Ein Ansatz zur Intrusion Detection für Prozessautomatisierungssysteme

Martin Naedele
ABB Corporate Research
CH-5405 Baden-Dättwil
Schweiz

Abstract: Auf Grund anwachsender Vernetzung wird Informationssystem- und Netzwerksicherheit immer stärker auch zu einem wichtigen Thema für Automatisierungs- und Prozessleitsysteme, wie sie z. B. in Fabriken oder zur Steuerung von Infrastruktureinrichtungen wie Gas-/Wasser-/Strom-Netzen verwendet werden.

Ein Sicherheitskonzept für ein System, das mit öffentlichen Netzen verbunden ist, sollte nicht nur verzögernde Elemente (z. B. Firewalls), sondern auch Erkennungs- und Reaktionskomponenten besitzen. In heutigen Automatisierungssystemen werden die beiden letzten Aspekte häufig vernachlässigt, da die existierenden Mechanismen nicht mit den speziellen Anforderungen derartiger Systeme kompatibel sind und nur wenige speziell für den Automatisierungsbereich entwickelte Mechanismen existieren.

In diesem Paper wird ein Ansatz zur Intrusion Detection für Prozessautomatisierungssysteme vorgestellt, der den Prozessbediener, der typischerweise kein IT-Experte ist, aufbauend auf seiner spezifische Erfahrung mit der Überwachung von Prozessen, als wichtiges Erkennungs- und Entscheidungselement in den Erkennungsprozess integriert. Neben der Begründung und Beschreibung dieses Ansatzes stellt das vorliegende Paper auch eine prototypische Implementierung vor.

1 Einführung

1.1 Hintergrund

Automatisierungssysteme beeinflussen viele Aspekte unseres Alltags. In der Form von Fabrikautomatisierungs- und Prozesskontrollsystemen ermöglichen sie hohe Produktivität in der industriellen Fertigung, und in der Form von Systemen für Strom-, Gas-, und Wasserversorgungsunternehmen bilden sie das Rückgrat unser technischen Zivilisation. Heutzutage bestehen die meisten Automatisierungssysteme aus verschiedenen Teilen, die im LAN, WAN, oder sogar weltweit verteilt sind. Bis heute sind die meisten dieser Systeme gegeneinander und gegenüber öffentlichen Netzen wie dem Internet isoliert. In den letzten Jahren jedoch werden Automatisierungssysteme, auf Grund von Anforderungen des Marktes und auch der Verfügbarkeit entsprechender neuer Technologien, zunehmend miteinander vernetzt, um Reaktionszeiten zu verringern, Entscheidungen zu optimieren, und die Zusammenarbeit und Koordination zwischen Fertigungsstandorten, Unternehmen und Industriebranchen zu verbessern. Anfänglich basierten derartige Vernetzungen auf spezia-

lisierten, kaum öffentlich dokumentierten, proprietären Protokollen. Heutzutage wird zunehmend Internet-Technologie zu diesem Zweck verwendet, weshalb Informations- und Netzwerksicherheit nun auch für die Automatisierungstechnik relevant sind.

1.2 Problemstellung

Eine Sicherheitsarchitektur für ein Informationssystem darf nicht nur Elemente enthalten, die den Angreifer verzögern, in dem sie bestimmte Angriffspfade verschließen (z. B. Firewalls), sondern gleichzeitig muss auch die Entdeckung eines Angriffs ermöglicht werden und Gegenmaßnahmen müssen initiiert werden [Sc99].

Der Einsatz von automatisierten, "intelligenten" Intrusion Detection Systemen zur Erkennung von Angriffen stößt heute noch in vielen Umgebungen auf praktische Schwierigkeiten. Das Hauptproblem, wie z. B. in den verschiedenen Beiträgen in [VJK03] erwähnt, heutiger regelbasierter und auch statistischer IDSs ist die hohe Anzahl von Fehlalarmen, die sie erzeugen. Diese Fehlalarme müssen manuell von menschlichen Experten bearbeitet werden, was in den meisten Fällen eine verzögerungsfreie Reaktion auf den Alarm verhindert.

Eine sofortige Reaktion auf einen möglichen Angriff ist jedoch in industriellen Automatisierungssystemen notwendig, insbesondere, wenn Menschenleben oder Umwelt durch eine Manipulation oder durch Verlust der Kontrolle über den Prozess gefährdet sein könnten. Die Reaktion muss hierbei zielgerichtet auf die Verhinderung von Schäden am überwachten Produktionsprozess ausgerichtet sein. Eine genaue Analyse des Angriffs ist nicht notwendig oder kann zeitverschoben erfolgen.

Gleichzeitig ist in vielen Firmen, die Automatisierungsanlagen betreiben, die Verfügbarkeit von Mitarbeitern mit IT- oder gar IT-Sicherheits-Kenntnissen noch stärker beschränkt als bei Firmen mit Büroinformationssystemen vergleichbarer Größe, da die einmalige Installation häufig von externen Firmen ausgeführt wird und das System im Laufe seiner Einsatzdauer nicht mehr verändert wird.

Andererseits wird die Erkennung von Angriffen und Reaktion auf diese durch bestimmte Eigenheiten von Prozessautomatisierungssystemen auch erleichtert:

- Die Verbindung zum externen Netzwerk, einschließlich des Firmenintranets ist normalerweise nicht unbedingt notwendig. Obwohl ein längerer Verbindungsunterbruch zu Unannehmlichkeiten und möglicherweise finanziellen Verlusten führen kann, hat er im Normalfall keine katastrophalen Auswirkungen für das Unternehmen.
- Die Daten, die die Grenze zwischen Automatisierungsnetzwerk und Intranet überschreiten können auf bestimmte Datentypen, z. B. nur nicht-ausführbaren Code, beschränkt werden.
- Die Topologie des Automatisierungsnetzwerks, die verwendeten Applikationen und ihre Kommunikationsbeziehungen sind vergleichsweise statisch.

- Aus Sicherheitserwägungen (Security und Safety) und um Trainingskosten zu reduzieren, ist der Arbeitsplatzrechner des Prozessbedieners häufig so konfiguriert ist, dass das Prozesskontrollsystem die einzige Anwendung ist, zu der er Zugang hat.
- Dadurch, dass die relativ geringe Zahl zulässiger Kommunikationsbeziehungen im Automatisierungsnetzwerk und in dessen Anschluss an die Außenwelt zur Konfigurationszeit bekannt sind, können die meisten Angriffe allein schon basierend auf IP Adresse und Port erkannt werden, ohne dass eine Inspektion des Paketinhalts notwendig ist.
- Da ein Prozessleitsystem nur wenige legitime Benutzer hat und diese ein sehr vorhersagbares Anmeldeverhalten aufweisen, z. B. korreliert mit dem Schichtwechsel, können ungewöhnliche Anmeldeereignisse mit hoher Wahrscheinlichkeit als Anzeichen für einen Angriff eingestuft werden.
- Das Verhalten des Prozesses wird via Prozessleitsysteme normalerweise rund um die Uhr von einem oder mehreren menschlichen Prozessbedienern überwacht.

Der letzte Punkt ist unserer Meinung nach von besonderer Wichtigkeit und bietet eine grosse Chance: Obwohl ein Prozessbediener typischerweise kein IT- oder IT-Sicherheits-Experte ist, sollte eine Sicherheitsarchitektur für ein Prozessleitsystem von seiner Anwesenheit Gebrauch machen und ihn als wertvolles Element in den Erkennungsprozess einbauen, um die Reaktionszeit auf Angriffe zu verringern.

Im Gegensatz zu einem vollautomatisierten IDS, kann der Prozessbediener flexibel seine, durch seine normale Tätigkeit sogar besonders trainierte, menschliche Fähigkeit zur Erkennung visueller Muster, seine Erfahrung über Tages- und Wochenmuster ebenso wie sein Vorwissen über spezielle außerordentliche Aktivitäten, z. B. Anlagenwartung, ohne zeitaufwändige Konfiguration von IDS Regeln in seine Entscheidung einfließen lassen.

Dieses Paper skizziert, wie zur Realisierung eines derartigen Ansatzes die Erfassung und Präsentation sicherheitsrelevante Kenngrössen in ein Prozessleitsystem realisiert werden könnte, und beschreibt den von uns implementierten Prototypen.

2 Verwandte Arbeiten

IT Sicherheit für Automatisierungssysteme und speziell Intrusion Detection für Automatisierungssysteme sind bisher wenig bearbeitete Forschungsgebiete. Uns ist keine frühere Arbeit bekannt, bei der in ähnlicher Weise IT-Sicherheit und Prozessleitsysteme kombiniert wurden.

In [YGF01] wird vorgeschlagen, sicherheitsrelevante SW-Objekte in einem Computersystem mit "Anzeige- und Warnattributen" zu instrumentieren, die dann mit den Mitteln der statistischen Prozesskontrolle (SPC) überwacht werden, um Anomalien im System zu erkennen. Diese Methode benutzt eine aus der Produktionstechnik stammende Methode (SPC) für allgemeine Intrusion Detection, der Ansatz hat aber nichts mit Prozessleittechnik oder der Anwendung speziell in Automatisierungssystemen zu tun.

Netzwerkmanagementwerkzeuge wie CA Unicenter oder IBM Tivoli sammeln und konsolidieren die Ausgaben verschiedener Geräte und Anwendungen im Netzwerk und könnten auch in Automatisierungssystemen eingesetzt werden. Sie sprechen jedoch als Benutzer-Zielgruppe IT-Spezialisten an und zeigen entsprechen qualitative, detaillierte information zum Systemzustand. Sie sind nach unserer Einschätzung ungeeignet für die zeitnahe und abstrakte Präsentation des Zustands der Systemsicherheit in Zahlen und Zeitreihen in einer Form, die es auch einem Nichtfachmann ermöglichen würde, Abweichungen vom normalen Betriebszustand zu erkennen.

3 Überlegungen zum Einsatz des Prozessbedieners als Erkennungselement in einem Intrusion Detection System

3.1 Benutzeroberfläche

Der menschliche Prozessbediener muss durch das Benutzerschnittstelle in die Lage versetzt werden, den Ernst der Sicherheitslage zu beurteilen, ohne dabei auf das Wissen eines IT-Sicherheitsexperten zurückgreifen zu können. Die für seine Entscheidung notwendigen Daten müssen deshalb in einfacher, konsistenter und für ihn intuitiver Weise präsentiert werden.

In Anlehnung an seine normale Arbeitsumgebung und Arbeitsweise für die Überwachung des Produktionsprozesses sollten die sicherheitsrelevanten Parameter des Systems dem Prozessbediener in Form eines Prozessbildes dargeboten werden. In diesem Fall ist der gezeigte Prozess das Informationssystem und Netzwerk des Prozessleitsystems (siehe Abb. 1). Das Prozessbild enthält numerische Anzeigen und verwendet außerdem Farben und Icons zur Zustandsvisualisierung. Trendkurven stellen die historische Entwicklung wichtiger Kenngrössen über verschiedene Zeitspannen dar.

Sicherheitsbezogene Alarme sollten nur für sehr seltene Ereignisse erzeugt werden, denn auf Grund seiner Erfahrungen mit dem Prozessleitsystem erwartet der Prozessbediener, dass bei einem Alarm ein wirklich schwerwiegendes Problem vorliegt, dass sofortiges Handeln erfordert. Wenn er bemerkt, dass das Sicherheitssystem alle paar Minuten einen Alarm unklarer Bedeutung und Wichtigkeit verursacht, wird er dieses Subsystem sehr bald ignorieren oder ausser Betrieb nehmen.

Auch ist es wenig sinnvoll, ihm direkt die Alarmmeldung eines IDS anzeigen, die detaillierte Informationen über das auslösende Datenpakete, die ausgelöste Regel und Verweise auf Dokumentation relevanter Sicherheitsschwachstellen enthält. Diese Angaben müssen stattdessen in Anzeigen und Meldung übersetzt werden, die ein Prozessbediener verstehen kann - ohne IDS/Netzwerk-Jargon - und die konform zu den Erfahrungen aus seiner Haupttätigkeit, der Steuerung eines industriellen Prozesses, sind.

Ereignisse grosser Häufigkeit und solche mit hoher Fehlalarmrate sollten quantitativ behandelt werden. Sie können gezählt, eventuell dabei auch mit ihrer Schwere gewichtet werden und dann zusammen mit anderen quantitativen Kenngrössen in Form graphischer

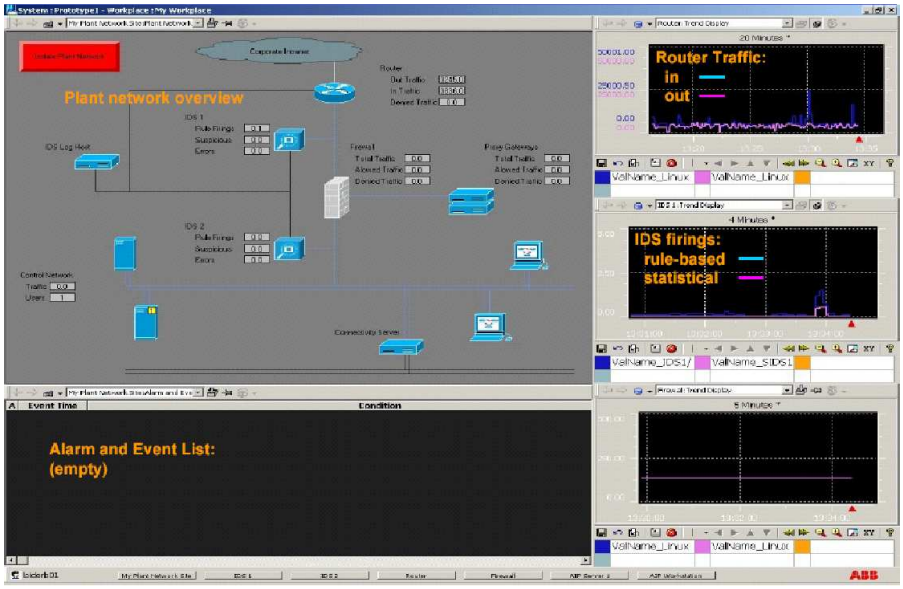

Abbildung 1: Benutzeroberfläche des Prototypen für die Integration von quantitativen Sicherheitsinformationen in ein Prozessleitsystem.

Trends visualisiert werden. Der Prozessbediener kann daraus Zustand und Veränderungstendenz des entsprechenden Parameters beurteilen.

3.2 Sicherheitsrelevante Datenquellen

Die folgenden Datenquellen und Kenngrössen können zur Beurteilung des Sicherheitszustandes im Automatisierungsnetzwerk verwendet werden:

Router, Firewalls, Proxies: Eingehender/ausgehender Verkehr durchgelassen und zurückgewiesen, pro Zeiteinheit, pro Protokoll, pro Quelladresse/Port, pro Zieladresse/Port, pro Regel; Benutzeranmeldungen, fehlgeschlagene Anmeldungen; Veränderung von Regeln.

Netzwerk, Switch: Gesamter Verkehr, pro Protokoll, pro Quelladresse, pro Zieladresse, als Prozentsatz der zur Verfügung stehenden Bandbreite.

Rechner: Benutzeranmeldungen (Anzahl, Konten, Berechtigungslevel), fehlgeschlagene Anmeldungen; Veränderung von Kontenberechtigungen; Prozessorauslastung, Zeit seit letztem Neustart, freier Platz auf Speichermedien; Audit-Ereignisse.

Anwendungen (z. B. Web Server): Benutzeranmeldungen, fehlgeschlagene Anmeldungen; versuchte Verbindungen pro Zeiteinheit; fehlgeschlagene Verbindungen.

IDS: Eventuell kann auch ein regelbasiertes oder statistisches konventionelles IDS als quantitative Datenquelle genutzt werden: Regelauslösungen pro Zeiteinheit, pro Regel; wegen Überlast nicht kontrollierter Verkehr; Benutzeranmeldungen am Sensor-Rechner, fehlgeschlagene Anmeldungen; Veränderung von Regeln, Datum/Zeit des letzten Regel-Updates.

Externe Informationen Als Hintergrundinformation über Virus- und Wurmaktivität könnten auch Daten über den allgemeinen Sicherheitszustand des Internets, z. B. von http://isc.incidents.org/ oder ähnlichen Web Sites eingeblendet werden.

4 Beschreibung des Prototypen

4.1 Architektur

Die Architektur unseres Prototypen soll erlauben, viele verschiedene Datenquellen anzuschließen, die Verarbeitung flexibel und erweiterbar zu halten, und so weit möglich existierende Werkzeuge und Anwendungen zu verwenden. Diese Anforderungen resultieren in der in Abb. 2 dargestellten und im folgenden näher beschriebenen fünfstufigen Architektur.

Datenerzeugung Wie in Kapitel 3.2 beschreiben, sind viele verschiedene Datenquellen möglich. In unserem Prototypen benutzen wir als Datenquellen einen Linux-Rechner konfiguriert als Router und einen Cisco Catalyst 2926 Switch zur Messung des Netzwerkverkehrsaufkommens, ausgewählte Meldungen aus dem Windows 2000 Event Log, die via Intersect Alliance SNARE in das Syslog-Protokoll konvertiert werden, sowie ein Snort und Spade IDS, deren Meldungen wir quantitativ auswerten.

Eingabe Mit spezifischen Eingabeeinheiten ist die Architektur prinzipiell in der Lage, beliebige Eingabeprotokolle im Pull- und Push-Modus zu verarbeiten. Jede Eingabeeinheit besteht typischerweise aus aus einem existierenden Werkzeug zum Datensammeln und einem speziell entwickelten Protokolwandlermodul, das die Eingangsdaten bzw. ihr Format für die weitere Bearbeitung normalisiert. Unser Prototyp unterstützt im Augenblick Syslog und SNMP als Eingabeprotokolle.

Verarbeitung Die Verarbeitungseinheiten konvertieren die Rohdaten in Information für den Bediener. Der Prototyp implementiert bisher vier Arten von Verarbeitungsmodulen:

Wert Das "Wert"-Modul extrahiert einen numerischen Wert aus den Eingabedaten und setzt die entsprechende OPC-Variable auf diesen Wert. Dies wird z. B. für Daten wie Prozessorauslastung und Zeit seit dem letzten Neustart verwendet.

Meldung Das Modul "Meldung" findet für die eingegangene Meldung eine zugehörige Textschablone, deren variable Teile mit Informationen aus der Mel-

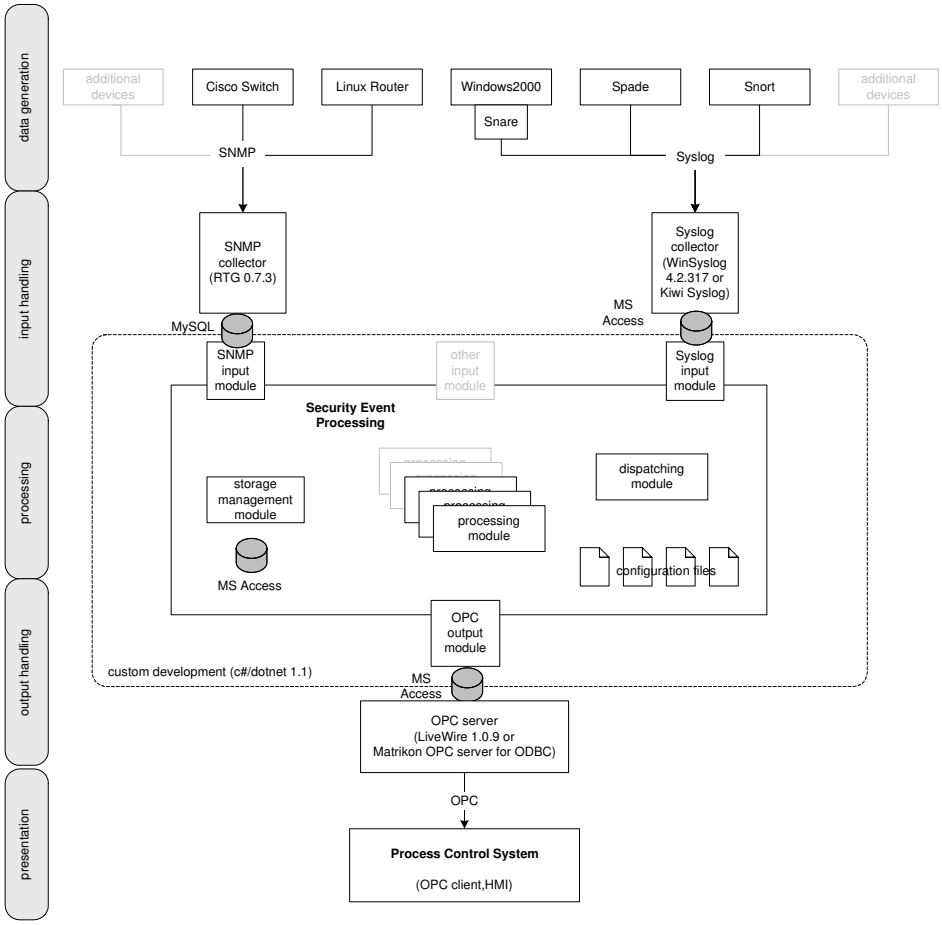

Abbildung 2: Systemarchitektur

dung gefüllt werden. Diese Funktion wird zur Übersetzung von Systemmeldungen in eine für den Prozessbediener verständliche Sprache genutzt. Beispiele für Meldungen sind Systemstart und -abschaltung (Parameter: betroffener Rechner), Benutzerkonto- und Rechteänderungen sowie fehlgeschlagene Anmeldungen (Parameter: betroffenes Konto).

Inkrement/Dekrement Dieses Modul verwaltet einen numerischen Wert auf der Basis von Mitteilungen über Statusänderungen. Es liest, ändert und schreibt die zugehörige Ausgabevariable. Dieses Modul kann z. B. für die Anzeige der Anzahl angemeldeter Benutzer verwendet werden.

Rate Das "Raten"-Modul berechnet die Anzahl bestimmter Ereignisse pro konfigurierbarer Zeiteinheit. Im Prototypen verwenden wir dieses Modul für die Berechnung der NIDS/SIDS-Feuerraten und das Netzwerkverkehrsaufkommen

(durchgelassen/zurückgewiesen) am Router.

Ausgabe Das System muss die Daten in einer Weise bereitstellen, mit der das Prozessleitsystem bzw. dessen Visualisierungskomponente etwas anfangen kann. Heutzutage ist das am weitesten verbreitete, herstellerunabhängige Protokol zum Austausch von Informationen zwischen Automatisierungskomponenten OPC, speziell OPC-DA [OP], welches auf Microsoft (D)COM basiert. Dieses Protokol wird auch in unserem Prototypen verwendet, aber natürlich kann die Ausgabeeinheit ersetzt werden, wenn z. B. der Wunsch nach einer HTML/HTTP basierten Visualisierung besteht, oder um Web Services in der Form der OPC-XML Protokollfamilie einzusetzen.

Visualisierung Die augenblicklichen und historischen Werte der sicherheitsrelevanten OPC-Variablen werden auf der Oberfläche des Prozessüberwachungssystems in einem Prozessbild dargestellt. Die Standardfunktionalität der Trendkurvenanzeigen erlaubt dann die Erkennung von stündlichen, täglichen, wöchentlichen und monatlichen Mustern durch interaktive Veränderung des angezeigten Zeitraumes. Die Benutzeroberfläche enthält auch Eingabemöglichkeiten zur Auslösung vordefinierter Gegenmaßnahmen, z. B. Isolierung eines Netzwerksegments.

Die Eingabe- und Verarbeitungseinheiten unseres Prototypen sind in C# für Microsoft .Net 1.1 geschrieben. Die einzelnen Stufen des Prototypen sind voneinander durch Datenbanken entkoppelt. Dies ist die Ursache für gewisse Durchsatzprobleme und somit nicht für ein Produkt geeignet, hat sich aber hier als bequemer und flexibler Mechanismus angeboten, insbesondere da die eingebundenen Werkzeuge wie RTG und WinSyslog standardmässig über Datenbanken kommunizieren.

4.2 Funktion

Die einzelnen Wartezyklen werden im Abstand einer konfigurierbaren Wartezeit angestoßen. Auf Grund der unterschiedlichen Verarbeitungszeiten pro Zyklus erzeugt diese einfache Lösung natürlich Jitter, aber dies bleibt ohne schwerwiegende Konsequenzen, da die Berechnung der Raten auf dem tatsächlichen Zeitintervall zwischen Datenerhebungen und und nicht die Abfrageperiode beruht. Ein Verarbeitungszyklus besteht aus dem Abfragen aller Eingabeeinheiten und dem anschließenden Bearbeiten der Eingaben. Abb. 3 zeigt die Verarbeitung der von den verschiedenen Werkzeugen gelieferten Eingaben. Dieser Ablauf demonstriert auch die verschiedenen Stellen, an denen die Systemfunktion durch Einträge in Konfigurationsdateien verändert oder ergänzt werden kann, ohne dass ein Rekompilierung der Anwendung notwendig ist.

1. Eine Datenquelle, hier z. B. das Snort NIDS, sendet Daten über ein möglicherweise sicherheitsrelevantes Ereignis via Syslog Protokoll an den Syslog-Server, der dann einen entsprechenden Eintrag in eine ODBC-Datenbank schreibt. Diese zwei Schritte, die ganz auf existierenden Werkzeugen beruhen, sind in der Abbildung nicht gezeigt. Im Laufe eines Verarbeitungszyklus liest dann das Syslog-Eingabemodul die neuen Meldungen aus der Syslog-Datenbank.

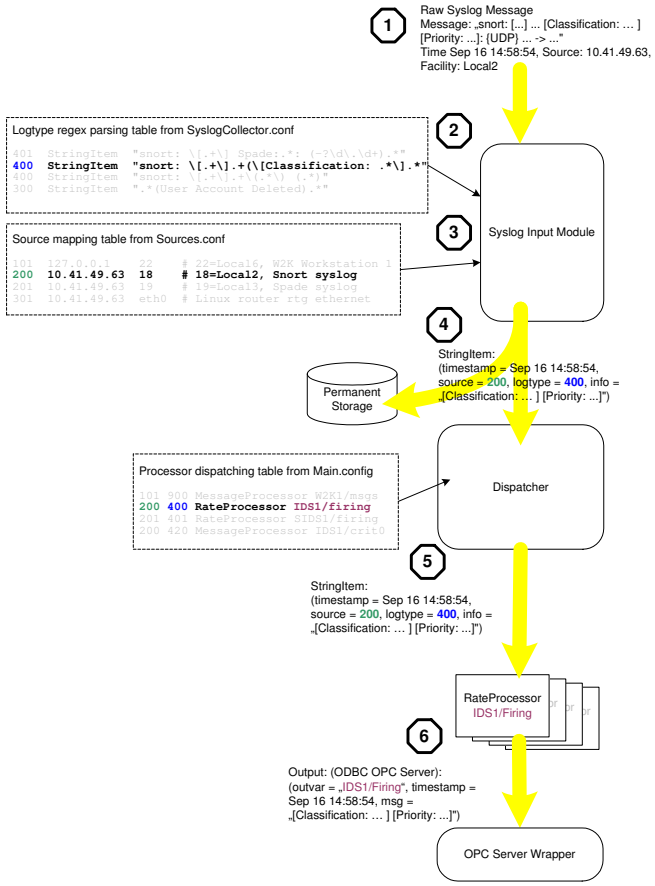

Abbildung 3: Beispiel für einen Verarbeitungszyklus.

2. Durch Vergleich mit den in einer Konfigurationsdatei festgelegten Regeln bestimmt das Eingabemodul den logischen Typ der Meldung.

3. Im nächsten Schritt werden Syslog Facilty und IP-Adresse der Datenquelle auf einen logischen Quellenidentifikator abgebildet.

4. Zuletzt erzeugt das Eingabemodul ein Datenobjekt, das es dann einerseits an die Einheit zur Langzeitspeicherung und andererseits an den Verarbeitungs-Verteiler übergibt.

5. Der Verteiler bestimmt aus dem logischen Meldungstyp und der logischen Quelle die Verarbeitungsmodulinstanz(en), an die die Meldung übergeben werden muss, und auch die OPC-Variable, die jeweils das Ergebnis der Verarbeitung aufnimmt. Da die Implementierung ausschließlich abstrakte Interfaces und Introspektion für das

Lebenszyklusmanagement der Objekte verwendet, können neue Verarbeitungsmodule einfach zur Konfigurationszeit dem System hinzugefügt werden. Man muss dazu nur die entsprechenden .Net Klassen bereitstellen und die notwendigen Zeilen in der Konfigurationsdatei `Main.Config` ändern oder hinzufügen. Mittels spezieller Methoden wird jedes Verarbeitungsmodul auch über Beginn und Ende jedes Verarbeitungszyklus informiert. Dies erlaubt es, die eingegangenen Meldungen nicht nur einzeln zu verarbeiten, sondern auch in Paketen, um z. B. Raten zu errechnen oder Duplikate zu entfernen.

6. Die Verarbeitungsmodule teilen der Schnittstelle zum OPC-Server mit, welche Variable als Resultat der Bearbeitung auf welchen Wert gesetzt werden muss.

4.3 Beschränkungen

Der Prototyp weist eine Reihe technischer Beschränkungen auf, die jedoch in einer Produktentwicklung leicht beseitigt werden könnten: Die verwendeten Datenbanken MySQL und MS Access hatten beide Probleme in Bezug auf Durchsatz und Stabilität in bestimmten Situationen. Die Verarbeitungsmodule können nicht direkt Alarme unter Benutzung der Alarmmanagementfunktionalität des Prozessleitsystems erzeugen, da die beiden verwendeten Demoversionen von OPC-Servern (Matrikon und PlantLIVE) den OPC-AE (Alarm & Event) Standard nicht unterstützen. Ausserdem könnte auch Klassifizieren der Eingangsmeldungen mittels regulärer Ausdrücke zu einem Durchsatz-Engpass werden, wenn auf noch mehr verschiedene Typen hin getestet werden muss.

5 Ergebnisse und nächste Schritte

Unsere ersten, beschränkten Experimente mit diesem Ansatz und dem hier beschriebenen Prototypen haben gezeigt, dass verschiedene Scans, die ein Angreifer vermutlich ausführen würde (z. B. mit Werkzeugen wie nmap und nessus), klar erkennbare visuelle Anomalien in den Trendkurven hervorrufen (siehe Abb. 4). Ähnliche klare Muster erwarten wir von Würmern und anderen Denial-of-Service-Angriffen. Während unser Prototyp an das Firmenintranet angeschlossen war, konnten wir auch klar beobachten, wie sich die Verkehrsmuster auf Grund der regulären Systemmanagementaktivitäten der IT-Abteilung über den Tag hinweg ändern.

Als nächstes planen wir, die Installation des Prototypen in einem größeren Labornetzwerk mit einer realistischen Installation des Prozessleitsystems, und anschließend dann einen Feldversuch in einer Fabrikumgebung, um an echten Prozessbedienern unsere Hypothese zu validieren, dass die quantitative Darstellung des Sicherheitsstatus eine sinnvolle Bearbeitung auch durch Nichtexperten ermöglicht.

Basierend auf einem solchen Feldversuch wäre dann der nächste Schritt zur Produktisierung, festzustellen, welche Untermenge von Kenngrössen die beste Information bei kleins-

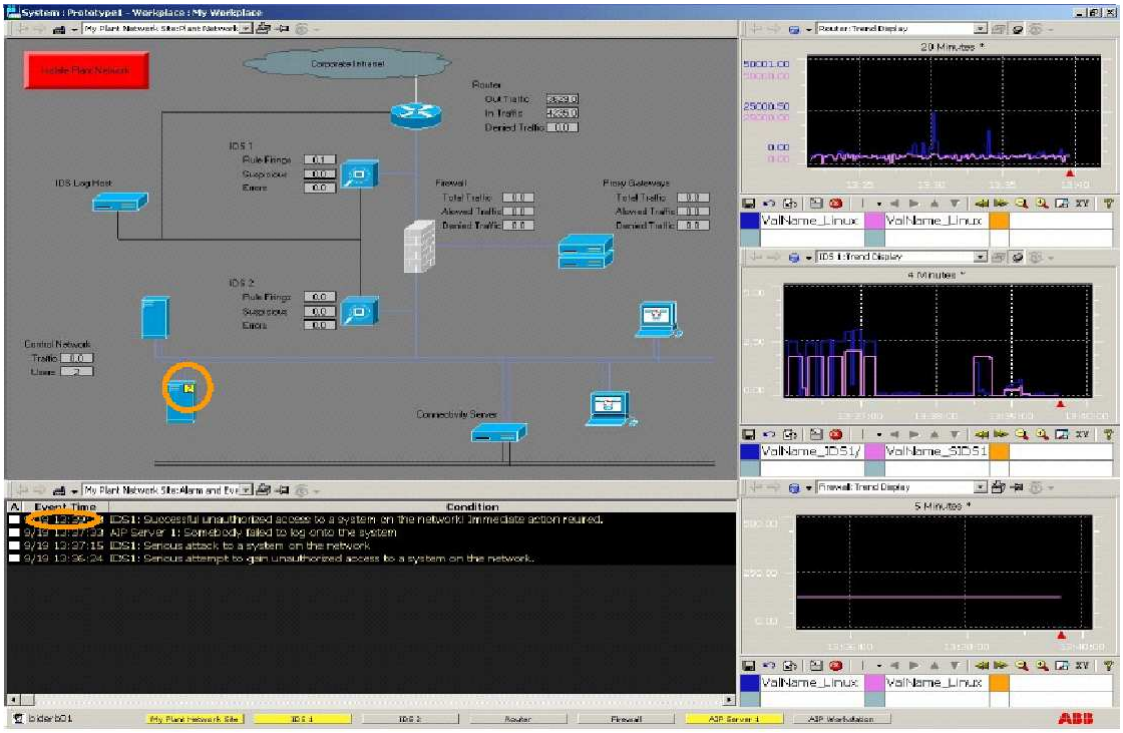

Abbildung 4: Auf dieser Bildschirmaufnahme sieht man Scanning-Aktivität und die Änderung der Anzahl der angemeldeten Benutzer.

tem ”Rauschen”liefert und gleichzeitig bei einer grossen Anzahl verschiedener Anlagen anwendbar ist, um den Inbetriebnahme- und Anpassungsaufwand pro Anlage zu minimieren.

Zusätzliche Arbeit ist auch in Bezug auf die ergonomische Gestaltung des IT-Sicherheits-Prozessbildes notwendig.

6 Zusammenfassung

Dieses Paper stellt einen neuen Ansatz zur Intrusion Detection für Prozessleitsysteme in industriellen Anlagen vor. Der Grundgedanke hierbei ist, nicht vollständig auf die ”Intelligenz” des IDS zu vertrauen, sondern nur die Kenngrössen des IT-Systems im Prozessleitsystem quantitativ zu visualisieren und die menschliche Intelligenz und ausgeprägte Fähigkeit zur Mustererkennung des ohnehin anwesenden Prozessbedieners zu nutzen um kritischen Anomalien zu erkennen und darauf angemessen zu reagieren.

Es wurde außerdem eine prototypische Anwendung zur Umsetzung dieses Ansatzes vorgestellt. Diese Anwendung sammelt sicherheitsrelevante Daten in ihren spezifischen Protokollen ein und bereitet sie für die Präsentation im Prozessleitsystem auf.

Erste Experimente mit diesem System waren vielversprechend, aber weitere Arbeiten sind notwendig, um den Nutzen und die Benutzbarkeit dieses Ansatzes zu verbessern. Eine in letzter Zeit steigende Anzahl von Anfragen unserer Kunden zu diesem Thema zeigt uns, dass ein grosser Bedarf nach IDS-Funktionalität speziell für Automatisierungssysteme besteht, der mit konventionellen Systemen schwer zu befriedigen ist.

Ergänzend zu diesen Arbeiten wäre es natürlich auch interessant, zu untersuchen, ob die in Kapitel 1.2 beschriebenen Eigenheiten der Topologie und des dynamischen Verhaltens von Prozessleitsystemen, auch bei Einsatz von konventionellen "intelligenten" Intrusion Detection Systemen zu einer vertretbar niedrigen Fehlalarmrate führen würden.

Literatur

[OP] OPC Foundation. http://www.opcfoundation.org/.

[Sc99] Schwartau, W.: *Time based Security*. Interpact Press. 1999.

[VJK03] Vigna, G., Jonsson, E., und Kruegel, C. (Hrsg.): *Recent Advances in Intrusion Detection (Proceedings 6th Int. Symposium RAID 2003)*. Number 2820 in LNCS. Springer. 2003.

[YGF01] Ye, N., Giordano, J., und Feldman, J.: A process control approach to cyber attack detection. *Communications of the ACM*. 44(8):76–82. 2001.