MICRO CONSULT

05/2012

Die drei Gesichter der Sicherheit von Software-Systemen

Teil 2 – Die Zugangs- oder Angriffssicherheit

In Teil 1 dieser Serie über den letzten Embedded Software Engineering Kongress (ESE Kongress) vom Dezember 2011 in Sindelfingen ging es um das Thema Safety, die Betriebssicherheit. Teil 2 dieser Serie fasst einige Expertenmeinungen und -empfehlungen zum Thema Security zusammen. Dabei zeigt sich, wie sehr die Betriebssicherheit von softwareintensiven embedded Systemen mit dem Schutz vor unbefugtem Zugriff oder gezieltem Angriff verbunden ist. Die dazu notwendigen Maßnahmen stellen Software-und Hardwareentwickler gleichermaßen vor hohe Herausforderungen.

Embedded Systeme sind stets Software- und Hardwarekomponenten, die in Verbindung mit Sensoren und Aktuatoren in umgebende technische Systeme integriert sind, um komplexe Steuerungs- und Datenverarbeitungsaufgaben zu übernehmen. Die Sicherheit im allgemeinen Sinne bezeichnet einen Zustand, der frei von unvertretbaren Risiken ist. Dies wird in der Norm IEC 61508 so definiert. Safety oder funktionale Sicherheit bedeutet in diesem Zusammenhang, dass ein System in den zu erwartenden Situationen und Zuständen bei ordnungsgemäßer Nutzung keine körperlichen oder finanziellen Schäden verursacht. Security könnte man demnach so genauer definieren: Die Betriebs- und Datensicherheit darf durch äußere Einflüsse nicht gestört oder gar verhindert werden. Mithin hat die Security eine doppelte Bedeutung: Einerseits sorgt sie dafür, dass die Betriebssicherheit erhalten bleibt, andererseits versteht man darunter die Gesamtheit der Maßnahmen, um ein System nach außen zu schützen. Das unterstreicht sowohl die Abgrenzung dieser Begriffe, zeigt aber auch, wie eng sie zusammenhängen.

Würmer, Viren & Co. – Schutz vor gezielten Angriffen

In seinem Vortrag "Sichere Netze der Zukunft" auf dem ESE Kongress 2011 definierte Prof. Dr.-Ing. Hans-Joachim Hof von der Hochschule München den Begriff pragmatisch: Die Security ist der Schutz vor gezieltem und höchstwahrscheinlich auch böswilligem Handeln mit dem Ziel, Vertraulichkeit, Integrität, Authentizität etc. zu erreichen.

Das kann sich in der Realität als nicht ganz einfach erweisen, da man zum Zeitpunkt der Projektierung in der Regel nicht weiß, welcher Art von Angriff ein System irgendwann ausgesetzt sein wird. Besonders bei älteren Systemen kann sich das als geradezu dramatische Gefahr erweisen. Prof. Dr. Hof spricht hier ganz plastisch von der Götterdämmerung der IT-Sicherheit. Ein Bild, das wirklich passt, denn das Vertrauen auf sichere IT-Komponenten, embedded Systeme und Ähnliches grenzt oft wirklich an ein sagenhaftes Gottvertrauen. Frappierendes Beispiel hierfür ist der Computerwurm Stuxnet, der 2010 für große Unruhe und Schäden verantwortlich war. Stuxnet war in einer Weise aktiv, die bis dahin im Grunde als nicht möglich galt: Der Wurm konnte Grenzen physikalisch getrennter Netze überspringen.

Stuxnet zielte vermutlich erfolgreich auf eine Urananreicherungsanlage Wegen dem hohen Schutzbedarf von Atomanlagen schien so ein Angriff bis dahin schlicht nicht möglich zu. Den aktuellen Möglichkeiten des technisch machbaren drückte Stuxnet damit seinen eigenen Stempel auf. Eine optimierte Betriebssicherheit, wie im Beispiel der Atomanlage, wurde mangels ausreichender Security zum Sicherheitsrisiko.

Der menschliche Faktor: Wenn Hacker aktiv werden

Bedrohungen durch Hacker stellen die Entwickler von embedded Systemen vor zunehmende Herausforderungen. Die Referenten des Schweizer Kommunikations- und Technikdienstleisters Albis Technologies AG stellten im Vortrag "Embedded Security: Hackern einen Schritt voraus" zunächst die zunehmende Vernetzung von embedded Systemen heraus. Und obwohl dadurch mechanische Barrieren zwischen den Systemen immer mehr wegfallen, beruht die Sicherheit paradoxerweise häufig auf physikalischer Sicherheit und weniger auf Cyber-Security oder Informationssicherheit. Und das, obwohl es gerade in Infrastrukturnetzen, wie zum Beispiel der aktuell diskutierten intelligenten Stromversorgung, von großer Bedeutung ist, dass die vernetzten Komponenten autonom, ohne Benutzereingaben und vor unbefugten Zugriffen geschützt miteinander kommunizieren können. Was könnte die Lösung sein? Die Referenten stellten ein Modell Hardware-basierter IT-Sicherheitsarchitektur für embedded Systeme in den Raum. In deren Zentrum steht ein Security Controller oder ein Trusted-Platform-Modul, also ein sicherer Speicher für digitale Schlüssel. Dies dient dazu, eine Art Zentrum für kryptografische Operationen zu bilden. Dieses muss an den lokalen Computer gebunden sein und nicht an einen bestimmten Benutzer. Es ist dann unmöglich, das Trusted-Platform-Modul entgegen den Interessen des Eigentümers zu nutzen, sofern dieser Beschränkungen festgelegt hat: eine Basis für die sichere Kommunikation der Netzteilnehmer. Mithin kann es gelingen, beispielsweise aus einem embedded Gerät mit angepasster Software und einem entsprechenden Betriebssystem eine "vertrauenswürdige Plattform" werden zu lassen.

Intelligente Stromnetze, Smart Grids, erfordern für die Ermittlung des individuellen Strombedarfs und die korrekte Abrechnung die Verwendung ebenso intelligenter Verbrauchszähler, der sogenannten Smart Meter. Dies bietet zahlreiche Szenarien, die böswilliges Eindringen mit Betrugsabsicht ermöglichen. Erschwerend ist die Liberalisierung des Strommarktes: Zahlreiche Klein- und Kleinststromlieferanten - aus Wasserkraft oder Photovoltaikanlagen - speisen in die Netze ein. Letztlich alles Einfallstore für mögliche Angriffe zum Zweck der kriminell motivierten Manipulation.

Auch vor diesem Hintergrund erarbeitet das Bundesamt für Sicherheit in der Informationstechnik (BSI) gemeinsam mit Industriepartnern für Smart Meter ein neuartiges Schutzprofil. Es strukturiert Bedrohungen und legt damit die Mindestanforderungen fest, die Sicherheitsmaßnahmen erfüllen müssen. Ziel ist dabei eine Zertifizierung von Produkten nach definierten Schutzkriterien. Das von den Partnern erarbeitete Schutzprofil /3/ definiert ein anhand gemeinsamer Kriterien festgelegtes "Security Modul" für alle kryptografischen Operationen.

Kryptologie – Software sollte nicht zu knacken sein

Bei der Entwicklung von embedded Systemen können durch die Anwendung kryptologischer Verfahren viele Prozesse sicher oder zumindest sicherer gemacht werden. Reinhard Wobst (UNIX Software) hat in seiner Funktion als Softwareentwickler festgestellt, dass mit großer Regelmäßigkeit immer wieder die gleichen Fehler bei der Implementierung von Kryptologie gemacht werden. Während seines Vortrags auf dem ESE Kongress erklärte er, warum und wie es Eindringlingen oft so unnötig leicht gemacht wird. So gehen Softwareentwickler häufig davon aus, dass niemand die von ihnen entwickelten Algorithmen knacken kann. Das Gegenteil ist der Fall. Ein gängiges Mittel ist das Schreiben unklaren oder scheinbar verworrenen Codes. Doch gerade bei der heutigen Vernetzung wird jedes Verfahren früher oder später bekannt. Interessant ist die auch von Wobst formulierte Erkenntnis, dass der Selbsttest eines eigenentwickelten Algorithmus möglicherweise zu dem Ergebnis führt, dass es nicht einmal dem ursprünglichen Programmierer gelingt, ihn zu knacken. Das gilt dann als besonders sicher.

Aber die Wirklichkeit sieht leider anders aus: "Jeder kann einen Algorithmus entwickeln, den er selbst nicht brechen kann, aber es geht darum, dass ihn andere nicht brechen können[1]". Letztlich sieht man es einem Algorithmus, auch wenn er noch so obskur erscheint, nicht an, ob er nun sicher ist oder nicht.

[1] B. Schneier, Memo to the Amateur Cipher Designer

Wobst rät seinen Entwicklerkollegen, konservativ zu sein und bekannte und altbewährte Algorithmen und Methoden zu nutzen. Das Forschen nach Sicherheit solle den Forschern überlassen werden. Anhand zahlreicher Beispiele belegte Reinhard Wobst, wie schnell es geschehen kann, dass ein Softwareentwickler Gutes wollend das Falsche tut bzw. das Richtige nicht richtig macht. Und er zeigte auch die Grenzen auf, die selbst einer perfekt implementierten Kryptologie gesetzt sind, wie z.B. denial of service Attacken oder der immer erfolgversprechende Versuch, das stets schwächste Element eines Walls von Maßnahmen für die Sicherheit zu instrumentalisieren: den User selbst. So kam Wobst bei seinem Vortrag zu der Schlussformel, dass gesunder Menschenverstand niemals hinter Formalismen verschwinden darf. Für die Realisierung von Sicherheit gilt immer, dass etwas Sicherheit besser ist als gar keine.

Qualität und höchstmögliches Sicherheitsniveau für Software

Wie real die Risiken von Angriffen auf embedded Systeme sind, zeigte Lucas von Stockhausen von Hewlett-Packard in seinem Vortrag "Mehr Sicherheit für Embedded Code" am Beispiel einer Manipulation einer handelsüblichen Insulinpumpe. Es sei möglich, durch Umgehen von Restriktionen die Insulindosis, die eine solche Pumpe automatisch verabreicht, aus 100 Metern Entfernung und völlig unbemerkt zu verändern. Damit könne man einem Diabetiker jederzeit eine Überdosis Insulin injizieren. Diese Gefahr mag konstruiert erscheinen, für den Pumpenhersteller (und für den Patienten) ist sie aber sehr real und höchst bedrohlich.

Das Bundesministerium für Wirtschaft und Technologie stellte bereits 2010 eine zunehmende und alarmierende Professionalisierung von Angriffen fest. Dabei berichten die Medien meist nur über die spektakulärsten Fälle, wie zum Beispiel den bereits erwähnten Stuxnet-Wurm, der für den Einsatz auf industrielle Ziele optimiert wurde. Die Mehrzahl der Angriffe ist zwar weniger spektakulär, doch dabei kaum weniger gefahrenvoll. Gerade überraschende Angriffsszenarien auf embedded Systeme zeigen, wie schwierig es sein kann, schon in der Softwareentwicklung vorauszusehen, wie ein späterer Angriff aussehen kann. Aus diesem Grund spielt die sichere Softwareentwicklung, der sichere Code, eine entscheidende Rolle. Angesichts des rasanten Fortschritts, der von zunehmendem Zeit-, Konkurrenz- und Erfolgsdruck zuverlässig begleitet wird, besteht die zentrale Herausforderung darin, neben Qualitätsforderungen auch das höchstmögliche Sicherheitsniveau zu realisieren.

Gerade bei embedded Software kann eine Verbesserung der Zugangssicherheit über Software-Updates erfolgen, so Prof. Dr.-Ing. Hans-Joachim Hof. Daraus ergeben sich gleichermaßen Aufgaben für die Entwicklung von Soft- und Hardware. In punkto Hardware werden dafür nämlich sichere Interfaces und Übertragungsprozeduren benötigt. Die Software muss so aufgebaut sein, dass sie autorisiert korrigiert, aber nicht ohne Autorisierung manipuliert werden kann. Das Motto "Verbauen und Vergessen" gehört damit der Vergangenheit an.

Die psychologischen Sicherheitslücken

Nicht zu vergessen sind die psychologisch bedingten Sicherheitslücken, zu denen Peter Siwon (MicroConsult) in seinem Seminar „Irren ist menschlich“ einige Beispiele nannte, sowohl auf Entwickler- wie auf Anwenderseite.

So bestehe gerade unter hohem Zeitdruck eine verstärkte Neigung, Sicherheitsregeln zu brechen. Menschen nehmen dann höhere Risiken in Kauf, um den Stress loszuwerden. Die Stresshormone spielen dabei eine entscheidende Rolle: Sie schränken unsere Fähigkeit ein, über die langfristigen Folgen unseres Verhaltens nachzudenken. Zudem leide unser Qualitätsbewusstsein unter dem Bedürfnis, den „Schmerz“ loszuwerden. Beides zusammen stellt eine explosive psychologische Mischung dar, die beispielsweise dazu führt, dass wichtige Voraussetzungen für Security, wie beispielsweise umfassende Tests, unter den Tisch fallen. Eine weitere Sicherheitslücke entstehe durch Denkfaulheit. Um unser Gedächtnis zu entlasten, greifen wir nicht nur bei Passwörtern gerne auf altgewohnte oder bequeme Denk- und Verhaltensmuster zurück. Für professionelle Codeknacker ein gefundenes Fressen.

Thomas Batt, beim Training- und Consultingspezialisten MicroConsult zuständig für Software Engineering Management, zieht folgendes Resümee: „(Horror-)Szenarien, bei denen embedded Systeme durch äußere Angriffe manipuliert werden oder bei denen sogar die komplette Kontrolle übernommen wird, können wir uns sehr gut vorstellen. Wenn nicht, dann hilft uns Hollywood dabei. Wir sind heute von einer rasant wachsenden Zahl von embedded Systemen umgeben und verlassen uns darauf, dass Herstellerfirmen sie zugangs- und angriffssicher entwickeln, testen und produzieren. Doch Tatsache ist, dass für den größten Teil des embedded Marktes die Qualitätsmerkmale Zugangs- und Angriffssicherheit noch Neuland sind. Die Berücksichtigung dieser Qualitätsmerkmale muss sich zukünftig wie ein roter Faden durch den gesamten Entwicklungs- und Produktionsprozess betroffener Produkte ziehen. Dazu benötigen wir geeignete Methoden - insbesondere für den Test - sowie unterstützende Tools, Standards und Zertifizierungen. Doch selbst mit noch so hochentwickelten Maßnahmen können wir die Risiken nur minimieren, nicht aber völlig ausschließen.“

Autor:

Peter Siwon ist Business Development Manager beim Münchner Unternehmen MicroConsult, Trainingsspezialist für Embedded Software Engineering.

Alexander Sedlak ist als freier Autor tätig.

MicroConsult GmbH:

Training, Coaching und Consulting für Software- und Hardwareentwicklung sowie Führungskräfte in der Industrie.

www.microconsult.de