softScheck
we identify vulnerabilities others don't

Smart Metering Security

Hartmut Pohl[1]

Einer der wesentlichsten potentiellen Angriffspunkte im Smart Grid ist das Smart Meter Gateway, das Kunden-bezogene Informationen wie Verbräuche angeschlossener Geräte sammelt und Informationen wie Tarifierungsprofile der Energieunternehmen speichert. Damit werden Qualitätssteigerungen und Kosteneinsparungen mit flexiblen Tarifen und aktueller Angebotssteuerung möglich – machen aber auch intensive Sicherheitsmaßnahmen erforderlich. Das Smart Meter Gateway muss daher besonders abgesichert werden.

Über das in jedem Haus oder in jedem Haushalt installierte Smart Meter Gateway sind alle Stromanbieter mit allen Verbrauchern (Endkunden) verbunden – also auch alle Endkunden untereinander. Eingesetzt werden – genauso wie bei Industriesteuerungen (Industrial Control Systems – ICS) – die mehr oder weniger sicheren klassischen (Hard- und Software-) Komponenten der Informationstechnik inklusive der Telekommunikationstechnik.

- Sind denn diese Verbrauchsdaten der Kunden gegen unberechtigte Kenntnisnahme und Auswertung hinreichend geschützt (Datenschutz)?
- Sind denn auch die Tarifdaten gegen Manipulation hinreichend geschützt (Datenschutz und IT-Sicherheit)?
- Sind generell die Steuerdaten in den Smart Meter so gegen Manipulation geschützt, dass z.B. der Strom nicht flächendeckend in ganzen Regionen von Unberechtigten abgeschaltet werden kann?
- Ist denn auch die IT der Energieunternehmen gegen Angriffe aus dem Smart Grid und vor Angriffen von z.B. Endverbrauchern hinreichend geschützt?

Dass Smart Meter geschützt werden müssen, zeigen auch die erfolgreichen Angriffe auf Industriesteuerungen mit den weltweit im Internet übertragenen Würmern Stuxnet, Duqu, Flame, Mahdi, Gauss, Shamoon etc. und deren Nachfolger, Varianten und Derivaten. Zukünftig wird dies noch deutlicher werden, wenn nicht nur weit entfernte Uran-Zentrifugen o.ä. angegriffen werden, sondern unser aller Stromerzeugung und Verteilung. Stuxnet & Co. waren und sind erfolgreich, weil sie eine Reihe von bis dahin unbekannten Sicherheitslücken ausgenutzt haben.

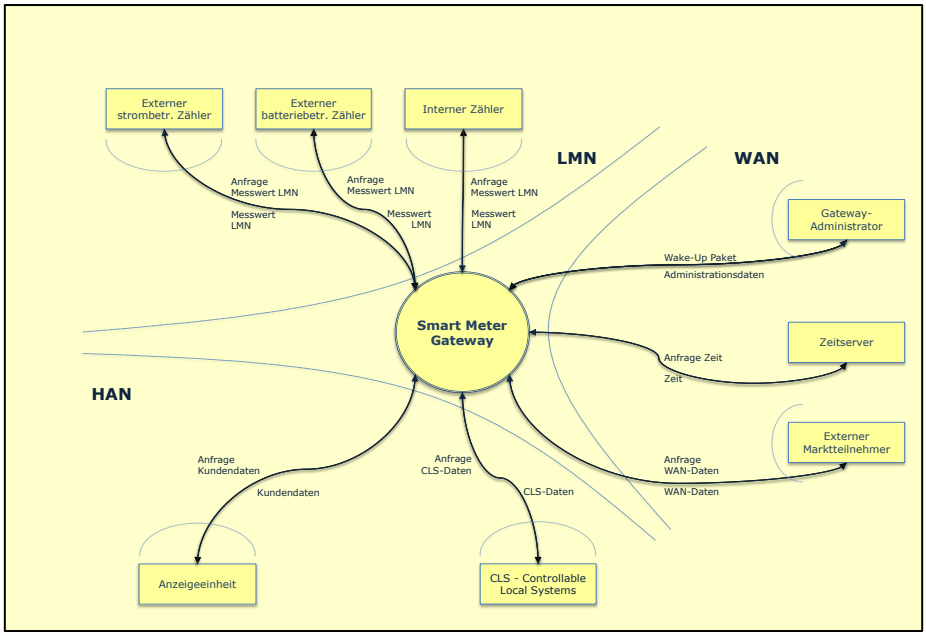

Abb.: Smart Meter Security Threat Model

Konkrete Sicherheitsmaßnahmen für die Software werden von der 'Technischen Richtlinie BSI TR-03109' vorgeschrieben und werden auch mit dem 'Protection Profile for the Gateway of a Smart Metering System' bei der Evaluierung nach den Common Criteria abgeprüft. Das reicht aber nicht aus! Die Risiken sind – wie in der klassischen Informationstechnik auch - nicht allein mit Firewalls, Intrusion Detection und Protection Systems, Anti-Viren-Software und Verschlüsselung abzuwenden.

[1] Prof. Dr. Hartmut Pohl, Geschäftsführender Gesellschafter softScheck GmbH, Köln – Büro: Sankt Augustin

Generell gilt:

Kein (erfolgreicher) Angriff ohne Sicherheitslücke.

Daher müssen alle beteiligten Systeme – insbesondere aber das Smart Metering Gateway auf bisher nicht erkannte Sicherheitslücken überprüft werden: Im ersten Zertifizierungsverfahren für ein Smart Meter Gateway in Deutschland (von Power Plus Communications AG und Open Limit Sign Cubes GmbH) werden daher insbesondere diese 3 Verfahren eingesetzt:

Architectural Analysis - Threat Modeling

Bereits im Design muss Sicherheit berücksichtigt werden: Nach vollständiger Identifizierung schützenswerter Komponenten (Assets) sowie zugehöriger Bedrohungen (potentieller Sicherheitslücken) beginnt die Identifizierung und der Nachweis von Sicherheitslücken mit der Analyse der Dokumentation – insbesondere des Sicherheitsdesigns - sowie eine Untersuchung der Programmablaufpläne und der Datenflussdiagramme (vgl. die Abb.) von und zu allen Kommunikationspartnern wie Stromherstellern und Verteilern bis hin zu den Haushaltsgeräten, Zählern und Anzeigeeinheiten.

Die Erstellung eines korrekten Datenflussdiagramms ist Voraussetzung für ein vollständiges Bedrohungsmodell. Damit lassen sich sicherheitsrelevante Designfehler identifizieren und mögliche Angriffspunkte (Attack Surface) vermeiden.

Static Source Code Analysis

Static Source Code Analysis (Code Review) wird Tool-gestützt durchgeführt. Analysiert wird der Source Code (White-Box-Test) der Zielsoftware ohne ihn auszuführen – bis hin zur semantischen Analyse. Damit ist es möglich, auch komplexe Fehler, die etwa auf Race Conditions, Deadlocks oder falsche Pointerverwaltung basieren, zu identifizieren.

Dynamic Analysis: Fuzzing

Bisher nicht-erkannte Sicherheitslücken werden kostensparend tool-gestützt identifiziert, ohne Kenntnis des Quellcodes. Das erfolgreiche Verfahren Fuzzing wird zur frühzeitigen Identifizierung von Sicherheitslücken und damit auch Kostenreduzierung des gesamten Patchverfahrens erfolgreich eingesetzt. Auch Endanwender nutzen das Verfahren zur Abnahmeprüfung von Software.

Geeignete Testdaten werden in das Zielprogramm eingespeist, um - im Programmcode unberücksichtigte - Eingabedaten zu erkennen; die Verarbeitung dieser ('unerwarteten') Daten führt zu einem fehlerhaften Verhalten (Crash, hoher Verbrauch an Ressourcen wie Rechenzeit, Speicher) des Zielprogramms. Dieses anomale Verhalten des Programms wird mit Hilfe eines Monitoring-Tools protokolliert, voranalysiert und dargestellt. Durch die Analyse der Monitorergebnisse können falsche Hinweise (False Positives) ausgesondert werden. Sicherheitslücken werden durch Reproduzierung der Anomalie und Erstellen eines Exploits nachgewiesen. Für diese im Grundsatz Black-Box-Technik wird ausschließlich der ausführbare Maschinencode benötigt.

Erst durch den Einsatz dieser drei Verfahren kann sichergestellt werden, dass die Forderungen der Richtlinie und des Protection Profile nach vertrauenswürdiger, Sicherheitslücken-freier Software erfüllt werden.