Der elektronische Sicherheitsinspektor eSI: Ein Tool zur dynamischen Analyse der IT-Sicherheit eines Systems

H. Sarbinowski, T. Shafi, C. Eckert

Fraunhofer Institut für Sichere Telekooperation (SIT)
Rheinstr. 75, D-64295 Darmstadt
{eckert, sarbinowski, shafi}@sit.fraunhofer.de

Zusammenfassung: Um in heutigen Systemen ein angemessenes Sicherheitsniveau zu erzielen, gehört es zur gängigen Praxis, ein IT-Sicherheitskonzept zu erstellen, die erforderlichen Sicherheitsmaßnahmen umzusetzen und von Zeit zu Zeit anlässlich eines Sicherheitsaudits diese stichprobenartig zu überprüfen. Das weitergehende Ziel, einen kontinuierlichen IT-Sicherheitsprozess einzurichten mit einem stets aktuellen Überblick über die gegenwärtige „Sicherheitslage" scheitert häufig an der fehlenden technischen Unterstützung. Mit dem „elektronischen Sicherheitsinspektor" (eSI) wird ein Tool entwickelt, um diesen Mangel zu beseitigen. Die Aufgabe des eSI ist es, die im IT-Sicherheitskonzept beschriebenen Maßnahmen, soweit wie möglich, kontinuierlich durch automatisch durchgeführte Tests auf ihren aktuellen Umsetzungsstand hin zu überprüfen. Auf diese Weise wird der Sicherheitsverantwortliche in die Lage versetzt, den aktuellen Status der Umsetzung der Sicherheitsmaßnahmen zu beurteilen. In einer ersten Realisierungsphase des Tools wurden Maßnahmen aus dem IT-Grundschutzhandbuch des BSI zur Überprüfung ausgewählt.

1 Einleitung

Sicherheitsbewusste Anwender erstellen Sicherheitskonzepte (u.a. [Eck03]), in denen die Sicherheitsmaßnahmen beschrieben sind, mit denen die gefährdeten Werte und Objekte geschützt und somit die nicht akzeptablen Risiken reduziert oder sogar ausgeschlossen werden können (u.a. [Zie01], [And01]). Orientiert man sich beispielsweise am Grundschutzhandbuch (GSHB) des BSI [GSHB], so können sich die umzusetzenden Maßnahmen auf die Bereiche Infrastruktur, Organisation, Personal, Hardware/Software, Kommunikation und Notfallvorsorge beziehen. Nach Umsetzung dieser im Sicherheitskonzept beschriebenen Maßnahmen findet jedoch in der Regel keine systematische Überprüfung der Maßnahmen mehr statt (u.a. [Mü99]). Ein Grund dafür ist in der fehlenden Tool-Unterstützung zu sehen. Dies motivierte die Entwicklung des elektronischen Sicherheitsinspektors eSI am Fraunhofer Institut für Sichere Telekooperation (FhI-SIT), mit dessen Hilfe eine entsprechende Tool-unterstützte Prüfung stattfinden kann.

1.1 Überprüfbare Maßnahmen

Ausgangspunkt für eine Tool-unterstützte, automatisierte Überprüfung ist eine Analyse der Sicherheitsmaßnahmen, um diejenigen Maßnahmen zu identifizieren, die für eine au-

tomatisierte Überprüfung überhaupt geeignet sind. Da Sicherheitsmaßnahmen z. T. unternehmensindividuell sind, haben wir uns bei der Entwicklung des eSI-Tools zunächst auf Grundschutz-Maßnahmen konzentriert, wie sie im Grundschutzhandbuch des BSI (GSHB) beschrieben sind.

Bei den in Frage kommenden Maßnahmen für zu schützende Objekte ist zunächst festzustellen, inwieweit sie für eine automatisierte Überprüfung geeignet sind. GSHB-Maßnahmen wie etwa „M 3.13: Sensibilisierung der Mitarbeiter für mögliche TK-Gefährdungen“ sind nur auf Umwegen – wenn überhaupt – technisch überprüfbar. Derartige Maßnahmen wurden daher bei der Tool-Entwicklung nicht weiter betrachtet. In einem nächsten Schritt ist festzustellen, ob es sich bei einer Maßnahme um eine elementare Maßnahme oder eine zusammengesetzte Maßnahme handelt. Elementare Maßnahmen enthalten konkrete Vorgaben, die nicht in weitere Teilmaßnahmen untergliedert sind. Hierzu gehört etwa die GSHB-Maßnahme „M 4.96 Abschaltung von DNS“. Die Mehrzahl der Maßnahmen des GSHB gehört jedoch zur Kategorie der zusammengesetzten Maßnahmen, wie z.B. die Maßnahme „M 2.174 Sicherer Betrieb eines WWW-Servers“. Um aus einer zusammengesetzten Maßnahme die Elementarmaßnahmen zu erhalten ist sowohl der Maßnahmentext hinsichtlich expliziter oder impliziter Hinweise zu analysieren als auch eigene IT-Sicherheitsexpertise erforderlich, um gegebenenfalls fehlende Teilmaßnahmen hinzuzufügen. Sind alle Teilmaßnahmen einer zusammengesetzten Maßnahme bekannt, werden die Prüfmethoden und erwarteten Prüfergebnisse festgelegt. Ziel ist es hierbei, die erforderlichen Prüfergebnisse so festzulegen, dass etwa die Frage „Ist die Maßnahme x für das zu schützende Objekt y umgesetzt?“ beantwortet werden kann.

Bei den Prüfmethoden werden fallspezifisch sowohl direkte als auch indirekte Methoden eingesetzt. Bei der direkten Prüfmethode wird etwa die Prozesstabelle überprüft, ob ein bestimmter, durch die zu prüfende Maßnahme vorgegebener, Prozess tatsächlich läuft. Bei der indirekten Prüfmethode wird das externe Verhalten eines zu schützenden Objekts geprüft. So kann durch einen „Portscan“ festgestellt werden, ob ein bestimmter Port für eine Anwendung geschlossen oder geöffnet ist. Das Ergebnis der Maßnahmenanalyse fließt in eine elektronische Checkliste ein, in der für jede Maßnahme und jedes zu schützende Objekt die Prüfmethode und die Ergebniskriterien enthalten sind.

Der Beitrag gibt in Kapitel 2 einen Überblick über die Architektur und Komponenten des aktuellen eSI-Prototyps, und erläutert in Kapitel 3 anhand eines konkreten Szenarios den Ablauf eines Prüfvorganges.

2 eSI-Architektur

Das Werkzeug eSI soll den Sicherheitsbeauftragten darin unterstützen, im Rahmen des IT-Sicherheitsmanagements die Wirksamkeit der umgesetzten Sicherheitsmaßnahmen für zu schützende Objekte kontinuierlich zu überprüfen. Hierbei beschränken sich seine Fähigkeiten auf die *technische* Überprüfbarkeit von Maßnahmen. Dazu wurde eine Architektur entwickelt und prototypisch umgesetzt, in die existierende Prüfwerkzeuge und Analysetools, wie beispielsweise Whisker [Whisk], Nmap [Nmap] oder ein Virenscanner, integriert und über eine einheitliche Schnittstelle zur automatischen Durchführung von Maßnahmen-Überprüfungen eingesetzt werden können. Der Sicherheitsbeauftragte stößt lediglich die Überprüfung einer Maßnahme an, wie beispielsweise die Überprüfung, ob auf

einem WWW-Server alle CGI-Scripte entfernt wurden, und lässt sich über das Ergebnis informieren. Hierbei überlässt er es dem eSI, welche Maßnahme mit welchem Werkzeug zu prüfen ist, wie das Werkzeug zu benutzen ist und wie die Ergebnisse zu analysieren und zu bewerten sind. Für das Beispiel der CGI-Scripte würde das bedeuten, dass das eSI-Werkzeug entscheidet, beispielsweise das Tool Whisker für diese Überprüfung zu starten und gleichzeitig Whisker mit den benötigten Eingabedaten, wie die IP-Adresse des zu prüfenden Rechners, versorgt. Anhand der in der eSI-Datenbank gespeicherten Informationen über Prüfmethode und die Ergebniskriterien, ermittelt das ausgewählte Werkzeug Whisker, ob auf dem zu analysierenden Rechner Schwachstellen vorliegen, nach denen mit den Prüfmethoden gesucht werden soll. Das Analyseergebnis wird dann durch weitere eSI-Komponenten für den Benutzer in verständlicher Form aufbereitet.

Die Architektur des eSI-Tools ist in Abbildung 1 dargestellt. Dessen Komponenten werden im Folgenden erklärt und deren Zusammenspiel anhand eines exemplarischen Kontrollablaufs wird verdeutlicht.

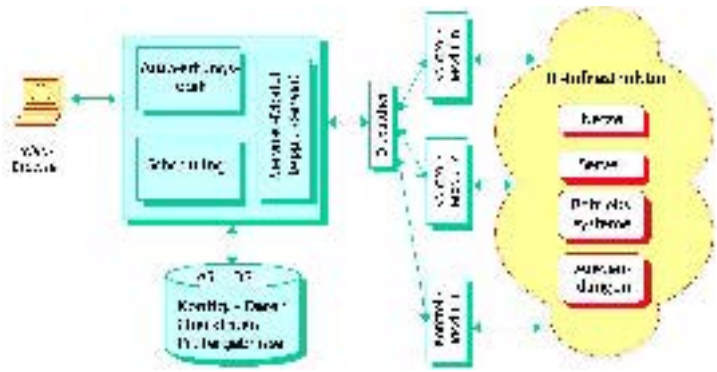

Abbildung 1: eSI-Architektur

2.1 Die Benutzungsschnittstelle (Web-Browser)

Als Schnittstelle zu dem Benutzer des Systems wird ein webbasiertes GUI auf Basis von JSP (Java Server Pages) [Ber01] eingesetzt. Hierdurch werden sowohl die Hardware- als auch die Software-Anforderungen an den Client minimiert. Es wird lediglich das Vorhandensein eines Webbrowsers mit SSL-Verschlüsselung (vgl. u.a. [Eck03]) vorausgesetzt. Der eSI kann so von jedem System innerhalb des lokalen Netzes durch den Sicherheitsbeauftragten benutzt werden.

2.2 Die eSI-Datenbank

Die eSI-Datenbank enthält Informationen über zu schützende Objekte wie z.B. DNS-Name, IP-Adresse, die Wichtigkeit des Objekts, die zuständige Person, implementierte Maß-

nahmen, die anzuwendenden Prüfmethoden und die erwarteten Prüfergebnisse. Die Information wird in Form von Checklisten gespeichert. Sowohl für einzelne Objekte als auch für die in Checklisten zusammengefassten Objekte kann das Prüfscheduling individuell vorgegeben werden (z.B. wann, oder in welchen Abständen wiederholt). Die Prüfergebnisse werden in aufbereiteter Form dem Sicherheitsbeauftragten zugänglich gemacht und in einer Log-Datei aufbewahrt. Für bestimmte kritische Prüfergebnisse kann auch vorgegeben werden, ob z.B. eine Mitteilung an einen Sicherheitsverantwortlichen zu senden ist.

Zur Implementierung wurde die Open Source relationale Datenbank PostgreSQL verwendet, sowie ein einfaches Datenmodell entworfen, in dem die Informationen über die zu schützenden Objekte erfasst sind. Ferner wurde eine Standardschnittstelle mit JDBC definiert, die ohne großen Aufwand die Benutzung anderer Datenbanken wie etwa MySQL, Oracle, etc. ermöglicht.

2.3 Der Dispatcher

Die Durchführung von Kontrollen wird von der Dispatcherkomponente zusammen mit dem Kontrollmodul-System (vgl. 2.4.) übernommen. Der Dispatcher bietet eine generische Schnittstelle, die von dem jeweilig eingesetzten Prüfwerkzeug unabhängig ist, und damit eine flexible Erweiterbarkeit der Menge der integrierbaren Prüfwerkzeuge ermöglicht. Ferner unterstützt der Dispatcher auch die Aufteilung des Prüfvorganges auf verschiedene Rechner, so dass das eSI-Tool auch für komplexe Infrastrukturen effizient einsetzbar ist. Der Dispatcher nimmt Prüfaufträge entgegen und leitet diese an die zuständigen Prüfwerkzeuge, die in so genannte Kontrollmodule gekapselt sind, weiter. Sowohl die Prüfaufträge als auch die Prüfergebnisse werden im XML-Format über Kommunikationsprotokolle (u.a. RMI, SOAP) ausgetauscht. Der Dispatcher besteht aus einem Auftragsmanager, der die zu bearbeiteten Aufträge verwaltet und die Durchführung der Überprüfung anstößt, sowie aus einem Kontrollmodul-Manager, der vom Auftragsmanager beauftragt wird, ein bestimmtes Kontrollmodul zu laden und auszuführen.

2.4 Die Kontrollmodule

Die Einführung des Konzepts der Kontrollmodule hatte zum Ziel, von konkreten Prüfwerkzeugen und Programmen abstrahieren zu können und es dem eSI zu ermöglichen, unabhängig von der Art des Kontrollmoduls die Durchführung einer Kontrolle zu starten, sie zu stoppen oder zu pausieren.

Zurzeit existieren drei Klassen von Kontrollmodulen. Jedes Modul der ersten Klasse besteht aus einem externen Programm (z. B. Nmap, Whisker) zusammen mit einem Wrapper, welcher die individuelle Struktur der Ergebnisse, die das Programm liefert, an die der generischen Schnittstelle des Dispatchers anpasst. Der Wrapper veranlasst und überwacht das Ausführen des externen Programms (.exe, .com,...). Hierzu ist es erforderlich, die Parameter des Programms so zusammenzustellen, dass die vom eSI gewünschten Informationen erfasst werden können.

Bei Kontrollmodulen der zweiten Klasse kann auf die Implementierung eines Wrappers verzichtet werden, da sie direkt in das eSI-Tool eingebunden werden können. Als Beispiel

ist hier die Anbindung des Virenscanners OpenAntiVirus an den eSI zu nennen. Da sowohl der eSI, als auch das anzubindende Werkzeug in Java erstellt wurden, besteht die Möglichkeit, direkt auf die Klassen des Werkzeugs zuzugreifen und es somit als Bestandteil des Kontrollmoduls auszuführen. Lediglich die Struktur der Ergebnisse muss an die des eSI angepasst werden.

Die dritte Möglichkeit besteht darin, eigene Kontrollmodule zu entwickeln, welche direkt den Anforderungen des eSI gerecht werden, und somit nicht adaptiert werden müssen (z.B. das SSL/TLS-Kontrollmodul für sichere Kommunikationsverbindungen).

2.5 Die Auswertungskomponente

Die Auswertungskomponente analysiert die Prüfergebnisse der Kontrollmodule und zieht Rückschlüsse auf den Umsetzungszustand der geprüften Maßnahmen. Die Auswertungskomponente ist zurzeit in einer prozeduralen Variante implementiert; eine deklarative Variante ist jedoch in Arbeit und soll den bestehenden Ansatz ablösen. Die Auswertungskomponente enthält die Regeln, die auf die Prüfergebnisse angewendet werden und die es ermöglichen, Aussagen über den Umsetzungszustand der Sicherheitsmaßnahmen zu treffen. Der entscheidende Vorteil der deklarativen Variante ist die leichte Erweiterbarkeit um das Wissen zur Auswertung von Prüfergebnissen (Regeln). Um dies zu gewährleisten, stützt sich der in Arbeit befindliche Ansatz auf eine deklarative Wissensrepräsentation (in PROLOG). Damit lassen sich Veränderungen am IT-Sicherheitskonzept leicht nachpflegen. Derartige Veränderungen ergeben sich etwa durch den periodisch durchgeführten Aktualisierungsprozess (gemäß GSHB des BSI).

2.6 Das Scheduling

Die Schedulingkomponente ermöglicht es, die Überprüfung von Maßnahmen nach einem individuellen Plan durchzuführen. Folgende Funktionalitäten sind für das Scheduling vorgesehen:

- Individuelle ad-hoc-Kontrollen, die durch den Sicherheitsbeauftragten angestoßen werden.
- Zeitgesteuerte Kontrollen: Zu vorgegebenen Zeiten werden diese Kontrollen durchgeführt. Da bestimmte Kontrollen (wie etwa das Scannen eines zu schützenden Objekts auf angebotene Dienste) zusätzliche Ressourcen (Netzlast, Rechnerlast) benötigen, können sie in die für die Routinenutzung unkritischen Zeitabschnitte gelegt werden.
- Wiederholen einer Kontrolle: Wenn zu einem vorgesehenen Zeitpunkt eine Kontrolle nicht durchgeführt werden kann, so ist vom Schdeduling ein erneuter Versuch automatisch einzuplanen. Ist es auch nach n Versuchen nicht möglich, die Kontrolle durchzuführen, so ist eine entsprechende Mitteilung zu generieren.
- Bei der Planung von Kontrollen sind „Belastungs“-Metriken zu berücksichtigen, auf deren Basis eine optimierte Belastung während der Kontrolldurchführung arrangiert werden kann. Die Kontrolldurchführung erzeugt typischerweise eine zusätzliche Last auf dem jeweiligen zu schützenden Objekt und im Netz.

Der Screenshot in Abbildung 2 zeigt ein Beispiel für eine zeitgesteuerte Kontrolle.

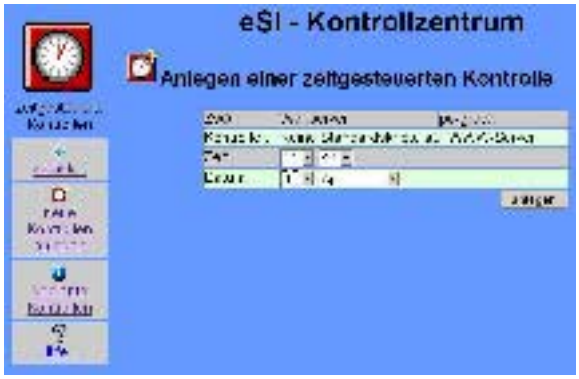

Abbildung 2: Beispiel einer zeitgesteuerten Kontrolle

2.7 Das Servicemodul

Das Servicemodul kommuniziert mit allen anderen beteiligten eSI-Komponenten. Es basiert auf dem Open Source Applikationsserver „JOnAS“. Jonas verfügt über einen integrierten EJB (Enterprise Java Beans) [EJB] Server und benutzt als Webdienst den Open Source Webserver Tomcat. Die serverseitige Anwendungslogik wird mit Hilfe von EJBs realisiert, die in Form von Komponenten Dienste für die Clients zur Verfügung stellen. Die Beans haben die vom EJB Server zur Verfügung gestellte Möglichkeit, über den Namens- und Verzeichnisdienst auf Datenbankverbindungen zuzugreifen. Damit kann erreicht werden, dass ihr Zustand „persistent“ gespeichert wird. Der EJB Server wählt auch das passende Kommunikationsprotokoll (RMI, RMI-IIOP, etc.) aus. Für die Datenspeicherung kann jede JDBC unterstützende Datenbank (MySQL, Oracle, etc.) benutzt werden. Die Datenbank kann auf einem beliebigen Rechner innerhalb des Netzes installiert sein.

Das Servicemodul verfügt über in Java geschriebene Klassen, die aus den (aus der Datenbank erhaltenen) Checklisten einen Kontrollauftrag (im XML-Format) vorbereiten können. Die Klassen dienen auch als Kommunikationskomponenten für den Auswertungslogikteil, den Dispatcher, das Scheduling und das GUI.

3 Ablauf eines eSI-Kontrollauftrags

Am Beispiel einer Teilmaßnahme, die zur Erfüllung der GSHB-Maßnahme „M 2.174 Sicherer Betrieb eines WWW-Servers“ erforderlich ist, wird der Ablauf und Einsatz des eSI-Werkzeugs im Folgenden erläutert. Der sichere Betrieb eines WWW-Servers fordert u.a., dass *die vom Hersteller mitgelieferten CGI-Programme vollständig entfernt werden*.

3.1 Zur Prüfung verwendetes Werkzeug

Diese Maßnahme gilt als umgesetzt, wenn alle vom Hersteller (des WWW-Servers) mitgelieferten CGI-Programme vollständig entfernt wurden. Die Prüfung kann durch das Prüfwerkzeug Whisker [Whisk] erfolgen, das versucht, auf Dateien eines Webservers über das

Netzwerk zuzugreifen. Das Prüfwerkzeug meldet den Status jeder Prüfung in einer Ergebnistabelle. Mit folgendem Aufruf wird beispielsweise das Werkzeug veranlasst, einen Server unter der angegebenen IP-Adresse (192.168.0.120) zu prüfen.

```
rechner:/opt/whisker/v1.4#perl whisker.pl -h 192.168.0.120 -iv
```

Das Ergebnis beinhaltet im Wesentlichen eine Bewertung durch das Plus- („+“) bzw. Minus- („-“) Zeichen und einen String mit zusätzlichen Prüfinformationen. Das Plus bedeutet, dass das Prüfwerkzeug aufgrund des darin repräsentierten Sicherheitswissens keine Schwachstelle finden konnte. Das Minus deutet jedoch auf ein potentielles Sicherheitsrisiko hin. Beispielsweise sagt das Ergebnis `+ 404 Not Found: HEAD /www/` aus, dass von Whisker versucht wurde, auf dem zu prüfenden Server mittels HTTP den Inhalt des Verzeichnisses „/www/“ zu erfragen. Die Antwort (HTTP-Format) des Servers enthält im header-Feld (HEAD) die Zeichenkette „404 Not Found“. Dies bedeutet, dass der Server den angeforderten Pfad nicht kennt und somit auch keinen Inhalt zurückschicken kann. Whisker meldet mit „+“ dem Anwender, dass die in seiner Datenbank vorhandene Schwachstelle auf dem Server nicht gefunden werden konnte.

Ein Ergebnis der Art: `- Content-Location: password.txt` besagt demgegenüber, dass Whisker eine positive Rückmeldung des Servers erhalten hat und gibt deshalb mit dem Zeichen „-“ eine mögliche Schwachstelle bekannt. Werden beispielsweise serverseitig in der Datei Password.txt Passwörter gespeichert, stellt dies ein erhebliches Sicherheitsrisiko dar, da diese Datei jedem zugänglich ist.

3.2 Kontrollablauf für die Maßnahmenüberprüfung

GUI: Der Benutzer veranlasst die Überprüfung der Maßnahme „mitgelieferte CGI-Programme vollständig entfernen“ für ein zu schützendes Objekt. Dazu ruft er über das GUI die Funktion „Kontrolle“ auf und ihm werden alle verfügbaren Typen zu schützender Objekte angezeigt. Nach der Auswahl eines Typs werden ihm alle in der Datenbank gespeicherten zu schützenden Objekte dieses Typs angezeigt. Nach Auswahl eines Objekts werden schließlich alle verfügbaren Maßnahmen, die zum assoziierten Typ gehören, angezeigt. Aus diesen wählt der Benutzer die gewünschte Maßnahme (entferne alle mitgelieferten CGI-Scripte) aus und initiiert durch die Funktion „Kontrolle ausführen“ deren Überprüfung. Die Kontrolle wird damit an das Servicemodul übergeben.

SERVICEMODUL: Das Servicemodul erstellt einen Kontrollauftrag im XML Format. Darin ist u. a. festgelegt, dass das Kontrollmodul Whisker die Kontrolle durchführen soll, die IP-Adresse des zu überprüfenden Objekts, welche Informationen zurückgeliefert werden und wann die Kontrolle durchzuführen ist. Der Kontrollauftrag besteht aus dem Eingabeteil (vgl. Abb. 3), der die Informationen enthält, die für die Durchführung der Kontrolle erforderlich sind (u.a. IP-Adresse oder DNS-Name, Kontrollauftragsnummer, Kontrollauftragspriorität, Kontrollmodul-ID) und einem Ausgabeteil mit Informationen, die für die Auswertung der Maßnahme erforderlich sind, wie Portnummer des geprüften Webservers und die URL des gefundene Skripts (vgl. Abbildung 3).

DISPATCHER: Der Auftrag wird an den Dispatcher übergeben, der das benötigte Kontrollmodul (Whisker) verwaltet. Anhand der erfragten Informationen stellt das Kontrollmodul die Parameter zusammen, und startet damit Whisker, das seine Kontrollen durchführt.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eSI:KontrollAuftrag PUBLIC "kontrollauftrag"
"/eSIDokumente/Kontrollmodule/KontrollAuftrag/KontrollAuftrag.dtd">
<eSI:KontrollAuftrag Priority="highest" KMNumber="1" ID="4711" DTDVersion="1">
        <eSI:Request>
                <eSI:Info Name="eSI:IP-Address" TYP="eSI:IP-Address">
                        <eSI:Value>192.168.59.82</eSI:Value>
                        <eSI:Info Name="eSI:Portnummer" TYP="eSI:Portnummer">
                                <eSI:Value>80</eSI:Value>
                                <eSI:Info Name="eSI:SkriptURL "
                        TYP="eSI:URL">
                                </eSI:Info>
```

Abbildung 3: Information zur Durchführung der Maßnahmenkontrolle

Aus den vom Tool gelieferten Daten, werden die relevanten Informationen extrahiert (kein „www.wiretrip.net“ etc.) und der um die Ergebnisse ergänzte Kontrollauftrag wird an den Kontrollmodul-Manager zurückgeschickt. Der Manager hat die Durchführung überwacht und meldet eventuell aufgetretene Fehler. Vom Auftragsmanager wird nun Beginn und Endzeitpunkt der Kontrolle eingetragen (für diesen Zeitraum sind die Informationen erwiesenermaßen gültig) und er sendet den mit Prüfergebnissen ausgestatteten Kontrollauftrag über den Dispatcher zurück an das Servicemodul. Für das vorliegende Beispiel enthält der ausgefüllte Kontrollauftrag die in Abbildung 4 angegebenen Informationen.

SERVICEMODUL: Um eine Aussage über den Umsetzungszustand der zu prüfenden Maßnahme treffen zu können, wird der Kontrollauftrag an die Auswertungslogik übergeben.

AUSWERTUNGSLOGIK: Hier werden Auswertungsregeln auf die Prüfergebnisse im Kontrollauftrag angewandt. Das Ergebnis „umgesetzt“ oder „nicht umgesetzt“ wird an das Servicemodul übergeben.

SERVICEMODUL: Das Ergebnis und der Kontrollauftrag werden persistent in der Datenbank gespeichert.

GUI: Der Benutzer kann das Kontrollergebnis in zusammengefasster oder detaillierter Form einsehen. Für bestimmte kritische Prüfergebnisse kann auch vorgegeben werden, ob etwa eine Mitteilung an einen Sicherheitsverantwortlichen direkt zu senden ist.

4 Zusammenfassung und Ausblick

In dem Beitrag wurde dargestellt, wie IT-Sicherheitsmaßnahmen vom eSI-Tool automatisiert auf ihre Umsetzung hin technisch überprüft werden können. Hierzu integriert das eSI-Tool unterschiedliche Prüfwerkzeuge und Verfahren. Die eSI-Funktionen sind über eine einheitliche Benutzungsschnittstelle verfügbar. Durch eine standardisierte Schnittstelle zu

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eSI:KontrollAuftrag PUBLIC "kontrollauftrag"
"/eSIDokumente/Kontrollmodule/KontrollAuftrag/KontrollAuftrag.dtd">
...
        <eSI:Response>
            <eSI:Info Name="eSI:IP-Address" TYP="eSI:IP-Address">
                <eSI:Value>192.168.59.82</eSI:Value>
                <eSI:Info Name="eSI:Portnummer" TYP="eSI:Portnummer">
                    <eSI:Value>80</eSI:Value>
                    <eSI:Info Name="eSI:SkriptURL " TYP="eSI:URL">
                    <eSI:Value>/cgi-bin/foo.pl</eSI:Value>
                    </eSI:Info>
                    <eSI:Info Name="eSI:SkriptURL " TYP="eSI:URL">
                    <eSI:Value>/cgi-bin/exploit.pl</eSI:Value>
                    </eSI:Info>
                </eSI:Info>
            </eSI:Info>
        </eSI:Response>
        <eSI:Log>
            <eSI:Begin>
                <eSI:Date>28/05/2003</eSI:Date>
                <eSI:Time>16:25</eSI:Time>
            </eSI:Begin>
            <eSI:End>
                <eSI:Date>28/05/2003</eSI:Date>
                <eSI:Time>16:30</eSI:Time>
            </eSI:End>
            <eSI:Error ErrorCode="0">
            </eSI:Error>
        </eSI:Log>
</eSI:KontrollAuftrag>
```

Abbildung 4: Ausgefüllter Kontrollauftrag

den Kontrollmodulen können weitere Prüfwerkzeuge einfach integriert werden. Die regelbasierte Auswertungskomponente stützt sich auf eine deklarative Wissensrepräsentation und ermöglicht so eine einfache Nachpflege bei Veränderungen des Sicherheitskonzepts.

Die eSI-Entwicklung ist Bestandteil des SKe Projekts[1], das im Rahmen des Programms *Leben und Arbeiten in einer vernetzten Welt* vom BMBF gefördert wird.

Literatur

[And01] Anderson, Ross: Security Engineering. John Wiley & Sons, Inc., 2001

[Ber01] Bergsten, Hans: JavaServer Pages. O'Reilly & Associates Inc. 2001

[Eck03] Eckert, Claudia: IT-Sicherheit. Oldenbourg, München, 2.Auflage, 2003

[EJB] Enterprise Java Beans Technology http://java.sun.com/products/ejb/index.html

[GSHB] IT-Grundschutzhandbuch des BSI, http://www.bsi.bund.de/gshb

[1] SKe: Durchgängige Sicherheitskonzeption mit dynamischen Kontrollmechanismen für eService-Prozesse, http://www.ske-projekt.de.

[Mü99] Münch, Isabel: Sicherheitsrevision – Praktische Erfahrungen des BSI und theoretische Ansätze. In: Tagungsband 6. Deutscher IT-Sicherheitskongress des BSI, 1999, S. 355

[Nmap] http://www.insecure.org/nmap/

[Whisk] Whisker Informationen: http://www.wiretrip.net/rfp/p/doc.asp?id=21&iface=2

[Zie01] Zieschang, Thilo: Security Engineering im E-Commerce: Best Practices und Standardsicherheitsmaßnahmen. In: Tagungsband 7. Deutscher IT-Sicherheitskongress des BSI, 2001, S. 211-225