Dipl.-Ing. (FH) Dipl.-Ing. Michael Georg Grasser
Institut für Technische Informatik
E-Mail: michael.grasser@tugraz.at
Tel.: 0316 873 6906

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Eugen Brenner
Institut für Technische Informatik
E-Mail: brenner@tugraz.at
Tel.: 0316 873 6402

Secure CPU - Eine sichere Prozessorarchitektur für den Einsatz in mobilen und eingebetteten Systemen

Secure CPU - A Secure Processor Architecture for Mobile and Embedded Systems

Eine der Schlüsselverantwortlichkeiten von heutigen Software- und Hardwarearchitekten ist die Sicherung von Computersystemen und deren Daten. Aktuelle Statistiken zeigen, dass die Ära der Systemverwundbarkeiten noch lange nicht vorüber ist. Wegen der strengen Anforderungen und Beschränkungen in eingebetteten, mobilen und sicherheitskritischen Systemen können nur spezielle Softwareanwendungen dafür verwendet werden. Daher ist es sinnvoll und notwendig, die Anwendungen durch die Verwendung einer sicheren Hardwarearchitektur passiv zu schützen. Das Kernstück jeder Hardware ist ein Prozessor, der für die Verarbeitung der Daten zuständig ist.

Die Grundidee unserer Forschung war es, jedes Prozessorregister mit zwei zusätzlichen Registern zu erweitern, welche den jeweils niedrigst- und höchstzulässigen Wert enthalten. Aufgrund dieser Daten können sowohl Wert- als auch Referenztypen erfolgreich auf die Einhaltung der Grenzen überprüft werden. Diese Methode der Überprüfung wird im Englischen Bound Checking genannt. Teile des Gesamtkonzepts wurde von Ideen und Mechanismen der Common Language Infrastructure (CLI) abgeleitet. Die CLI, die heute ein ISO/IEC/ECMA Standard ist, bietet die Möglichkeit, von vielen Programmiersprachen auf die Common Intermediate Language (CIL) zu kompilieren. Das bedeutet, dass es hier eine einheitliche, plattformunabhängige Zwischensprache gibt, von der aus der Just-In-Time-Compiler oder ein Interpreter auf die Hardwaresprache übersetzt. Das CIL-Assembly beinhaltet neben den Programmdaten auch Metadaten der inkludierten Datentypen, welche für das weitere Bound Checking herangezogen werden.

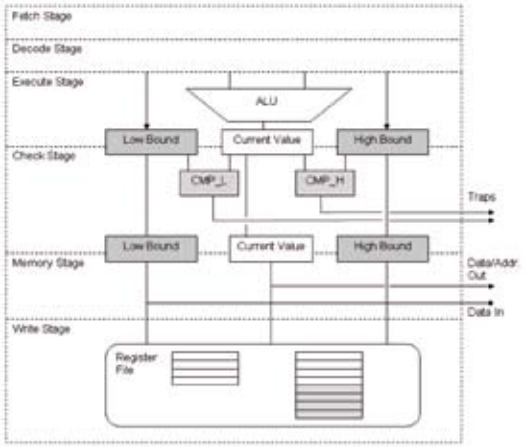

Abb.1: Ausschnitt des veränderten LEON 2 Prozessorkerns

Um unser Konzept zu verifizieren, wurde ein bestehender und kostenfreier Prozessorsimulator namens CPUSim modifiziert, welcher die neue Hardwarearchitektur inklusive des sicheren Prozessorkerns (SecureCPU) beinhaltet. Die Resultate zeigten, dass die Implementierung dieser SecureCPU einen merkbaren Sicherheits- und Leistungsvorteil ergab.

Die aktuell laufende Forschung beinhaltet die Hardware-Realisierung des genannten Konzeptes. Der von Gaisler Research entwickelte, auf der SPARC V8 basierende LEON 2 Prozessorkern wurde für ein Entwicklungsbord GR-XC3S1500 der Firma Pender adaptiert und synthetisiert. Nach erfolgter Inbetriebnahme wurde dieser Prozessorkern modifiziert, um implizites Bound Checking zu ermöglichen. Hierzu wurde die 5 stufige Pipeline des LEON 2 Prozessor um eine Check-Stufe erweitert, welche Hardware-Komparatoren für die Vergleiche des aktuellen Wertes mit den assoziierten Grenzen beinhaltet. Diese neue Stufe wurde mit dem integrierten Trap-Handling Mechanismus verbunden und kann im Fehlerfall Interrupts auslösen. Für die interne Speicherung der Grenzen mussten einerseits das Registerfile und andererseits die Ladebefehle des Instruktionssatzes erweitert werden. Zusätzlich wurde eine Markierung der für die Grenzen zugewiesenen Speicherstellen im Random Access Memory vorgesehen. Diese Idee wurde mit Hilfe des SecureTags realisiert, welches eine Erweiterung zum NX- und XD-Bit der Prozessorhersteller Intel und AMD darstellt. Aufgrund der Architektur und der verwendeten Tool-Chain ist die Implementierung für die Verwendung mit der CLI nicht notwendig, da eine implizite Speicherstellenüberprüfung bei jedem Zugriff stattfindet. Dies ist wohl aber bei anderen Zwischensprachen der Fall, da in der Regel bedeutend weniger Checks auf dieser Ebene durchgeführt werden.

Die Vorteile unseres Konzeptes liegen in der nur geringfügigen Änderung der Architektur, in einer sicheren Ablage von Grenzen im Registerfile und in einer hohen Kompatibilität zu neuen, aber auch bestehenden Softwareanwendungen.

Nationale und internationale Förderungsprogramme, wie das Rahmenprogramm IST-FP6 der Europäischen Union, sind an der Forschung im Bereich der Sicherheit eingebetteter und vernetzter Systeme interessiert, dieses Projekt SecureCPU vom Bundesministerium für Technologie und Innovation gefördert. Diese Arbeit wurde bereits in einigen Printmedien veröffentlicht und bei internationalen Konferenzen präsentiert. Zunehmend interessieren sich auch Vertreter der Industrie, welche den Mangel an Sicherheit in eingebetteten Systemen erkannt haben.

Weiterführende Links
http://www.sparc.org (SPARC International)
http://www.gaisler.com (Gaisler Research)
http://www.ecma-international.org (ECMA international)
http://msdn2.microsoft.com/netframework/default.aspx (Microsoft .NET)
http://www.iti.tugraz.at (Institut für Technische Informatik)

Secure CPU - A Secure Processor Architecture for Mobile and Embedded Systems

Our research topic is getting more important, as shown in the National Vulnerability Database from NIST, where the percentage of software defects, due to buffer overflows, currently holds at 19% and is constantly increasing.
Due to strict restrictions in mobile and embedded security-critical systems, only special software applications are used. Consequently, it is necessary to secure those applications passively by the use of a secure processor architecture.
Our basic idea was to extend each processor register by two additional registers, which represent the lowest and highest value the particular register is allowed to store. As a result, a value or a pointer can be checked whether it is within the given bounds. Furthermore, a technique, called SecureTag, was proposed to mark memory lines and to separate between code, data, as well as low and high bound. To proof our concept, we adapted a processor simulator to implement the proposed security features. The advantages of our concept are minor changes in the architecture, the secure storage of bounds in memory and the high compatibility to new and legacy software applications. When compared to existing solutions, our implementation of bound checking results in a noticeable increase in security of mobile and embedded systems.