, wo der Angreifer seine Instruktionen (Shellcode) platziert hat. Ein Exploit für Buffer Overflows besteht also aus Füllzeichen (meist 0x90), Rücksprungadresse und den Instruktionen (Payload/Shellcode). Die Reihenfolge kann dabei variieren.

Ein ähnliches Prinzip verfolgen Format String Attacken. Diese waren im Untergrund bereits mehrere Jahre im Einsatz, bevor das Problem in den späten Neunzigern öffentlich bekannt wurde. Der Angreifer manipuliert die Formatierung von Ausgaben. Beispiel: Wird einer Funktion ein Zeiger auf eine Zeichenkette übergeben und mit der Funktion «printf» ausgegeben, kann der Benutzer durch die Eingabe des Format-Codes «%08x» Bereiche des Stacks ausgeben. Durch Wiederholungen dieser Sequenz kann der Angreifer bis zum kontrollierten Bereich des Stacks wandern. Dort platziert er zum Beispiel einen Zeiger auf einen String, welchen er mit dem Format Code «%s» referenziert. Dadurch wird es möglich, ganze Strings des Arbeitsspeichers auszugeben. Mit der Formatierung «%n» können indirekt Werte geschrieben werden. Wie so viele Angriffsmethoden nutzen die beschriebenen Techniken Programmierfehler aus, um die Daten- und Programmwege zu kreuzen.

> «Von sicherheitskritischen Funktionen wie etwa printf sollten Programmierer absehen.»

Schutzmassnahmen: Systemadministratoren sollten ihre Systeme auf einem aktuellen Patchlevel halten. Sobald ein Patch verfügbar ist, kann mittels Reverse Engineering und dem Vergleich des gepatchten Codes exakt auf die Lücke geschlossen werden. Ein modernes IDS/IPS kann solche Angriffe erkennen und Reverse Proxys filtern verdächtige Anfragen.

Der bessere Ansatz ist natürlich, die Lücken zu vermeiden. Hierbei sind die Entwickler gefragt. Sie sollten sowohl beim Design als auch beim Programmieren die Sicherheitsaspekte präsent halten und sich nicht nur auf die Funktionalität beschränken. Dies ist vor allem bei der zeitlichen Planung und bei der Wahl der Programmiersprache zu berücksichtigen.

Methoden und Funktionen, welche einen String annehmen, sollten stets einen weiteren Parameter für dessen Länge verarbeiten. Von der Verwendung sicherheitskritischer Funktionen wie «printf» sollte der Programmierer absehen und nie davon ausgehen, dass ein String nullterminiert ist. Benutzereingaben sollten vor allem bei Fehlermeldungen nur wenn unbedingt nötig rezitiert werden. Die Definition von Konstanten für Puffergrössen hat sich ebenfalls bewährt. Teils frei verfügbare Tools suchen bereits im Source Code nach solchen Sicherheitslücken.

Bei Client/Server-Applikationen reichen Längenbeschränkungen für Eingabefelder nicht aus, da den Angreifer nichts daran hindert, das Frontend zu übergehen und direkt mit der Serverkomponente zu kommunizieren. Die Eingaben sind also serverseitig auf die enthaltenen Zeichen und Längen zu validieren. Erfahrene Programmierer wissen: Der normale Benutzer allein ist schon böse, aber der Angreifer gar mehr! ■

Der Autor
Simon Wepfer ist Consultant bei der Sicherheitsberaterin Oneconsult, Thalwil, www.oneconsult.com

Unsere Experten beantworten Ihre Fragen. Schreiben Sie uns: it-security@computerworld.ch

Ein Archiv der Helpdeskartikel finden Sie im Internet: www.computerworld.ch