Topologie-Editoren zur graphischen Konzeption von virtuellen Sicherheitsinfrastrukturen

Prof. Dr. Kai-Oliver Detken[1] · Prof. Dr. Evren Eren[2] ·
Falk Krämer[3] · Dr. Sascha Müller[4]

[1] DECOIT GmbH, Fahrenheitstraße 9, D-28359 Bremen
detken@decoit.de

[2] Fachhochschule Dortmund – Fachbereich Informatik,
Emil-Figge-Str. 42, D-44227 Dortmund
evren.eren@fh-dortmund.de

[3] Collax GmbH, Basler Str. 115a, D-79115 Freiburg
falk.kraemer@collax.com

[4] IT-Security@Work GmbH, Robert-Bosch-Straße 18,
D-63303 Dreieich-Sprendlingen
sascha.mueller@isw-online.de

Zusammenfassung

Der Betrieb komplexer IT-Infrastrukturen bedingt den flexiblen Entwurf und das Testen von neuen Strukturen und Topologien. Im diesem Kontext nimmt die Bedeutung von Server- und Netzwerkvirtualisierung immer mehr zu. IT-Infrastruktur können in virtualisierter Form viel einfacher und schneller abgebildet oder bestehende physikalische IT-Infrastrukturen in eine virtualisierte Topologien übernommen und einer Analyse unterzogen werden. Doch, obwohl das Angebot an Virtualisierungssoftware sowie deren Fähigkeiten stetig zunimmt, sind die Möglichkeiten der Konzeptionierung und Steuerung von virtuellen Netzwerktopologien noch immer beschränkt und es existieren kaum brauchbare Lösungen. Die Maßnahmen zur Steuerung kompletter virtueller Netzwerke basieren in der Regel auf Konsolenbefehlen oder Verbindungstabellen. Derartige Steuerungsmethoden erfordern bei der Anwendung einen hohen Abstraktionsgrad und sind dadurch entsprechend fehleranfällig. Graphische Tools zum Entwurf und Umsetzung von Netztopologien, die das Spektrum Layer 1 (Kabel) bis Layer 3 (Switches, Router) mitsamt Virtual Machines (VM) abdecken, sind für Anwender im KMU-Umfeld nicht zu finden oder zu komplex in der Bedienung. Vor diesem Hintergrund wurden im BMBF-Projekt VISA sog. Topologie-Editoren entwickelt, indem eine Simulationsumgebung für teilautomatisierte Tests auf Basis einer erhobenen oder erstellten Netztopologie realisiert werden kann. Konfigurationen können so analysiert, getestet, verbessert und optimiert werden, um dann in die physikalische Umgebung zurückgeführt werden zu können. Das Ziel war es dabei, das Sicherheitsniveau in Unternehmen signifikant zu erhöhen. Zudem lässt sich die Compliance eines Netzes effektiv testen bzw. sicherstellen.

P. Schartner · P. Trommler (Hrsg.) · DACH Security 2013 · syssec (2013) pp-pp.

1 Konzeption und Steuerung von Netztopologien

Durch die starke Heterogenität von IT-Infrastrukturen, der relativ begrenzten Ressourcen sowie des relativ geringen technischen IT-Know-hows müssen in Zukunft Klein- und Mittelständische Unternehmen (KMU) bessere und geeignete Methoden zur flexiblen Konfektionierung, Erprobung und Optimierung ihrer IT-Infrastrukturen an die Hand bekommen. Dies ist insbesondere für die IT-Sicherheit wichtig. Um eine höhere Autonomie in der Konfiguration sowie im Betrieb ihrer IT-Infrastruktur zu erhalten, sind modulare und erprobte Lösungen und Systeme essentiell. Dies wurde im VISA-Projekt (www.visa-project.de) durch die Entwicklung von sog. Topologie-Editoren sowie Methoden und Tools zur Netzmodellierung/-simulation adressiert.

Virtualisierungslösungen haben sich inzwischen zu einer unverzichtbaren Methode in IT-Infrastrukturen entwickelt. Eine Vielzahl von Lösungen zur Administration und Monitoring existieren seit einigen Jahren am Markt. Diese unterscheiden sich jedoch in Bezug auf Funktionen, Zielgruppen, Komplexität und Umfang wesentlich. Viele Hersteller von Virtualisierungslösungen stellen darüber hinaus eigene Tools und inhärente Funktionen bereit.

Eine Möglichkeit zur Netzwerkkonfiguration ist die Erstellung virtueller Interfaces, mit denen man die Virtual Machines (VM) über Bridges verbinden kann. Bei diesen handelt es sich um eine direkte Weiterleitung des Datenverkehrs von einem Netzwerkinterface zu einem anderen. Diese Art der Verbindung erfolgt über eine Konfiguration der Netzwerkadapter des Betriebssystems und ist weit verbreitet. Mit Bridges und virtuellen Netzwerkadaptern ist es somit möglich, ein beliebig großes Sternnetzwerk aufzubauen. Dies spiegelt jedoch nur bedingt die Realität wieder, da beispielsweise keine Switches mit entsprechender Funktionalität abgebildet werden können. Auch lassen sich durch diese Art der Netzwerkvirtualisierung nur perfekte (unmittelbare, verlust- und fehlerfreie) Verbindungen erstellen. Da virtuelle Netzwerke auch zu Testzwecken benutzt werden, stellt dies eine Einschränkung der Funktionalität dar.

Graphische Tools zum Entwurf und Umsetzung von Netztopologien, die das Spektrum Layer 1 (Kabel) bis Layer 3 (Switches, Router) mitsamt VM abdecken sind für Anwender im KMU-Umfeld nicht zu finden oder zu komplex in der Bedienung. Es ist festzustellen, dass derzeit sehr wenige Lösungen zum Erstellen und Steuern von virtuellen Netztopologien existieren. Hierzu zählt beispielsweise GNS3, GPL (www.gns3.net). Diese Software ist ein grafischer Netzwerksimulator, der auf dem Router-Emulator Dynamips[1] und dessen Front-end Dynagen (www.dynagen.org) aufbaut. Dynagen ermöglicht, komplexe Router-Netzwerke mit Hilfe einer einfachen Konfigurationsdatei zu erstellen. GNS3 ist als eine GUI für Dynagen zu verstehen. Durch die grafische Unterstützung lassen sich mit GNS3, auch ohne Kenntnisse der Dynagen-Syntax, sehr komfortabel komplexe virtuelle Netzwerke erstellen.

Aufgrund des Mangels an Lösungen und Produkten wurden im Forschungsprojekt VISA mit unterschiedlichen Ansätzen drei Topologie-Editoren entwickelt: Virtual Wizard (VirWi) von der FH Dortmund, Topologie-Editor (TE) von der DECOIT GmbH und Spotlight von der Collax GmbH. Diese unterschiedlichen Topologie-Editoren, die entwickelt wurden, um eine flexiblere Möglichkeit zur graphischen Konzeption von virtuellen IT-Sicherheitsinfrastrukturen zu schaffen, werden im Folgenden genauer vorgestellt und miteinander kurz verglichen.

[1] http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator

2 Virtual Wizard (VirWi)

Der Topologie-Editor VirWi der FH Dortmund (www.fh-dortmund.de) ist eine Client-Server-basierte Software, die der Darstellung, Verwaltung und Steuerung einer vollständigen virtuellen Netzwerkumgebung dient. Die eigentliche Virtualisierung und Steuerung wird hierbei von KVM (Kernel-based Virtual Machine)[2] und VDE (Virtual Distributed Ethernet)[3] geleistet. KVM ist eine Open-Source-basierte Virtualisierungslösung für Linux, welche Vollvirtualisierung auf x86-Hardware ermöglicht. Hierbei werden die Befehlssatzerweiterungen der modernen Prozessoren Intel VT und AMD-V genutzt. Das Verfahren der Vollvirtualisierung ermöglicht es, mehrere unveränderte Linux- und Windows-Betriebssysteme parallel auf einem System zu betreiben. KVM baut auf dem Emulator QEMU auf. QEMU kann verschiedene Prozessorarchitekturen emulieren, hierzu zählen PowerPC, ARM, Alpha, m68k, MIPS und Sparc. Er stellt weiterhin die virtuelle Hardware (z.B. virtuelle Netzwerkinterfaces) den VMs bereit. VDE stellt eine allgemeine, virtuelle Infrastruktur zur Verbindung verschiedener Softwarekomponenten zur Verfügung. Es lassen sich VMs verschiedener Virtualisierungslösungen, Emulatoren, reale Betriebssysteme und Netzwerke miteinander verbinden. Auf Basis von VDE lassen sich sehr einfach und flexibel virtuelle Netzwerke erstellen. Teilbereiche solcher virtuellen Netzwerke lassen auf mehrere physikalische Rechner verteilen. VDE ist Ethernet-konform und stellt virtuellen Infrastrukturen virtuelle Switche und virtuelle Kabel zur Verfügung.

Mit VirWi lassen sich sowohl einzelne Virtuelle Maschinen (VM) als auch ganze Netzwerktopologien konfigurieren. Die Planung (das Editieren) des Netzwerkes erfolgt über eine graphische Oberfläche (WYSIWYG). Die Netzwerkpläne werden dann von der Software auf einem KVM-Server direkt umgesetzt.

Die meisten der geplanten Leistungsmerkmale sowie Anforderungen an den VirWi-Editor konnten umgesetzt werden. Dies sind u.a.:

- **Verwalten von virtuellen Festplatten:** Virtuelle Festplatten können erfasst und verwaltet werden. Beliebige Hardware kann zu einer VM hinzugefügt und entfernt werden.
- **Erstellen von VMs:** VMs können erstellt und konfiguriert und verwaltet werden. Die jeweilige Hardware der VM kann durch einfache Menüs freigestaltet werden.
- **Remotezugriff:** Die gesamte Steuerungs-GUI wird nicht auf dem KVM-Server selbst, sondern per Remotezugriff auf einem beliebigen Client per VNC ausgeführt.
- **Speicherung:** Einzelne VMs wie auch Netzwerkpläne können abgespeichert werden.
- **Verwalten von Bridges:** Bridges in der Netzwerkkonfiguration des Server-Hosts können verwaltet und diese mit echten Interfaces verlinkt werden.
- **Verwalten von Netzwerken:** Das Einsehen, Verwalten, Löschen, Laden und Speichern von Netzen ist möglich. VMs und Switches können im Betrieb an-/abgeschaltet werden.
- **Migration:** Netzwerkpläne können auf mehrere Server verteilt werden. Die Kommunikation zwischen den Servern erfolgt verschlüsselt.
- **Layout:** Farben und Formen der Oberflächengestaltung kann man durch Stylesheets ändern. Die Icons der VMs lassen sich frei wählen, im Gegensatz zum Steuerungslayout.

---

[2] http://www.linux-kvm.org/page/

[3] http://wiki.virtualsquare.org/index.php/Introduction#Virtual_Square_Networking

2.1 Umgebung und Systemanforderungen

Die Softwareumgebung besteht aus einer Client- und einer Server-Software. Beide bestehen aus Java-Code und benötigen die JRE von Sun Microsystems bzw. Oracle. Das Programm ist damit plattformunabhängig. Der Server benötigt VDE, selbst kompiliertes KVM mit VDE-Support sowie Java mit SSL-Support. Zur Steuerung der Switches und VMs im laufenden Betrieb sollte auf dem Server ein SSH-Dämon laufen und Programme wie unixterm, nc und ssh zur Verfügung stehen. Darüber hinaus sollten die TAP-Interfaces Zugriffsrechte für den User haben, unter dem das Server-Programm läuft.

Als Hardware-Umgebung reicht ein handelsüblicher PC. Je nach Größe der zu erstellenden Netzwerke ist für die Serverkomponente ein leistungsstarkes Serversystem mit entsprechender Netzwerkanbindung, Prozessorkernen und Speicher vonnöten.

2.2 Zentrale Komponenten und Elemente

Im VM-Manager werden alle virtuellen Maschinen (VM), die in der Datenbank gespeichert sind, verwaltet. Jedes Netzwerk erhält beim Start automatisch alle TAP-Interfaces des Betriebssystems. Wird ein Switch an ein TAP-Interface angeschlossen und ist dieses auf dem Server korrekt mit dem echten Interface „gebridged“, so ist es möglich, das echte Netzwerk des Servers zu erreichen. Durch Nichtverbinden der TAP-Interfaces erstellt man ein isoliertes Netzwerk.

Kabelverbindungen lassen sich einfach erstellen. Besitzt ein Element mehr als ein Interface, erscheint ein Popup-Menü, in dem man die Interface-Nummer, an der das Kabel angeschlossen werden soll, auswählen kann. Jede Verbindung muss mindestens einen Switch enthalten, d.h., erlaubte Verbindungen sind: Switch-Switch, Switch-VM, Switch-Complex und Switch-TAP.

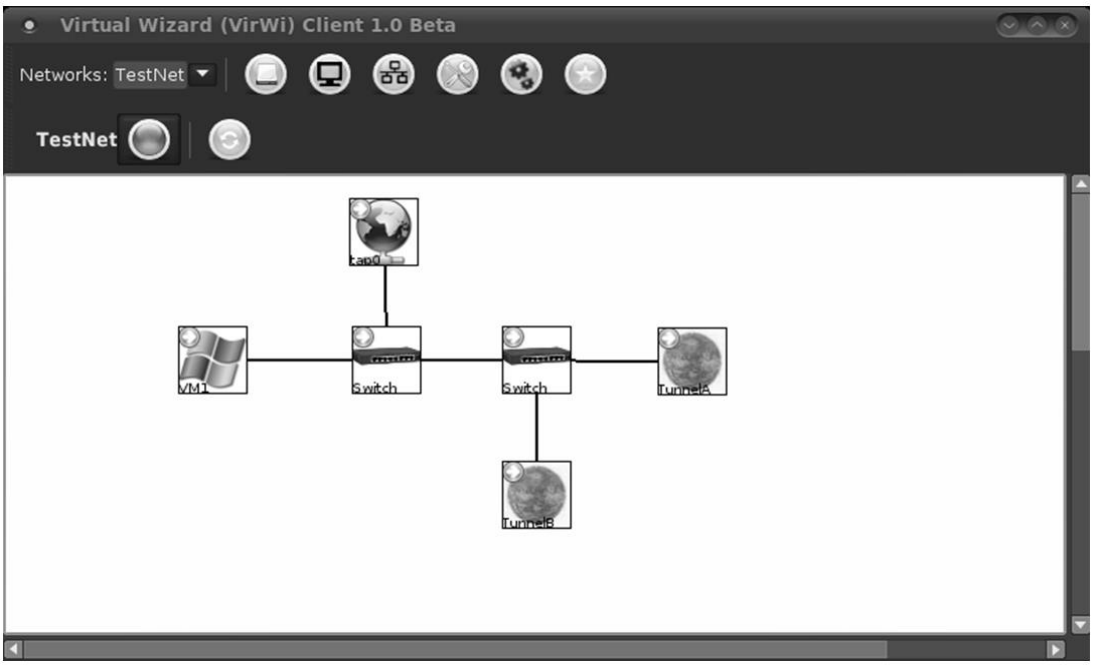

Abb. 1: VirWi-GUI

Die sog. Complex Elements (CE) stellen eine weitere Option dar. Sie werden wie Switches in das Netzwerk eingefügt. Jedoch müssen bei der Erstellung ein Name und ein Befehl angegeben werden. Bei dem Befehl handelt es sich um einen beliebigen Prozess, der auf dem Server ausgeführt wird. Der Prozess wird beim Netzwerkstart mit gestartet und mit ihm zusammen wieder beendet. Der „Std Input Stream“ und „Std Output Stream“ des Prozesses wird durch die Kabelverbindung hergestellt.

Ein Beispiel für ein sinnvolles CE ist ein Netzwerktunnel. In Netzwerk A wird ein CE mit dem Befehl „nc -lc localhost 10010“ erstellt und durch einen Switch verbunden. In Netzwerk B wird ein CE mit dem Befehl „nc -c localhost 10010“ erstellt und ebenfalls verbunden. Beim Starten von A und anschließend B wird mithilfe des „nc“ Befehls automatisch ein Tunnel zwischen diesen Netzwerken aufgebaut. Somit ist es möglich, zwei Netzwerke auf zwei unterschiedlichen Servern zu verbinden. In diesem Beispiel muss (aufgrund der Funktionalität des „nc“-Befehls) Netzwerk A stets vor Netzwerk B gestartet werden. Anstatt „nc“ kann z.B. auch „SSH“ benutzt werden, um einen verschlüsselten Tunnel aufzubauen. Da VDE-Switches jeden beliebigen Prozess akzeptieren, sind den Anwendungsmöglichkeiten von CEs keine Grenzen gesetzt.

Ein Bearbeiten des Netzwerkes im laufenden Betrieb ist nicht möglich. Jedoch kann durch Klicken auf einen Switch oder eine VM im Online-Modus ein Verwaltungsmenü aufgerufen werden. Es ist somit möglich, einzelne VMs zu starten oder zu stoppen, die Log-Datei der VM einzusehen und per VNC auf diese zuzugreifen. Hierzu sollten die Angaben zum externen VNC-Client unter „Client Konfiguration“ überprüft werden. Wenn beim Login eine SSH-Verbindung angegeben wurde, ist es auch möglich, sich direkt mit dem Monitor der VM zu verbinden.

Bei Switches kann ebenfalls die Log-Datei eingesehen und bei vorhandener SSH-Verbindung auf den Kontrollsocket zugegriffen werden. Mithilfe des nun roten Online/Offline Buttons kann das gesamte Netzwerk wieder heruntergefahren werden. Nach dem Herunterfahren wechselt die Benutzeroberfläche wieder in den Offline-Modus. Um Änderungen an einer VM im laufenden Betrieb vorzunehmen existiert eine sog. Monitor-Console in „qemu“.

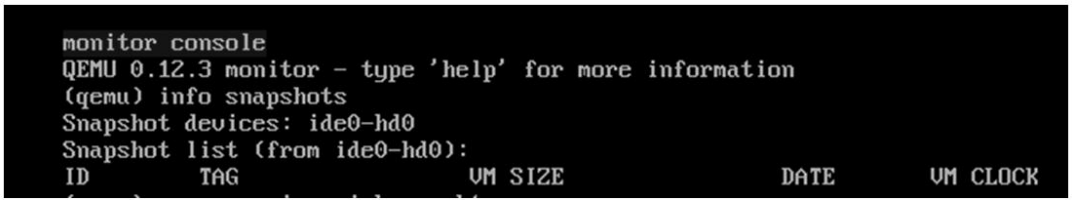

Abb. 2: Monitor-Konsole des QEMU

2.3 Probleme und zukünftige Entwicklungsarbeiten

Eine Anforderung war technisch nicht lösbar, was zum Planungszeitpunkt nicht bekannt war. So lassen sich Kabelverbindungen im laufenden Betrieb beliebig nicht ändern, was an der fehlenden Unterstützung durch die Virtualisierungssoftware liegt. Das native Monitorprotokoll lässt sich aber in wenigen Schritten so erweitern, dass neue Funktionen in die GUI eingebettet werden können. Des Weiteren sind erweiterte Online-Status-Abfragen, z.B. Prozessor- und RAM-Verbrauch oder Mehrsprachigkeit, technisch problemlos umsetzbar und in Planung. Die technischen Voraussetzungen für deren Umsetzung sind auf jeden Fall vorhanden.

3 VISA Topologie Editor (VTE)

Der VISA Topologie Editor (VTE) der DECOIT GmbH (www.decoit.de) bietet dem Benutzer die Möglichkeit, eine bereits bestehende und vorher erhobene Topologie zu bearbeiten sowie neue Komponenten hinzuzufügen. Weiterhin kann auch eine neue bzw. bestehende Topologie von Hand nachmodelliert werden. Der VTE besteht dabei aus zwei Kernkomponenten:

- Back-end: ein in Java geschriebener Serverdienst
- Front-end: eine Web-basierte grafische Oberfläche

Bereits erhobene IT-Infrastrukturen können entweder durch eine RDF-/XML-Datei oder durch eine TLS-gesicherte TCP-Verbindung in den VTE importiert werden. Ebenso besteht die Möglichkeit, selbst erstellte oder veränderte IT-Infrastrukturen zu sichern. Zum Erheben einer IT-Infrastruktur ist das Interconnected-asset Ontology (IO) Tool von Fraunhofer SIT (www.sit.fraunhofer.de) notwendig, dass ebenfalls im Forschungsprojekt VISA entwickelt wurde. Das sog. IO-Tool-Set ist eine Software, die es ermöglicht, den Ist-Zustand eines Netzes, d.h., die Konfiguration von Komponenten und die Verbindungen der Komponenten untereinander, zu erheben. Diese Daten werden in ein formales Datenmodell überführt und mit Hilfe dieser Daten Anfragen zum Zustand des Netzes beantwortet. Kern des Systems ist eine Ontologie, die zum einen das formale Datenmodell darstellt, aber auch gleichzeitig die Daten dieses Datenmodells enthält. Im Rahmen des VISA-Projekts stellt das IO-Tool-Set Methoden bereit, die dem VTE wiederum Topologien bereitstellen. Dadurch lassen sich Änderungen speichern, vorhandene Quellumgebungen erheben oder erhobene Umgebungen replizieren. Eine ausführlichere Darstellung des IO-Tool-Sets ist in dem zweiten Beitrag „Design und Implementierung von Virtual Security Appliances (VSA)“ zu finden.

3.1 Back-end

Wie in Abb. 3 zu sehen ist, besteht das Back-end aus drei Modulen: Topologie-, RDF- und HTTP-Modul. Das Topologie-Modul stellt die Topologie, die entweder über das Webinterface oder aus RDF/XML-Daten importiert wurde, mit Hilfe von Java-Klassen dar.

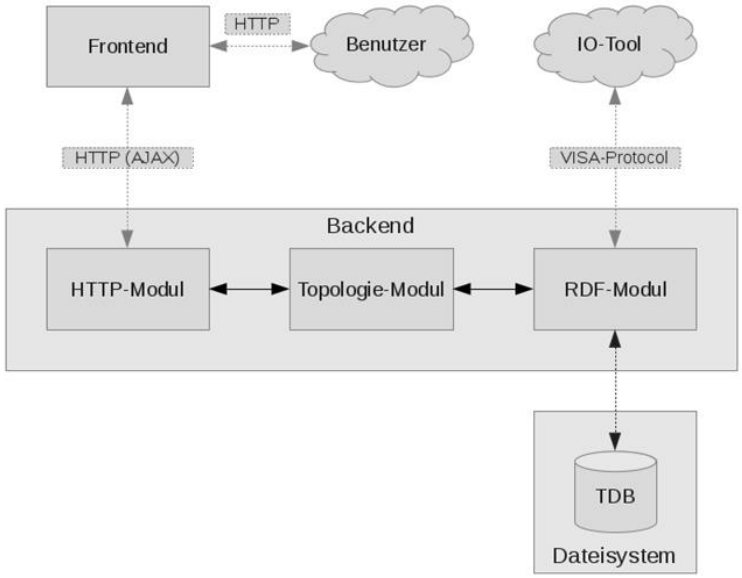

Abb. 3: Architektur des VISA Topologie Editors (VTE)

Das zentrale Element dieses Moduls ist die abstrakte Klasse „NetworkComponent“, die als Basisklasse aller Komponenten in der Topologie dient. Als Komponente werden alle physikalischen und virtuellen Geräte bezeichnet, also z.B. Switches, Router, Firewalls und Server, die wiederum in drei Unterklassen sich einteilen lassen. Diese grobe Aufteilung ist notwendig, da das IO-Tool vom Fraunhofer SIT, welches die Topologie erhebt, nicht in der Lage ist feinere Unterscheidungen zu machen. Neben den Komponenten gibt es noch Klassen für Netzwerkschnittstellen und Kabel. Die Klasse für Netzwerkschnittstellen ist als innere Klasse von „NetworkComponent“ implementiert und heißt „Interface“. Sie enthält Eigenschaften, die für jedes Interface unterschiedlich sind. Zum Beispiel die Konfiguration für das Internet Protocol (IP), also Adresse, Subnetzmaske, IP Version und Netzwerkadresse.

Abb. 4 zeigt die Struktur, die im Topologie-Modul erzeugt wird. Jedes Objekt der Klasse „NetworkComponent“ besitzt ein oder mehrere Objekte von „Interface“, die die Schnittstellen dieser Komponente darstellen. Die Interface-Objekte werden wiederum mit Objekten der Klasse „NetworkCable“ verknüpft, womit die Komponenten, zu denen die beiden Ports gehören, miteinander verbunden sind. In dieser Topologie könnte also „NetworkComponent 1“ ein Switch sein, der die Geräte „NetworkComponent 2“ und „NetworkComponent 3“ miteinander verbindet.

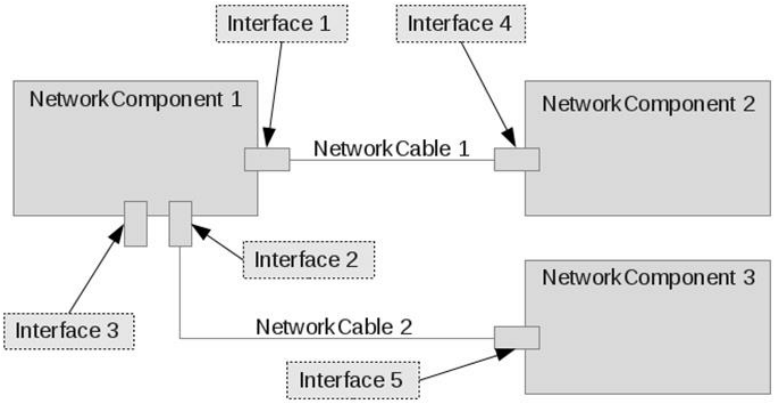

Abb. 4: Schematische Darstellung der im Topologie-Modul erzeugten Objektstruktur

Das RDF-Modul sorgt für die Verwaltung der RDF-Informationen. Dazu baut es auf dem Open Source Framework Jena[4] auf, das von der Apache Foundation entwickelt wird. Die zentrale Klasse des Moduls ist der RDF-Manager. Wird ein Objekt dieser Klasse erzeugt, wird ein sogenanntes „Dataset“ erstellt, welches die RDF-Informationen speichert und verwaltet. Ein Dataset enthält mindestens ein RDF-Modell, das „Default Model“. Neben diesem können noch beliebig viele „Named Models“ in einem Dataset enthalten sein. Das im RDF-Manager verwendete Dataset verwendet das Datenbanksystem TDB als Speichermedium, welches von Jena mitgeliefert wird. Dieses erlaubt die Verwendung von Transaktionen für jeglichen Zugriff auf die gespeicherten Modelle, so dass im Falle eines Fehlers bei der Verarbeitung kein dauerhafter Schaden an den Daten entstehen kann.

Das dritte Hauptmodul ist das HTTP-Modul. Dieses stellt einen einfachen HTTP-Server zur Verfügung, der die vom Front-end abgesetzten AJAX-Requests verarbeiten und beantworten kann. Die zentrale Klasse dieses Moduls ist der AJAX-Server. Der AJAX-Server startet eine

[4] The Apache Software Foundation: Apache Jena Framework. http://jena.apache.org/index.html, 2011-2012

Instanz dieses einfachen HTTP-Servers und definiert die HTTP-Handler, die die einzelnen AJAX-Requests bearbeiten.

3.2 Front-end

Das Front-end des VTE wurde als Web-Oberfläche, größtenteils in JavaScript, entwickelt. Im oberen Bereich der Oberfläche (Abb. 5) unterhalb der Kopfzeile, befindet sich die Optionsleiste. Sie erlaubt verschiedene Einstellungen und gibt Zugriff auf diverse Funktionen des Editors. So lässt sich hier die automatische Wegfindung für Kabel auf dem Editor-Raster ein- und ausschalten. Die zweite Option erlaubt das Einblenden der vollen Namen aller Komponenten auf dem Raster und die dritte das Einfärben der Kabel entsprechend des VLANs zu dem sie gehören. Diese beiden Optionen können zeitgleich aktiviert werden und blockieren jeweils die Drag-and-Drop-Funktionalität des Editors um Anzeigeprobleme zu verhindern.

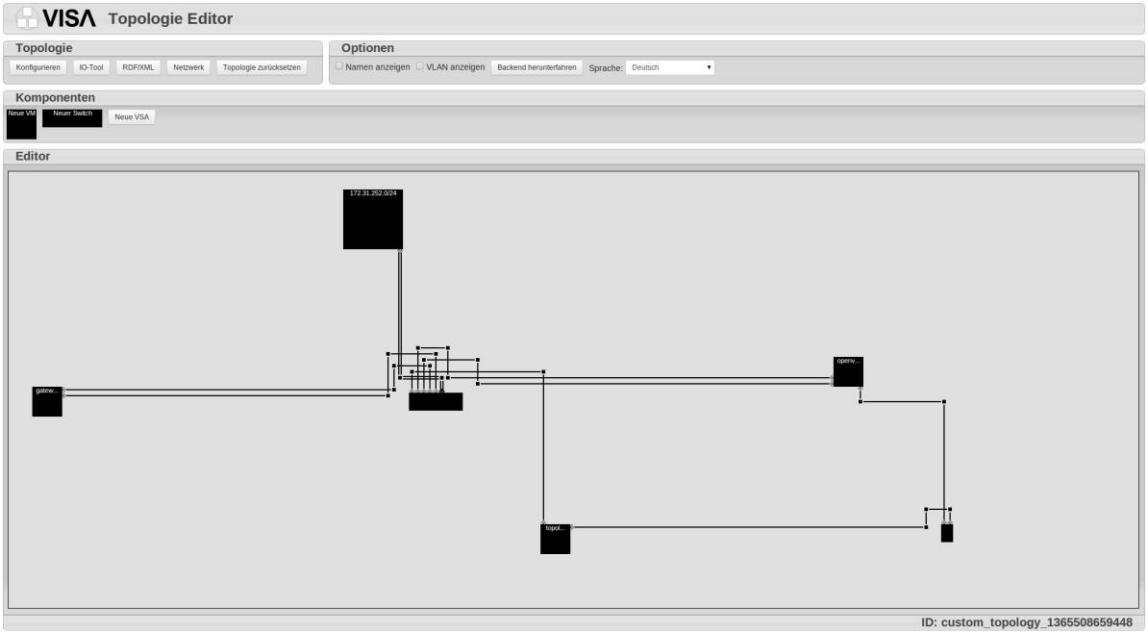

Abb. 5: Startseite des VISA Topologie Editors (VTE)

Aktive Komponenten werden als schwarze Boxen dargestellt. Beim Hinzufügen neuer Komponenten stehen dem Benutzer folgende Optionen zur Verfügung: der Name, die Breite/Höhe, die Anzahl der Netzwerkschnittstellen und die Position der Schnittstellen an der dargestellten Box.

Die Möglichkeit mehrere Komponenten zu einer Gruppe zusammenzufassen ist eine der zentralen Funktionen des VTE. Da das Raster, auf dem die Komponenten abgebildet werden, relativ klein ist, reicht der Platz höchstens für ca. 20 Geräte. Danach wird die Darstellung unübersichtlich oder ist überhaupt nicht mehr möglich, da sich die Komponenten nicht mehr überlappungsfrei platzieren lassen. Um trotzdem größere Topologien darstellen zu können, werden Komponenten, die bestimmte Eigenschaften teilen, zusammengefasst. Die Gruppen werden im Editor als Objekte dargestellt und nehmen dadurch relativ wenig Platz ein. Ein Klick auf die Gruppe öffnet diese in einem weiteren Raster und erlaubt so die Einsicht der Inhalte dieser. Zurzeit können die Komponenten nur automatisch vom Back-end in Gruppen aufgeteilt werden. Eine manuelle Konfiguration der Gruppen wird eventuell zu einem späteren Zeitpunkt noch stattfinden.

4 Spotlight

Spotlight ist der Topologie-Editor, der innerhalb des VISA-Projektes von der Collax GmbH (www.collax.de) entwickelt wurde und daher genau auf die Collax-Produkte abgestimmt ist. Er soll Administratoren das zentralisierte Verwalten von verteilten Server-Umgebungen erlauben und die Evaluation im VISA-Projekt unterstützen, indem bereitgestellte Werkzeuge im TE mit den eingesetzten Virtual Security Appliances (VSA) kommunizieren. VSAs sind Sicherheitskomponenten, die aus einer oder mehreren VMs bestehen können und auf virtueller Basis einen Sicherheitsdienst anbieten. Konfigurationen der VSAs sind im TE editierbar und VSA-Informationen können abgerufen werden. Die verwalteten VSAs werden in einer Datenbank hinterlegt und lassen sich über eine GUI hierarchisch strukturieren. Für den Datentransfer zwischen Spotlight und den VSAs sind auf den VSAs entsprechende Agenten notwendig.

4.1 Spotlight-Server

Realisiert wurde Spotlight als eigenständiges Modul für die Collax-Server-Plattform. Neben dem nicht-virtualisierten Betrieb auf dedizierter Hardware wurde Spotlight auch für den Betrieb als VSA vorbereitet. Dazu wurde die Unterstützung der Paravirtualisierung integriert, die für den virtuellen Betrieb modifizierte Konfiguration hinterlegt und das automatische Deployment als VM, basierend auf dem Template, umgesetzt. Die so bereitgestellte VSA wird über das Management-Interface als virtuelle Instanz in Betrieb genommen.

Die grafische Administrationsoberfläche von Spotlight wurde, um plattformunabhängig als Web-Anwendung einsetzbar zu sein, mit Hilfe des Ajax-Frameworks Qooxdoo (www.qooxdoo.org) implementiert. Qooxdoo ist eine unter der L-GPL lizenzierte Programmbibliothek für die Implementierung von Web-basierenden Applikationen, die über einen Web-Browser gesteuert werden können. Es stellt zahlreiche in JavaScript vorliegende Komponenten zur Verfügung, die das Look-and-Feel einer klassischen Desktop-Anwendung vermitteln. Gemäß dem Ajax-Grundgedanken kommuniziert Qooxdoo zwischen dem Server und einem Browser über eine asynchron geführte Datenverbindung. Dies erlaubt es, Änderungen in der Darstellung der Anwendung durchzuführen, ohne die Seite komplett neu zu laden.

Die Darstellung der Topologie eines Netzwerks ist in zwei Bereiche aufgeteilt: in einem werden die verwalteten Server in einer Baumstruktur dargestellt, im zweiten kann man die Detailansicht eines ausgewählten Servers oder eine Übersicht mehrerer Server sehen (siehe Abb. 6). Betrifft die vom Administrator durchzuführende Aufgabe nur einen Teil der Server, kann man die hierarchisch angeordnete Server-Liste mit Hilfe von Filtern beschränken. Dabei kann die Auswahl auf bestimmte Ebenen der Hierarchie reduziert werden. Es können aber auch beliebige Attribute (z.B. CPU-Auslastung, Update-Stand) der Server für die Filterung herangezogen werden. Beide Filtermechanismen lassen sich auch miteinander kombinieren.

Auf einem Server aufgetretene Probleme und Warnungen werden gewichtet und dem Administrator in Form eines Statuslämpchens farbcodiert als konsolidierte Information angezeigt. Spotlight verfügt je Server über drei Ebenen der Informationsaufbereitung. Die oberste Ebene fasst die wichtigsten Informationen zusammen, die unterste gibt Auskunft über jedes Detail. Auf jeder Ebene können die Informationen ausgewählter Server gegenübergestellt werden. In der Detailansicht besteht die Möglichkeit, Auswertungen, Statistiken und Statusinformationen abzurufen. Außerdem stehen Funktionen zur Verfügung, um Kommentare zu einem Server zu hinterlegen und Kommandos remote auszuführen.

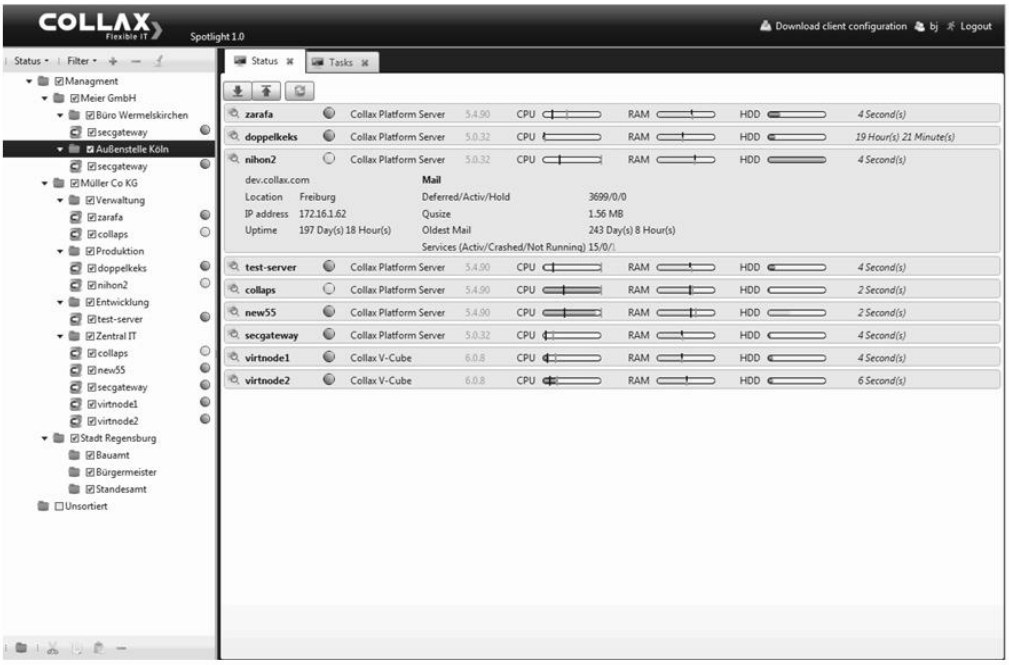

Abb. 6: Übersichtsdarstellung ausgewählter Server

Für die Speicherung und Verwaltung aller Daten der Remote-Server wird die relationale, GPL-lizenzierte MySQL-Datenbank verwendet. Hierzu wurde die bestehende Implementierung mit der Datenbank-Engine InnoDB integriert. Werden Daten von einem Agenten eines Remote-Servers empfangen, werden sie mit Referenz zum entsprechenden Server als Tabellen abgelegt. Ebenfalls werden die Tasks und die Kommentare in der Datenbank gespeichert. Offene Tasks werden bei der nächsten Verbindung zum entsprechenden Agenten übermittelt.

Die Verbindung zwischen Spotlight und den Agenten wird von Letzteren initiiert und zertifikatsbasierend, auf Basis von AES256-SSL, verschlüsselt. Hierzu wird auf die Open-Source-lizenzierte Bibliothek openSSL zurückgegriffen. Die Schnittstelle für die Kommunikation ist als Netzwerk-Dämon realisiert. Der in „C“ programmierte Dämon überwacht auf der Protokollebene TCP/IP den Port 443 für https. Die Verbindungen werden durch ein Modul für den Apache Web-Server authentisiert. Das Modul ist ebenfalls in „C“ geschrieben und wird beim Start von Apache geladen. Die Authentisierung der Verbindung wird anhand eines X.509-Zertifikats durchgeführt.

4.2 Spotlight-Agent

Für die Kommunikation zwischen Spotlight und den verwalteten Servern wurden Agenten implementiert. Für die Erhebung und Bereitstellung der Daten aus verschiedenen Quellen des Remote-Servers ist ein Plug-In-Framework für die sog. Kollektoren notwendig. Die Kollektoren werden von dem Agenten aufgerufen und übergeben diesem ihre Daten. Der Agent steht mit Spotlight in Verbindung und überträgt die gesammelten Daten. Er stellt eine Laufzeitumgebung für Plug-Ins (Kollektoren) bereit und definiert die Schnittstelle für den Austausch der Daten. Kollektoren haben zwei obligatorische und eine optionale Komponente:

- **Konfigurationsdatei:** Die Konfigurationsdatei enthält eine Beschreibung der Daten und Informationen, die der Kollektor bereitstellen wird.

- **Executable:** Die ausführbare Datei, die die Datenerhebung durchführt und die Daten zur weiteren Verarbeitung bereitlegt.
- **Kommandos (optional):** Über zusätzliche Kommandos können notwendige Aktionen ausgelöst werden, die für die Erhebung notwendig sind

Für die Ablage dieser Dateien im Dateisystem werden definierte Pfade verwendet. Der Agent überwacht kontinuierlich diese Verzeichnisse auf Änderungen. Wird eine Datei modifiziert, wird sie durch das Laufzeit-Framework vom Agenten neu eingelesen. Der Agent ruft das „Executable“ auf, damit die Kollektor-spezifischen Daten des Remote-Servers erhoben werden können. Auf der Standardausgabe „stdout“ werden die Daten an den Agenten übertragen. Durch die Spezifikationen aus der Konfigurationsdatei werden die gesammelten Daten eindeutig beschrieben und für die weitere Verarbeitung nutzbar gemacht.

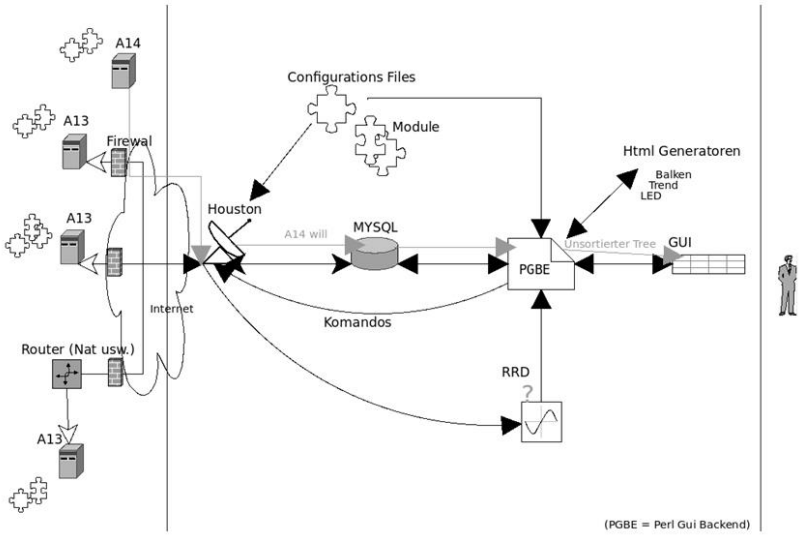

Abb. 7: Erste Kontaktaufnahme von einem Agenten zum Spotlight-Server

Über Konfigurationsoption des Remote-Servers wird der Agent aktiviert. Dabei ist konfigurierbar, welche Informationen an Spotlight weitergegeben und welche Befehle von Spotlight ausgelöst werden dürfen. Nimmt ein Agent zum ersten Mal die Verbindung zu Spotlight auf (Abb. 7), wird er in der Datenbank als unbekannter Server abgelegt. Der Administrator hat nun die Möglichkeit ihn in die bestehende Struktur einzugliedern. Der Agent stellt in regelmäßigen Intervallen (5 min) die Verbindungen erneut her, um auf die Instruktionen von Spotlight zu antworten. Spotlight selbst kann keine Verbindungen initiieren.

5 Umsetzung der Compliance

Die im VISA-Projekt entwickelten Topologie-Editoren zur Verwaltung von VSAs zielen auf einen Einsatz im Unternehmensumfeld ab. In einem solchen muss sich die gesamte VISA-Infrastruktur in die bestehende IT einfügen und den Anforderungen entsprechen, die an diese gestellt werden. Hier sind besonders IT-Sicherheitsstandards von Bedeutung, mit denen einerseits ein systematischer Umgang mit der IT-Sicherheit erreicht werden soll und andererseits aus Sicht des Managements eine gewisse Risikotransparenz entsteht. Die meist genutzten Sicherheitsstandards sind ISO 27001 und der BSI-Grundschutz, weshalb das VISA-Projekt auch diese hauptsächlich einbezogen hatte.

Im Zusammenspiel mit Sicherheitsstandards sind für VISA folgende Aspekte zu beachten:

- Die VSAs können helfen, bestehende IT-Risiken zu vermindern und so allein durch ihren Einsatz die Compliance eines Unternehmens verbessern.
- Die Topologie-Editoren selbst unterstützen bei der Umsetzung der Standards.
- Umgekehrt müssen die VSAs selbst Sicherheitsanforderungen erfüllen und es ergeben sich neue Compliance-Anforderungen, die von der IT erfüllt werden müssen, sowohl in technischer als auch in organisatorischer Hinsicht.

5.1 VSAs für Compliance

Eine wichtige Anwendung der mit den Topologie-Editoren verwalteten VSAs ist es, zu möglichen, dass mit geringem Aufwand Sicherheitsmaßnahmen in einer bestehenden IT-Infrastruktur umgesetzt werden können. Die entwickelten VSAs bieten daher gezielt Funktionalitäten, mit denen die Compliance einer IT verbessert werden kann. So kann, die im Projekt entwickelte VSA „E-Mail-Proxy“, die den E-Mail-Verkehr mit Hilfe etablierter Virenfilter gegen Malware (wie Viren, Trojaner, Würmer) schützt, genutzt werden, um die BSI-Grundschutzmaßnahme „Einsatz eines E-Mail-Scanners auf dem Mailserver“ (M 5.109) umzusetzen, die Bestandteil des Bausteins „Groupware“ (B 5.3) ist. Im Kontext von ISO 27001 hilft dieselbe VSA dabei, die Maßnahmen aus Kapitel 10.4 „Schutz vor Schadsoftware und mobilem Programmcode“ der ISO 27002 umzusetzen. Zugleich wirkt die VSA in beiden Fällen auf ein identifiziertes, nicht vernachlässigbares Risikopotenzial der Art „Schadvorfälle durch Schadprogramme in E-Mails“ risikoreduzierend, indem die Eintrittswahrscheinlichkeit des Schadensfalles durch den Proxy verringert wird. Dadurch kann das entsprechende Risiko unter das Risikoakzeptanz-Level eines bestehenden ISMS gebracht werden.

Im Gegensatz zu einer manuellen Einrichtung einer Malware-Erkennung kann so durch VISA mit sehr geringem Aufwand die Compliance verbessert werden, indem die entsprechende VSA im Topologie-Editor der bestehenden Topologie hinzugefügt wird. Entsprechend lassen sich für alle im Rahmen des Projekts entwickelten VSAs Komponenten der Standards identifizieren, die durch ihren Einsatz erfüllt oder zumindest teilweise umgesetzt werden, so dass der Gesamtaufwand zur Erreichung der Compliance für ein Unternehmen sinkt. Oft lassen sich durch den Einsatz sinnvoller Kombinationen von VSAs im Zusammenspiel auch komplexe Sicherheitsmaßnahmen umsetzen.

5.2 Compliance durch Topologie-Editoren

Durch den Einsatz von Topologie-Editoren wird eine systematische Umsetzung sicherer virtueller Netze ermöglicht. Obwohl die Nutzung solcher Tools nicht in den verbreiteten Sicherheitsstandards vorgesehen sind, wird durch ihren Einsatz der Aufbau standardkonformer ISMS vereinfacht: So enthalten die verschiedenen Grundschutzbausteine der Gruppe „Netze“ (B 4), insbesondere „Heterogene Netze“ (B 4.1) Phasen, in denen ein Netzkonzept und Netzpläne entwickelt werden. Ähnliche Anforderungen stellt auch ISO 27002, etwas abstrakter formuliert, im Kapitel „Maßnahmen für Netze“ (10.6.1). Der Standard „ISO 27033-1: Network Security“ enthält entsprechende Anforderungen an eine systematische Umsetzung einer Sicherheitsarchitektur sowie an eine Testphase. Testphasen sind auch im Grundschutz und in ISO 2700x vorgesehen (bei ISO 2700x eher indirekt durch „Systemabnahme“ (10.3.2)). Sowohl die Umsetzung als auch das Testen wird durch Topologie-Editoren vereinfacht und systematisiert.

Die Umsetzung kann automatisch dokumentiert und Sicherheitsmaßnahmen direkt getestet werden. Anhand von Topologie-Editoren lässt sich außerdem die Einhaltung der verschiedenen weiteren Sicherheitsanforderungen, die an Netze gestellt werden, leichter überprüfen bzw. umsetzen.

Darüber hinaus unterstützen die Topologie-Editoren bei einem sicheren Management von Netzen. Einige Grundschutzmaßnahmen, bei deren Umsetzung Topologie-Editoren helfen können, sind „Dokumentation der Sicherheitsprozesse“ (M 2.201), „Entwicklung eines Netzkonzeptes“ (M 2.141), „Entwicklung eines Netz-Realisierungsplans“ (M 2.142), „Ist-Aufnahme der aktuellen Netzsituation“ (M 2.319), „Vorgaben zur Dokumentation und Kennzeichnung der IT-Verkabelung“ (M 2.396), „Auswahl einer geeigneten Netztopologie“ (M 5.2), „Geeigneter Einsatz von Elementen zur Netzkopplung“ (M 5.13) sowie im Bereich ISO 2700x die zahlreichen Anforderungen von Kapitel „Zugangskontrolle für Netze“ (11.4).

5.3 Modellierung der VSAs

Um in einem IT-Umfeld einsetzbar zu sein, das durch die Compliance zu Sicherheitsstandards betroffen ist, müssen die VSAs der Topologie-Editoren selbst im Sinne dieser Standards sicher sein. Das zu erfüllende Sicherheitslevel einer konkreten Anwendung variiert mitunter stark. Damit sind auch die umzusetzenden Sicherheitsmaßnahmen, um die Forderungen der Standards zu erfüllen, sehr unterschiedlich. Für VISA wird davon ausgegangen, dass jeweils der höchste Schutzbedarf bzw. die höchste Kritikalität der entsprechenden IT-Prozesse zutrifft. Wesentliche technische Sicherheitsmaßnahmen für VSAs und ihre Umgebung ergeben sich dadurch automatisch und wurden beim Design mit berücksichtigt. Weitere Maßnahmen sind ebenfalls implementiert, aber optional. Hierzu wird die Möglichkeit der Konfiguration über die Topologie-Editoren genutzt. Auf diese Weise kann konfiguriert werden, welche Sicherheitsmaßnahmen in welcher Ausprägung genutzt werden soll.

Die Auswahl der Sicherheitsmaßnahmen orientiert sich an den Vorgaben der Standards. Im Falle des BSI-Grundschutzes lassen sich den VSAs konkrete Bausteine zuordnen, aus denen sich konkrete Maßnahmen ergeben. Aus ISO 27001 bzw. ISO 27002 lässt sich ein ähnliches Paket von Sicherheitsmaßnahmen entnehmen (z.B. in Bezug auf die Zugriffskontrolle), auch wenn diese allgemein etwas abstrakter formuliert sind. Die entwickelten VSAs setzen diese Maßnahmenpakete um. Sie setzen hierzu auf etablierten Technologien sowie Open-Source-Tools auf und erreichen so neben einer hohen Sicherheit auch eine gute Kompatibilität mit bestehenden Umgebungen. Da die VSAs wenig bis keine Nutzerinteraktion benötigen, ist nur eine verhältnismäßig kleine Zahl von Sicherheitsmechanismen der erwähnten Standards umzusetzen. Tatsächlich erfüllen sie bereits in ihrer Grundkonfiguration die meisten dieser technologischen Maßnahmen.

Sicherheitsmaßnahmen können allgemein von technischer und von organisatorischer Natur sein. Während die technischen von den entwickelten Topologie-Editoren wie beschrieben erfüllt werden, müssen die organisatorischen im konkreten Einsatzumfeld umgesetzt werden. Diese organisatorischen Maßnahmen werden in der Praxis abhängig vom Einsatzgebiet, Komplexität und weiteren Rahmenbedingungen sehr unterschiedlich implementiert, weshalb sie im Rahmen von VISA nicht allgemein modelliert wurden, sondern stattdessen an einem konkreten Anwendungsfall auf die bestehende IT-Umgebung zurechtgeschnitten und in deren Betriebsprozesse integriert worden sind.

6 Vergleich und Fazit

Bestehende Prozesse wie das „Change Management“ müssen auf die VSAs zusätzlich angewandt werden, um ein vollständige Abdeckung der Sicherheitsmaßnahmen zu erreichen. Hierfür ist ein Compliance-Katalog vorgesehen, der angibt, welche organisatorischen Maßnahmen umgesetzt werden müssen, inkl. der notwendigen Schnittstellen. Dadurch erhalten die Topologie-Editoren eine hohe Flexibilität, um sie in unterschiedlichen Netzen einsetzen zu können.

Tab. 1: Gegenüberstellung der drei Topologie-Editoren

Eigenschaften	Virtual Wizard (VirWi)	VISA Topologie Editor (VTE)	Collax Spotlight
Erheben eines bestehenden Netzes	Nein	Bedingt	Nein
Erstellen von Netztopologien	Ja	Ja	Nein
Erstellen von VMs	Ja	Bedingt	Ja
Erstellen aktiver Komponenten	Bedingt	Bedingt	Nein
Erstellen von Kabelverbindungen	Ja	Ja	Nein
Speicherung von Netzwerkplänen	Ja	Ja	Nein
Verwalten virtueller Festplatten	Ja	Nein	Ja
Verwalten von Bridges	Ja	Ja	Nein
Verwalten von Netzwerken	Ja	Ja	Nein
Verwalten von VMs	Ja	Bedingt	Ja
Verschlüsselte VM-Verbindungen	Ja	Ja	Ja
Gruppierung von Komponenten	Nein	Ja	Nein
Tool-Analyse der Konfiguration	Nein	Ja	Nein
VLAN-Unterstützung	Ja	Ja	Ja
Fat-/Web-Client	Ja/Nein	Nein/Ja	Nein/Ja
Datenbank-Unterstützung	Ja	Ja	Ja
Compliance-Unterstützung	Ja	Ja	Ja
Lizenz	Open Source	GPLv3	Proprietär

Die hier vorgestellten Topologie-Editoren verfolgen unterschiedliche Ansätze, um eine virtuelle Umgebung erstellen, verwalten und konfigurieren zu können, wie auch die Tab. 1 verdeutlicht. Alle drei verfolgen aber das gemeinsame Ziel, eine zentrale Management- und Verwaltungsmöglichkeit für VMs zu schaffen. Wenn man dies mit Analyse-Funktionen und Netztopologie-Konfiguration koppelt, wie dies in VISA geschehen ist, können sichere und leistungsfähige VSA-Komponenten entstehen, die bereits einen Großteil der Compliance-Anforderungen abdecken, die an ein Unternehmen heute gestellt werden. Zusätzlich können so im Vorfeld Netztopologien getestet werden, bevor sie in die Produktivumgebung gelangen. Somit ermöglichen Topologie-Editoren in jedem Fall eine Erhöhung des Sicherheitsgrads.

7 Danksagung

Das VISA-Projekt ist ein gefördertes BMBF-Projekt mit einer Laufzeit von zwei Jahren, das im August 2011 seine Arbeiten begonnen hat. An dem Projekt sind die Firmen DECOIT GmbH (Projektleitung), Collax GmbH, IT-Security@Work GmbH sowie die deutschen Forschungseinrichtungen Fraunhofer SIT und Fachhochschule Dortmund beteiligt. Zusätzlich ist der australische Partner NICTA (National ICT Australia) mit im Konsortium vertreten.