Roßbach, Peter; Gießamer, Dirk

Working Paper

Ein eLearning-System zur Unterstützung der Wissensvermittlung von Web-Entwicklern in Sicherheitsthemen

Frankfurt School - Working Paper Series, No. 116

Provided in Cooperation with:
Frankfurt School of Finance and Management

Suggested Citation: Roßbach, Peter; Gießamer, Dirk (2009) : Ein eLearning-System zur Unterstützung der Wissensvermittlung von Web-Entwicklern in Sicherheitsthemen, Frankfurt School - Working Paper Series, No. 116, Frankfurt School of Finance & Management, Frankfurt a. M., https://nbn-resolving.de/urn:nbn:de:101:1-200907211333

Frankfurt School – Working Paper Series

No. 116

Ein eLearning-System zur Unterstützung der Wissensvermittlung von Web-Entwicklern in Sicherheitsthemen

by Peter Rossbach, Dirk Gießamer

März 2009

Frankfurt School of Finance & Management
Bankakademie | HfB

Sonnemannstr. 9 – 11 60314 Frankfurt am Main, Germany
Phone: +49 (0) 69 154 008 0 Fax: +49 (0) 69 154 008 728
Internet: *www.frankfurt-school.de*

Abstract

With the development of new internet technologies new security problems arise. Thus, it is important to consider security aspects during the complete development process of a web application. Every participant of such a process should have the appropriate knowledge. Classical learning methods are not sufficient, because they are not able to satisfy the needs of knowledge in the required topicality. As a consequence, we developed an eLearning application that supports the transfer of this knowledge time- and location-independent. To get a deep understanding about security problems and the resulting vulnerabilities and attacks the application consists of three complementing components. One component is a tool that provides all necessary information about the security problems. The second component is a vulnerable online banking application where the user can apply attacks in a virtual environment. The third component is a monitor application where one can inspect in real-time which information the attacker receives.

Key words: eLearning, IT-Security, Internet Applications

ISSN: 14369753

Contact:

Peter Roßbach
Frankfurt School of Finance & Management,
Sonnemannstr. 9-11,
60314 Frankfurt,
Tel: +49 69 154008739,
Fax: +49 69 1540084739
eMail: p.rossbach@frankfurt-school.de

Dirk Gießamer
Deutsche Bank AG
Grosse Gallusstrasse 10-14
60311 Frankfurt am Main
Tel: +49 69 95630525

eMail: dirk.giessamer@db.com

Content

1 Einleitung

Die Fortschritte im Bereich der Web-Technologien, insbesondere die Web 2.0 Technologien, führen zu immer komplexeren Websites aus Sicht der Entwicklung. Dies betrifft in gleichem Maße die Komplexität hinsichtlich der Sicherheit der Web-Anwendungen. Hier ist in der vergangenen Zeit eine deutliche Zunahme der Anzahl verschiedenster Angriffstechniken sowie der auch tatsächlich ausgeführten Angriffe zu verzeichnen. Insbesondere Unternehmen stehen vor dem Dilemma, einerseits die neuen Technologien einsetzen zu müssen, da sie zu Wettbewerbsvorteilen führen können oder schlicht vom Kunden erwartet werden, und andererseits die Herausforderungen hinsichtlich der Sicherheitsaspekte zu meistern.

Je komplexer die Technologien werden, desto besser sollten die Gestaltung der Entwicklungsprozesse und die verwendeten Entwicklungstools bei der Einhaltung von Sicherheitsniveaus unterstützen. Hier sind in der Praxis vor allem im Hinblick auf Web 2.0 Technologien noch deutliche Schwachstellen zu beobachten. Zudem müssen aber auch die Entwickler Spezialisten in Sicherheitsfragen sein, da die meisten Sicherheitslücken durch eine nicht ausreichend sicherheitsbewusste Systementwicklung entstehen.

Durch die zunehmende Verwendung von client-seitigen Programmiertechniken, wie JavaScript in Verbindung mit AJAX, steht zunehmend auch der Kundenrechner im Mittelpunkt der Sicherheitsproblematik. Es reicht nicht mehr aus, sich auf die Sicherheit der Server auf Unternehmensseite zu konzentrieren, die Unternehmen tragen auch eine Verantwortung für die Sicherheit auf Kundenseite, mindestens soweit sie die Programmteile betrifft, die als Web-Anwendung vom Unternehmen ad hoc an den Kundenrechner übertragen und dort ausgeführt werden. Während sich in der Nutzung die Vorteile von client- und serverseitigen Technologien vereinen, vereinen sich gleichzeitig auch deren Nachteile in der Sicherheit.

Aufgrund dieser Problematik ist es von essenzieller Bedeutung, dass alle an einem Web-Applikationsprojekt Beteiligten, wie Entwickler, Designer und Tester, über hinreichende Kenntnisse verfügen, um der Vielzahl an möglichen Sicherheitsproblemen bereits innerhalb des Entwicklungsprozesses zu begegnen. Dazu müssen diese entsprechend ausgebildet werden. Bedingt durch die Schnelllebigkeit der Technologien sowie der Häufigkeit des Auftretens neuer Sicherheitsprobleme handelt es sich hier um einen hochgradig dynamischen Bereich, der eine ständige Wissensaktualisierung erfordert.

Klassische Schulungs- und Weiterbildungskonzepte greifen hier oft zu kurz, da ihre Durchführung i.d.R. nicht zeitlich synchron mit dem Erfordernis des Wissensbedarfs verläuft. Moderne Ausbildungskonzepte in diesem Bereich müssen daher flexibel hinsichtlich der zeitlichen Nutzbarkeit sowie der inhaltlichen Gestaltung sein. Dies bedeutet, dass sie stets zeitnah und aktuell über Sicherheitsprobleme informieren sowie möglichst ad hoc nutzbar sind. Damit ist die Nutzung computer-basierter Ausbildungsmedien unabdingbar. Hinzu kommt, dass das Wissen den Bedarfsträgern möglichst umfassend und verständlich vermittelt werden muss, da ein Halbwissen aufgrund der Bedeutung der Sicherheitsthemen nicht ausreichend sein kann.

Um hierzu einen Beitrag zu leisten, wurde an der Frankfurt School of Finance & Management das Demobox-System entwickelt. Es handelt sich um ein multiperspektivisches Tool, das einerseits über Sicherheitsprobleme bei modernen Web-Anwendungen informiert und dem Nutzer andererseits die Gelegenheit bietet, diese an einer entsprechend angreifbaren Online-

Banking-Anwendung auszuprobieren. Damit werden spielerisch am Exempel die Ursachen sowie die Auswirkungen von derartigen Angriffstechniken bis hin zu den sich bietenden Möglichkeiten für den Angreifer im Hintergrund aufgezeigt und ein entsprechendes Bewusstsein geschaffen.

Im Folgenden soll zunächst die Architektur der Demobox beschrieben werden. Im Anschluss daran wird beispielhaft die Anwendung des Systems demonstriert. Schließlich werden die derzeit implementierten Angriffstechniken aufgeführt und die Arbeit mit einem Ausblick auf die Weiterentwicklung beendet.

2 Architektur der Demobox

2.1 Anforderungen an das System

Zu Beginn der Planung des Demobox-Systems stand zunächst die Erfassung der Anforderungen, die an ein derartiges System zu stellen sind. Es wurden folgende Anforderungen identifiziert:

1. Verständlichkeit
   Ein Schulungssystem, das die Vermittlung eines komplexen Wissensbereichs zum Gegenstand hat, muss in erster Linie die Anforderung der Verständlichkeit erfüllen. Der Anwender sollte die dargebotenen Inhalte nach der Nutzung nicht nur einmal „gesehen“, sondern auch tatsächlich verstanden haben, um das so erworbene Wissen anschließend unmittelbar im Rahmen seiner Tätigkeit anwenden zu können. Dies erfordert eine nach didaktischen Prinzipien aufbereitete Gestaltung der Inhalte. Andererseits ist aber auch bekannt, dass die aktive Einbeziehung des Anwenders einen höheren Lerneffekt erzeugt als die rein passive Präsentation von Inhalten. Entsprechend folgt die Aufbereitung der Inhalte in Form eines medialen Mix mit Eigenanteil des Anwenders.

2. Multiperspektivität
   Die Anforderung nach einer Multiperspektivität hängt eng mit der Verständlichkeit zusammen. Um Angriffstechniken richtig zu verstehen, ist es sinnvoll, sie aus den Perspektiven aller Beteiligten darzustellen. Diese sind i.d.R. der eigentliche Anwender, z.B. der Bankkunde, das Unternehmen, das den Webdienst offeriert, z.B. in Form einer Online-Banking-Anwendung, und der Angreifer, der über die jeweilige Angriffstechnik bestimmte Handlungsmöglichkeiten erhält, z.B. in Form von vertraulichen Informationen, die dann für schädigende Aktivitäten ausgenutzt werden können. Die Darstellung dieser Perspektiven kann dem Nutzer der Demobox einerseits für das Verständnis der ursächlichen Zusammenhänge von Nutzen sein und andererseits helfen, bei der Gestaltung der eigenen Systeme die ursächlichen Schwachstellen zu vermeiden.

3. Benutzerfreundlichkeit
   Eine der maßgeblichen Ursachen für die Nicht-Akzeptanz von Software-Systemen ist das Fehlen einer vom Anwender wahrgenommenen Benutzerfreundlichkeit. In Ergän-

zung zur Verständlichkeit der Inhalte geht es hier in erster Linie um die Benutzerführung, mittels derer der Anwender zu den entsprechenden Inhalten gelangen kann. Für die Demobox folgt daraus die Gestaltung von flachen und zueinander vernetzten Inhaltsstrukturen, so dass der Anwender sich nicht in hierarchisch zu tiefen Inhaltsstrukturen verliert. Ziel ist es dabei, die Benutzerführung so zu gestalten, dass nur ein Minimum an Mausklicks und Tastatureingaben notwendig ist. Zur Evaluierung der Benutzerfreundlichkeit ist eine enge Einbeziehung von typischen Vertretern des späteren Benutzerkreises während der Entwicklung unabdingbar.

4. Einfache Erweiterbarkeit
   Bei Web-Technologien generell und bei den darauf abzielenden Angriffstechnologien im Speziellen handelt es sich um ein hochgradig dynamisches Gebiet. Es werden ständig neue Angriffsformen entwickelt, deren Kenntnis für die am Entwicklungsprozess von Web-Applikationen beteiligten Personen von Bedeutung ist. Somit muss ein Schulungssystem auch in der Lage sein, ohne eine zeitaufwändige Weiterentwicklung um derartige Angriffstechniken erweitert werden zu können. Eine solche Erweiterbarkeit ist eine Eigenschaft, die bereits bei der Architekturgestaltung des Systems zu berücksichtigen ist.

5. Universelle Einsetzbarkeit / Portabilität
   Schulungs- und Weiterbildungsmaßnahmen, die nicht zeitlich organisiert sind, werden von Mitarbeitern oft in Arbeitsphasen, in denen weniger zu tun ist, bzw. in der Freizeit in Anspruch genommen. Entsprechend sollte das System keine Einschränkung hinsichtlich seiner Einsetzbarkeit haben, sondern möglichst orts-, zeit- und Rechnerungebunden genutzt werden können.

Die aufgeführten Anforderungen haben alle einen Einfluss auf die Gestaltung der Architektur des Systems. Zudem haben einige ebenfalls einen Einfluss auf die inhaltliche Gestaltung. Letztere kann im Hinblick auf ein produktiv eingesetztes System noch um die Anforderungen Vollständigkeit im Hinblick auf die relevanten Angriffsformen und Aktualität, um das Wissen möglichst zeitnah zu vermitteln, erweitert werden.

2.2 Technische Systemarchitektur

Basierend auf den Anforderungen wurde der Aufbau des Systems geplant. Dazu wurden zunächst die notwendigen Komponenten identifiziert. Aufgrund der Fokussierung auf Web-Technologien ist zur Umsetzung des Systems eine Server- und eine Client-Seite notwendig.

Als erste wichtige Frage ergab sich, ob das System als Closed-Box-Lösung oder als Web-Applikation realisiert werden sollte. Im Falle einer Closed-Box-Lösung würde das System komplett in einem eigenen, abgeschlossenen Mikrokosmos realisiert, während bei einer Web-Applikation lediglich die Server-Seite umgesetzt und die Nutzer darauf mit ihren eigenen Browsern zugreifen würden.

Folgende Überlegungen führten schließlich zur Wahl der Closed-Box-Lösung: Bei einem Schulungssystem muss sichergestellt werden, dass die Demonstrationen auf Anhieb funktionieren. Dies wird am ehesten gewährleistet, wenn die verwendeten Komponenten aufeinander

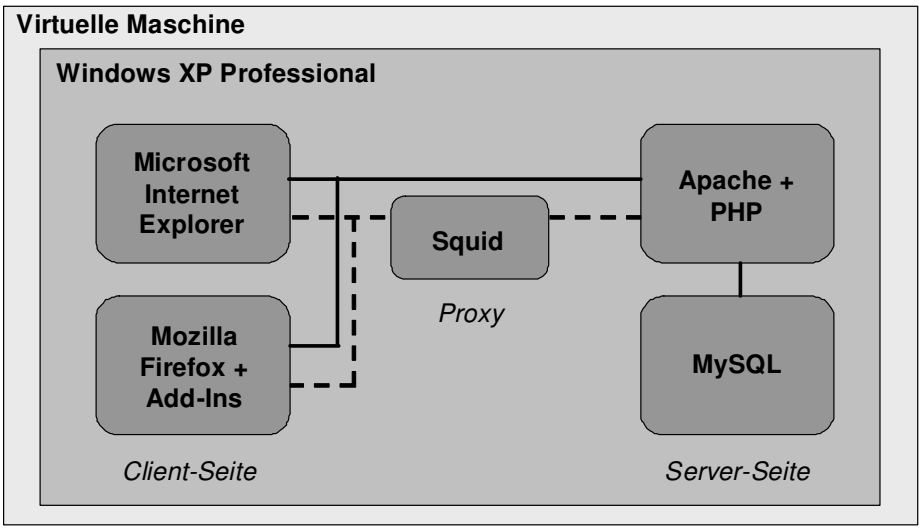
abgestimmt und getestet sind. Im Falle einer Web-Applikation kann dies nicht sichergestellt werden, da hinsichtlich Browser-Typ, -Version und -Konfiguration eine große Vielfalt besteht, die nicht ex ante vollständig berücksichtigt werden kann. Hinzu kommt, dass die einzelnen Angriffstechniken zum Teil nur auf einzelnen Browsern funktionieren, so dass der Einsatz von mehreren Browsern notwendig ist. Darüber hinaus bestehen bei einer Closed-Box-Lösung keine Erfordernisse an eine Online-Verbindung, so dass hinsichtlich der universellen Einsetzbarkeit hier ein weiterer Vorteil besteht. Schließlich müssen bei einer Closed-Box-Lösung auch keine aufwändigen Zugriffsschutzmechanismen realisiert werden, da sie nicht über das Web erreichbar ist.

Die flexibelste Variante für eine Closed-Box-Lösung ist die Realisierung des Systems in Form einer virtuellen Maschine. Hierbei simuliert eine Virtualisierungssoftware einen vollständigen PC mit entsprechender virtueller Hardware, auf dem dann ein Betriebssystem und darin wiederum die restliche Software installiert werden kann. Es läuft quasi ein Rechner in einem Rechner. Die Festplatte sowie Konfigurationen, wie die Menge an Arbeitsspeicher, werden dabei auf dem realen Rechner (Host) in Form von Dateien gespeichert und sind somit hochgradig portabel. Ist auf einem beliebigen Rechner die verwendete Virtualisierungssoftware installiert, so kann die virtuelle Maschine mittels des Verfügbarmachens der Dateien, z.B. über einen USB-Stick, gestartet und ausgeführt werden. Zudem bietet die Lösung den Vorteil, dass immer die identische Hardwarebasis und das identische Betriebssystem genutzt wird, so dass von dieser Seite ebenfalls keine Probleme hinsichtlich der Funktionsfähigkeit des Systems auftreten können.

Die Wahl fiel auf die Open-Source-Software VirtualBox von Sun Microsystems. Diese ist für die relevanten Plattformen Windows, Macintosh und Linux verfügbar, was einen hohen Grad an Portabilität gewährleistet.

Abbildung 1 zeigt die technische Architektur des Systems. Als Betriebssystem wurde Windows XP gewählt. Die Gründe dafür waren einerseits der hohe Verbreitungsgrad, womit die überwiegende Mehrzahl der Nutzer spontan mit dem von ihnen gewohnten System zurecht kommen dürften, und andererseits die Notwendigkeit, den Internet Explorer von Microsoft als einen Browser in das System integrieren zu können. Diese Notwendigkeit ergibt sich aus dem Umstand, dass der Internet Explorer der am weitesten verbreitete Browser ist und zudem spezifische Schwachstellen aufweist, die von einem Teil der implementierten Angriffstechniken ausgenutzt werden.

Abbildung 1: Technische Architektur



Als weiterer Browser wurde Mozilla Firefox ausgewählt, da dieser im Verbreitungsgrad an zweiter Stelle steht und zudem durch diverse Add-Ins für den vorliegenden Zweck sinnvoll erweitert werden kann, wie z.B. das Firebug-Add-In, das u.a. über Debugging-Funktionalitäten für JavaScript sowie eine Übersicht über den übermittelten Datenverkehr zwischen Server und Client verfügt. Damit kann dem Benutzer im Bedarfsfall ein vertiefter Einblick in die verschiedenen client-seitigen Angriffstechnologien gegeben werden. Die Hinzunahme weiterer Browser ist jederzeit möglich.

Auf Server-Seite wurde der Webserver Apache aufgrund seines Verbreitungsgrads gewählt. Die server-seitigen Anwendungen wurden in PHP programmiert, das als Modul in Apache integriert ist. Damit fiel die Wahl auf eine der am häufigsten verwendeten Sprachen für Web-Anwendungen. Als Datenbanksystem wurde MySQL gewählt, das in der Praxis häufig in Kombination mit Apache und PHP vorkommt.

Schließlich wird als Proxy noch die Open-Source-Software Squid verwendet. Ein Proxy ist notwendig für die Demonstration bestimmter Angriffe, wie z.B. dem HTTP Request Splitting, und wird innerhalb des Systems in diesen Fällen dem Webserver vorgeschaltet.

Insgesamt ist das System so gestaltet, dass die Komponenten jederzeit erweitert bzw. durch Alternativen ausgetauscht werden können, z.B. um die Java- und die .Net-Welt. Am aufwändigsten wäre dabei, wenn die Online-Banking-Applikation vollständig in eine andere Sprache umgeschrieben werden müsste.

2.3 Anwendungsarchitektur des Systems

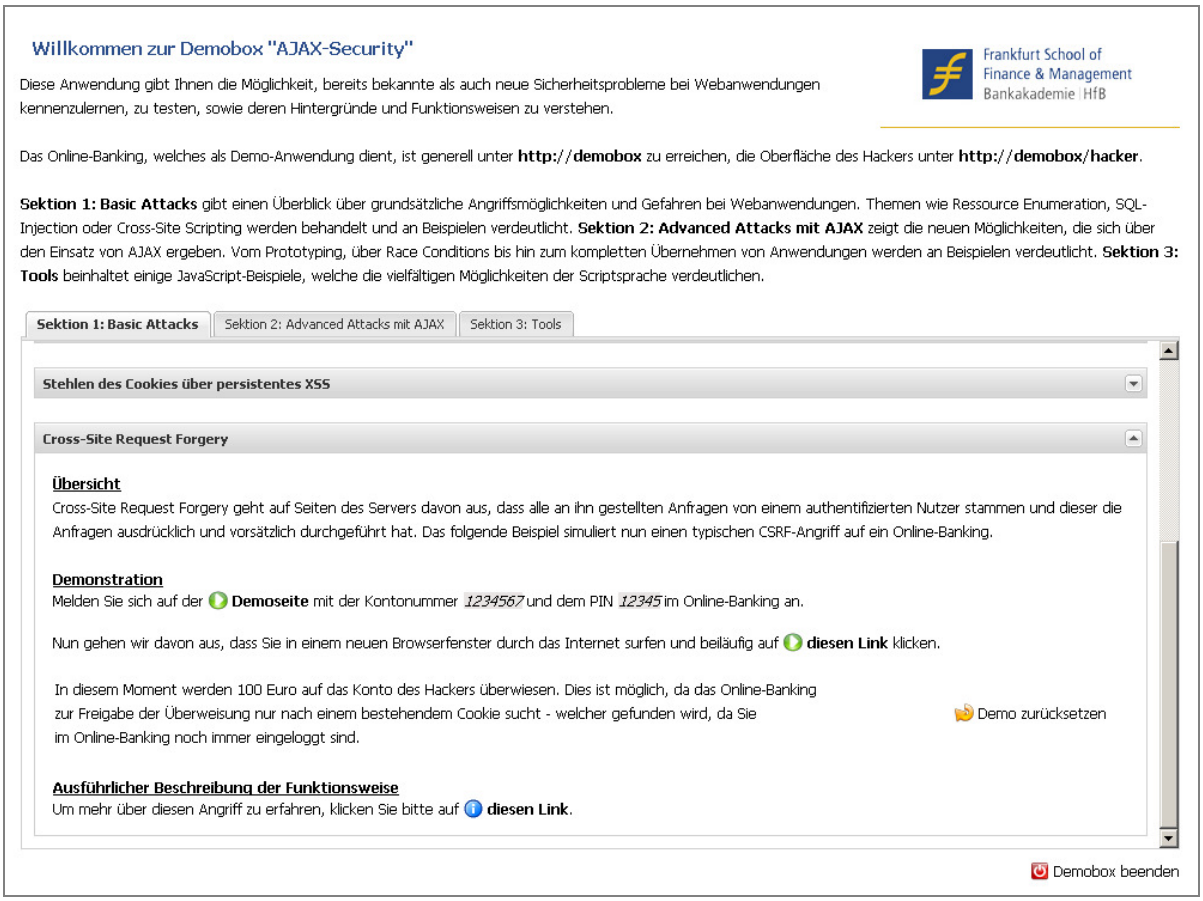
Die Anwendungsarchitektur der Demobox besteht im Kern aus drei Anwendungen. Der *Demobox-Desktop* erläutert dem Nutzer die einzelnen implementierten Angriffstechniken. Zudem können von hier aus die Demonstrationen gestartet werden. Die *Online-Banking-Applikation* ist das Anwendungsobjekt an dem die Demonstrationen durchgeführt werden. Das *Angreifer-Tool* ist ein Monitor, der dem Nutzer die mittels der Angriffe ausspionierten

Informationen zeigt, die im Normalfall beim Angreifer ankommen. Die Anwendungen sollen im Folgenden genauer beschrieben werden.

2.3.1 Der Demobox-Desktop

Nach dem Start des Systems öffnet sich automatisch die in HTML und JavaScript realisierte Desktop-Applikation als Vollbild-Anwendung. Über diese Oberfläche wird der Nutzer über die Funktionsweise der einzelnen implementierten Angriffsformen informiert. Zudem erhält er eine Beschreibung, wie die jeweilige Demonstration durchzuführen ist und kann diese direkt über den Desktop starten. Als weitere Funktionen können von hier aus das Angreifer-Tool gestartet, die für den Angriff relevanten Teile der Quelltexte angezeigt sowie die nach der Demonstration kompromittierte Online-Banking-Applikation in den Ausgangsstand zurückversetzt werden.

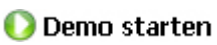
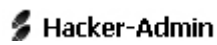
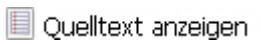
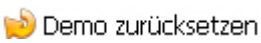
Abbildung 2: Der Demobox-Desktop



Der Desktop lässt sich in verschiedene Bereiche unterteilen, so dass die Angriffsformen nach ihren Typen gruppiert werden können (vgl. Abbildung 2). Das Wechseln zwischen den Bereichen erfolgt über Tabs, so dass neben der vertikalen Navigation auch eine horizontale Richtung besteht. Die Darstellung der einzelnen Angriffsformen wurde in Form von auf- und zuklappbaren Panels realisiert. Jedes Panel ist nach dem gleichen Prinzip aufgebaut und besteht aus einer allgemeinen Beschreibung der Angriffsform bzw. des Sicherheitsproblems sowie der Anleitung zum Durchführen der Demonstration. Zudem kann sich der Benutzer

eingehend über die Funktionsweise des Angriffs informieren, die durch Aktivieren des entsprechenden Links in übersichtlicher Form in einem neuen Fenster dargestellt wird. Weitere verwendete Elemente sind in Abbildung 3 zusammengefasst.

Abbildung 3: Interaktive Elemente eines Panels

Start der Demonstration	
Aufrufen des Angreifer-Tools	
Anzeigen relevanter Quelltext-Ausschnitte	
Zurücksetzen der Anwendung in den Originalzustand	
Ausführliche Beschreibung der Funktionsweise	

Mit dieser Art der Benutzerführung wurde ein übersichtlicher Aufbau der Anwendung erreicht. Der Nutzer hat immer die wesentliche Grundstruktur im Blick und wird strukturiert durch die einzelnen Inhalte geführt.

Derzeit ist die Desktop-Applikation in drei Bereiche unterteilt. „Sektion 1: Basic Attacks“ gibt einen Überblick über grundsätzliche Angriffsmöglichkeiten und Gefahren bei Webanwendungen. „Sektion 2: Advanced Attacks using AJAX“ zeigt die neueren Möglichkeiten, die sich durch den Einsatz von AJAX ergeben. „Sektion 3: Tools“ beinhaltet einige JavaScript-Beispiele, welche die vielfältigen Möglichkeiten der Scriptsprache verdeutlichen. Eine Beschreibung der implementierten Angriffsformen wird später in Kapitel 4 gegeben.

2.3.2 Die Online-Banking-Applikation

Als Anwendung zur Demonstration der Sicherheitsprobleme wurde das Online Banking gewählt, wobei jede andere Internetanwendung, wie Online Shop oder Ticket-System, gleichermaßen brauchbar ist. Für die einzelnen Demonstrationen wurden ein öffentlicher und ein geschlossener Bereich mit den im Folgenden beschriebenen Funktionen nachgebildet.

Der öffentliche Bereich besteht aus den drei Seiten „Home“, „Contact“ und „Investor Relations“ (vgl. Abbildung 4). Letztere wird für die Demonstrationen genutzt. Auf dieser Seite steht eine Suchfunktion bereit, die ungefilterte Eingaben entgegen nimmt und verarbeitet. Dieser Punkt ist die zentrale Sicherheitslücke des Systems, da hierüber fremder Code in die Anwendung geschleust werden kann und somit der Großteil der Angriffe hier ansetzt. Auf den Seiten des öffentlichen Bereichs wird zudem der Aktienkurs der Bank dargestellt, der sich mittels AJAX-Technologie alle 5 Sekunden aktualisiert

Abbildung 4: Startseite der Online-Banking-Applikation

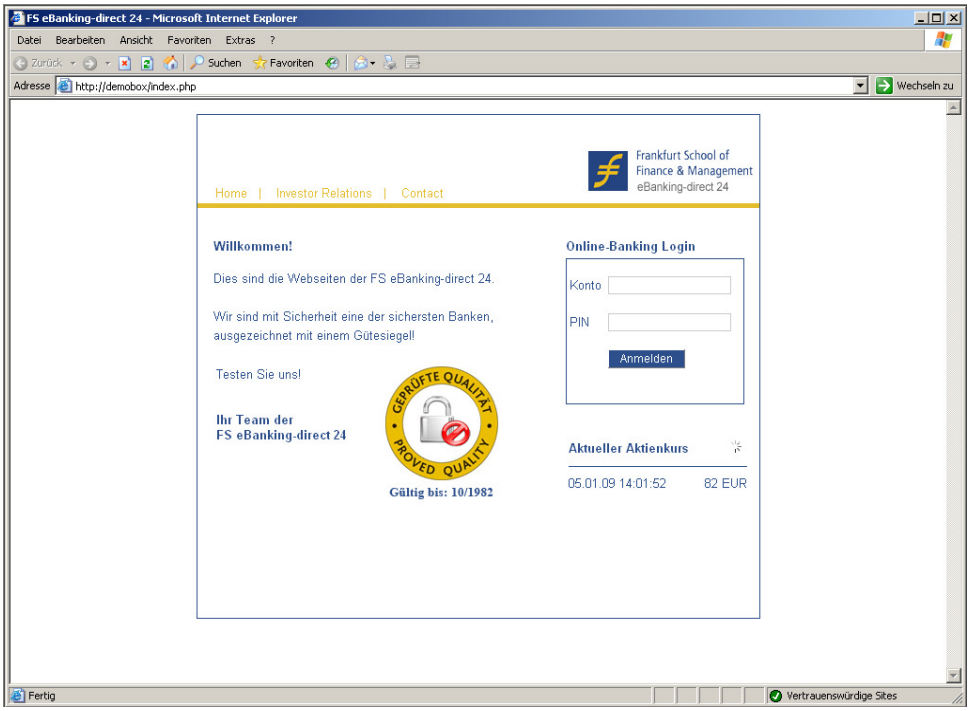
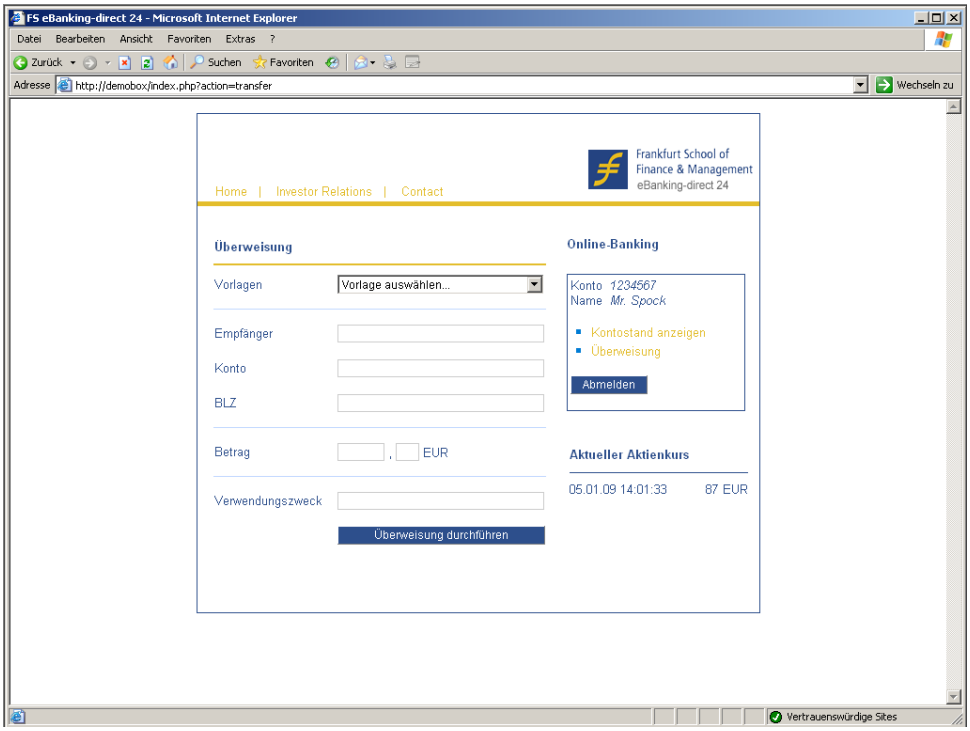


Abbildung 5: Geschlossener Bereich der Online-Banking-Applikation



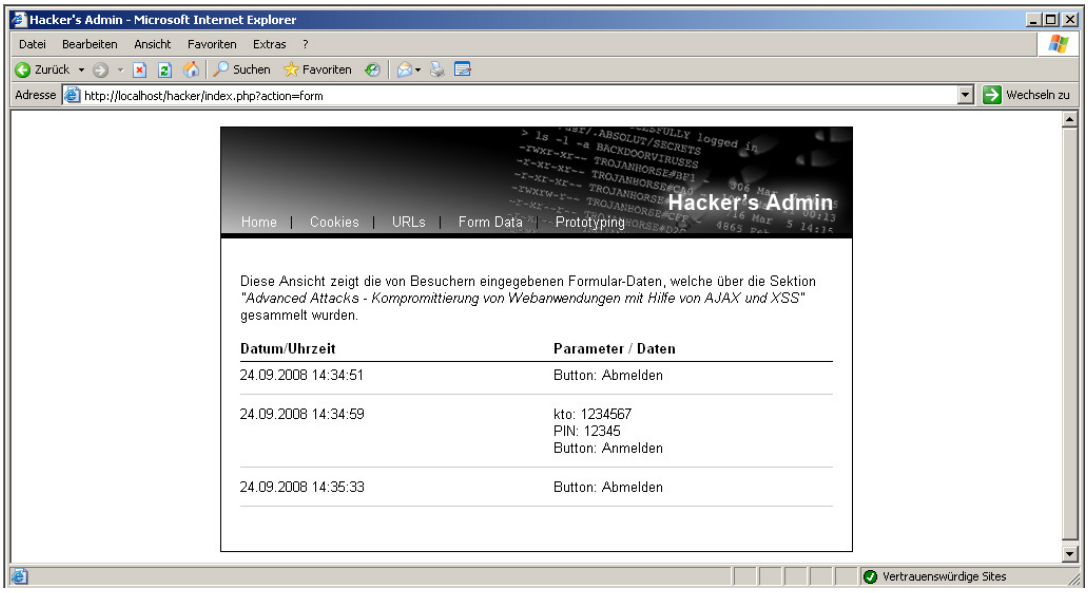
Auf allen drei Seiten des öffentlichen Bereichs steht ein Login-Formular zur Verfügung, das den Zugang zum geschlossenen Bereich über die Eingabe von Kontonummer und PIN ermöglicht. Nach erfolgreicher Authentifizierung gelangt der Nutzer auf die Übersichtseite des ge-

schlossenen Bereichs. Hier besteht nun die Möglichkeit, den Kontostand mit Detailübersicht zu den einzelnen Buchungen aufzurufen oder eine Überweisung vorzunehmen (vgl. Abbildung 5).

2.3.3 Das Angreifer-Tool

Einige der Demonstrationen sind darauf ausgelegt, sensible Daten des Benutzers auszuspionieren und an den Angreifer zu übermitteln. Diese werden dem Demobox-Nutzer über das Angreifer-Tool zur Ansicht und Auswertung bereitgestellt (vgl. Abbildung 6). Dabei handelt es sich bei den Daten z.B. um Cookie-Informationen, von Online-Banking-Nutzern aktivierte Links und eingegebene Formulardaten (beispielsweise Suchbegriffe oder Login-Daten wie Kontonummer und PIN).

Abbildung 6: Das Angreifer-Tool



3 Anwendung der Demobox

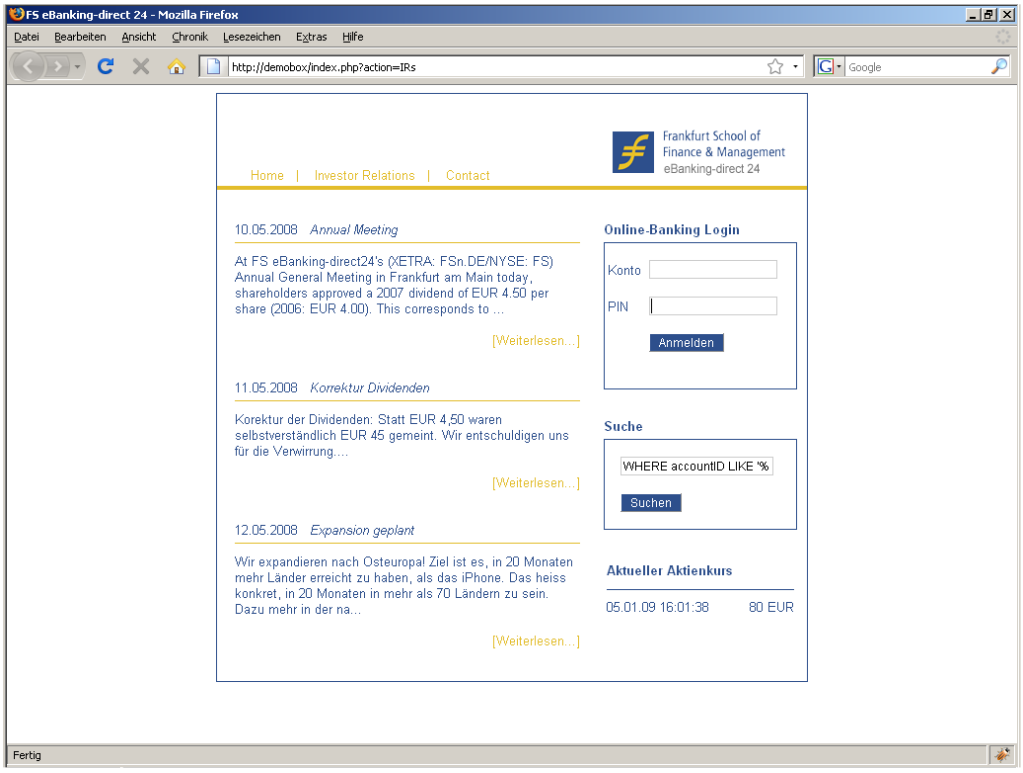
In diesem Kapitel soll die Nutzung der Demobox beschrieben werden. Dargestellt wird dies am Beispiel des AJAX Prototype Hijacking.[1] Hierbei handelt es sich um eine Angriffsform, bei der ein Angreifer durch die Injizierung von Code in eine AJAX-Applikation eine transparente, also für das Opfer unmerkbare, Zwischenschicht erzeugen kann, über die er den auf AJAX basierenden Kommunikationsfluss zwischen der browser-seitigen und der server-seitigen Applikation vollständig kontrollieren und manipulieren kann.

Im ersten Schritt wählt der Nutzer die gesuchte Angriffsform in der Desktop-Anwendung aus (vgl. Abbildung 7).

[1] Vgl. Di Paola/Fedon (2006), S.3ff.

Abbildung 7: Auswahl der Angriffsform in der Desktop-Anwendung

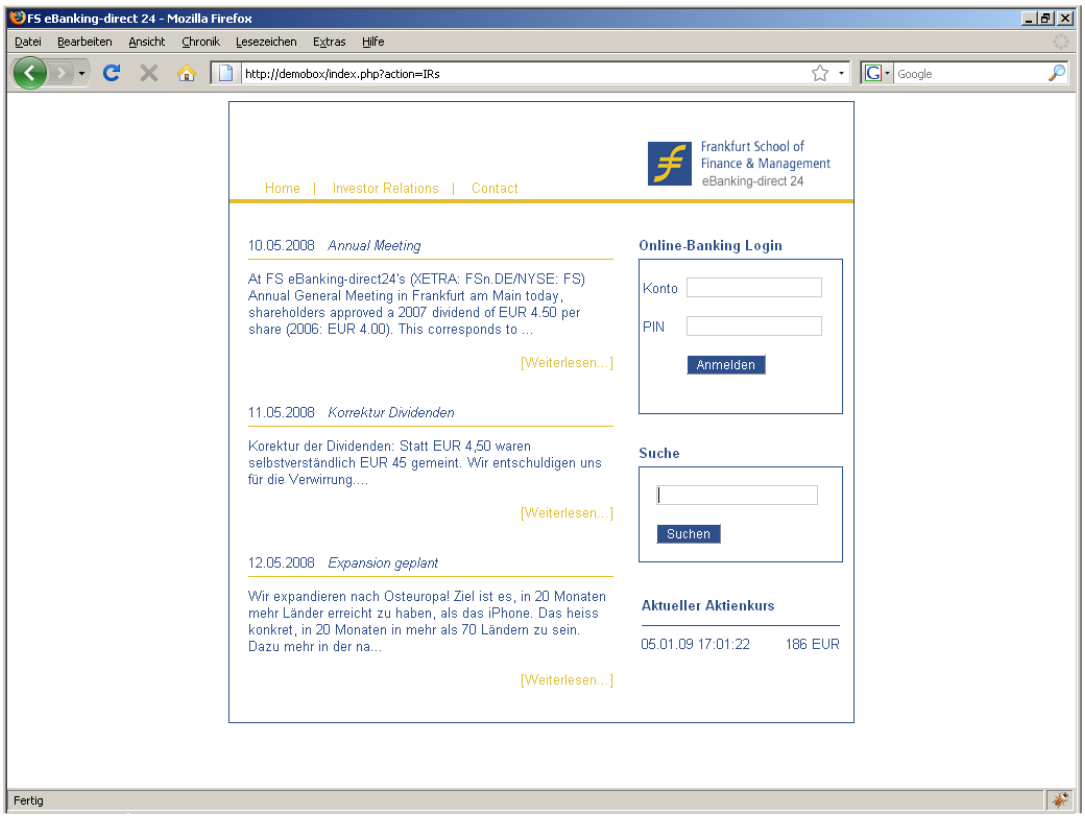
Abbildung 8: Start der Demonstration



Nach der Beschreibung der Angriffsform findet sich ein entsprechend gekennzeichneter Text, mittels dessen der Schadcode über die Suchfunktion der Online-Banking-Applikation in das System eingeschleust werden kann. Dieser kompromittierende Suchstring kann nun in die Zwischenablage kopiert und die Demonstration mittels des Links gestartet werden. Es öffnet sich der entsprechende Browser (vgl. Abbildung 8).

Der Suchstring wird nun aus der Zwischenablage in das entsprechende Textfeld kopiert und die Suche ausgeführt. In diesem Fall wird über eine SQL-Injection über die Online-Banking-Applikation ein JavaScript injiziert, das über ein sog. Prototyping die Kernmethoden des für AJAX grundlegenden XMLHttpRequest-Objekts „open()“ und „send()“ neu definiert und sämtliche Daten an den Angreifer übermittelt.[2] Gleichzeitig wird der eigentliche AJAX-Request über die Originalmethoden weiterhin normal durchgeführt, jedoch der vom Bankserver zurückgegebene Aktienkurs manipuliert, indem 100 addiert werden (vgl. Abbildung 9). Der Nutzer der Online-Banking-Applikation merkt weder dies noch realisiert er den Datendiebstahl.

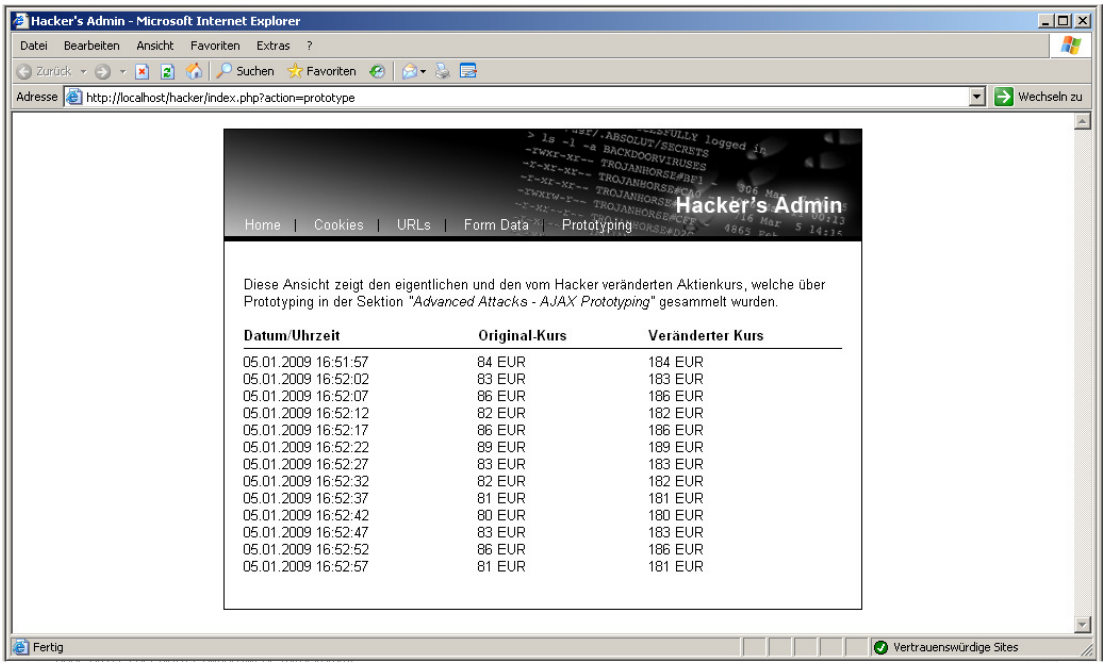
Abbildung 9: Ergebnis des Angriffs



Über das Angreifer-Tool kann der Nutzer der Demobox nun die Kommunikation zwischen Browser und Bankserver verfolgen (vgl. Abbildung 10). Dies wären auch die Informationen, die der Angreifer als Ergebnis seines Prototyping-Angriffs erhält. Die neu definierten Methoden „open()“ und „send()“ sorgen nun dafür, dass der Angreifer die Kontrolle über die Aktienkursaktualisierung erlangt mit dem Ergebnis, dass er sich die Informationen schicken lassen und, wie oben gezeigt, den Kurs manipulieren kann.

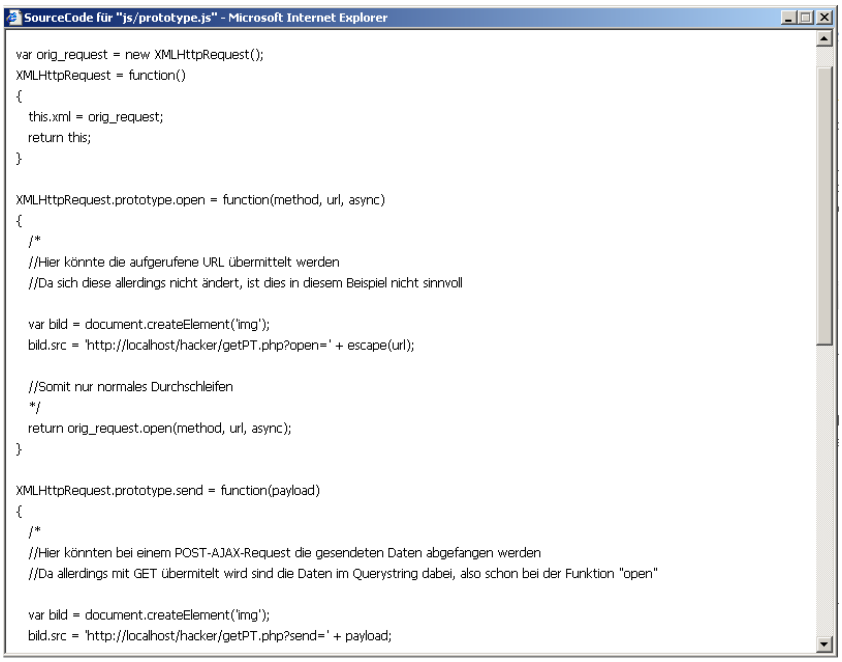
[2] Dieser Angriff nutzt die Eigenschaften von JavaScript als sog. „Prototype-based language“. Für das Vererben von Eigenschaften wird die Methode des Klonens von existierenden Objekten, den sog. Prototypen, genutzt (vgl. Steyer (2006), S.335f). Weiterhin können Objekte mit Methoden und Attributen neu gebildet und definiert werden. Auch nativen Objekten, wie dem für AJAX zentralen XMLHttpRequest-Objekt, können neue Methoden zugefügt werden.

Abbildung 10: Darstellung des Spionageergebnisses



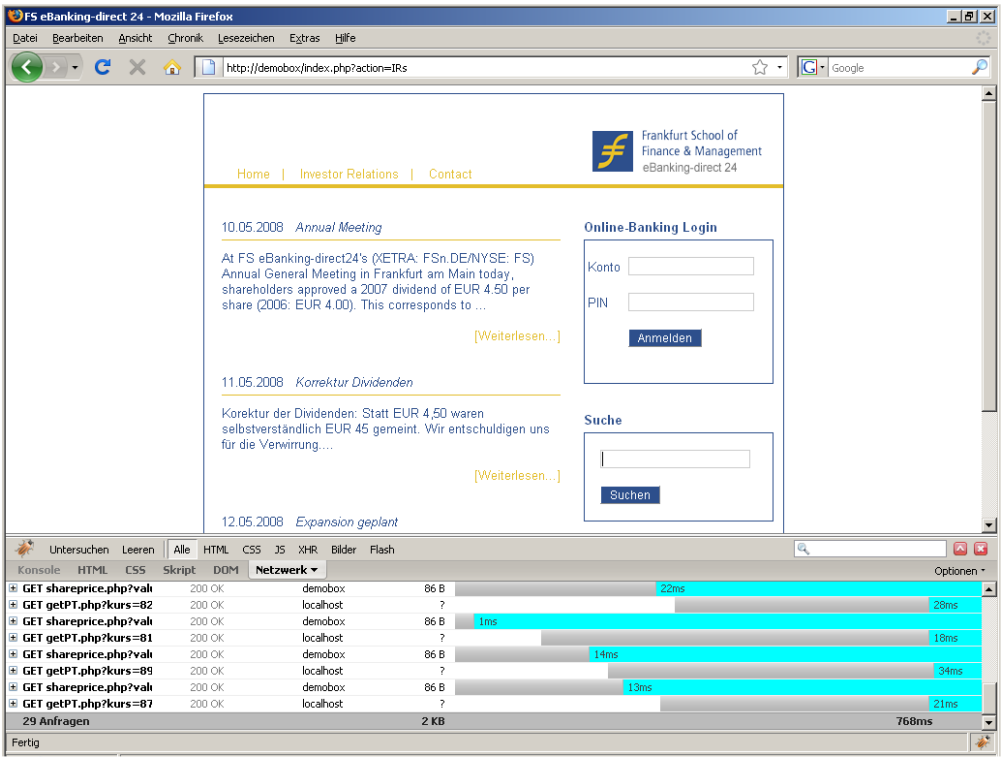
Zur Erlangung eines tieferen Verständnisses für den Angriff kann der Anwender nun noch den kommentierten JavaScript-Quellcode betrachten, indem er auf den entsprechenden Link in der Desktop-Applikation klickt (vgl. Abbildung 11).

Abbildung 11: JavaScript-Quellcode des Angriffs



Darüber hinaus kann er im Browser über das Firebug-Add-in den Verlauf der Kommunikation beobachten (vgl. Abbildung 12).

Abbildung 12: Darstellung des Kommunikationsverlaufs



Schließlich kann er das System über den Link „Demo zurücksetzen“ in der Desktop-Applikation in seinen Ausgangszustand zurückführen und sich in der Folge beliebigen anderen Angriffsformen widmen.

4 Implementierte Angriffsformen

Wie bereits in Abschnitt 2.3.1 beschrieben, sind die implementierten Angriffstechniken des Demobox-Systems in drei Bereiche unterteilt. Die Techniken sollen im Folgenden gegliedert nach diesen Bereichen kurz dargestellt werden.

4.1 Sektion 1: Basic Attacks

Ressource Enumeration
Bei der Resource Enumeration wird versucht, durch direktes Eingeben von URLs auf vorhandene Inhalte zuzugreifen, die nirgends innerhalb der Applikation verlinkt sind. Es wird dabei zwischen blindem und auf Kenntnis basierendem Erraten unterschieden.[3] Blindes Erraten sucht nach gebräuchlichen Dateien oder Ordnern, die auf dem Server liegen und wertvolle Informationen über das System liefern können, z.B. „readme.txt“ oder „WS_FTP.log“. Das kenntnisbasierte Erraten baut auf bereits erlangtem Wissen über die Applikation und dessen

[3] Vgl. Hoffman/Sullivan (2008), S. 46f.

Struktur auf. Existiert bei einem Online Banking z.B. eine Datei „konto.php“, so sucht der Angreifer nach Abwandlungen wie „konto.php.bak“ oder „Copy of konto.php“.

Manipulation von Parametern
Hier wird versucht, über systemseitig verwendete Parametermuster in den URLs auf nicht verlinkte Dateien oder noch nicht veröffentlichte Informationen zu schließen.[4] Dies können z.B. Datumsangaben, Sequenznummern oder auch Pfade sein.

SQL-Injection
SQL-Injection bezeichnet das Einschleusen von schadhaften SQL-Befehlen über von der Server-Applikation ausgeführte Datenbankabfragen.[5] Dies ist u.U. möglich, wenn die SQL-Befehle dynamisch über Benutzereingaben oder sonstige im Browser beeinflussbare Parameter zusammengebaut werden. Damit können z.B. Spionage- oder Löschangriffe auf den Server durchgeführt bzw. weitere Angriffe vorbereitet werden, indem auf diesem Weg schadhafter Code auf dem Server platziert wird, der dann bei den entsprechenden Aufrufen auf den clientseitigen Browsern übertragen wird.

Cross-Site Scripting
Cross-Site Scripting (XSS) bezeichnet die Manipulation einer Web-Applikation, so dass schadhafter Script-Code eingeschleust und auf der Browser-Seite ausgeführt wird.[6] Der Browser verarbeitet den injizierten Code, als wäre es ein legitimer Inhalt der Webseite. Ein Angreifer kann den eingelagerten bösartigen Code beispielsweise nutzen, um Informationen, z.B. über Zugangsinformationen oder Cookies, auszuspähen oder das System für eigene Zwecke zu manipulieren. Folgende Varianten[7] sind in der Demobox implementiert:

- *Reflektiertes XSS*: Hier wird schadhafter Code vom Nutzer selbst an den Webserver gesendet. Dafür muss dieser eine entsprechend präparierte URL aufrufen, die den Schadcode enthält. Diese kann ihm z.B. vom Angreifer per E-Mail geschickt worden sein oder ist als Link auf einer Webseite angelegt.
- *Persistentes XSS*: Der schadhafte JavaScript-Code wird hier auf dem Webserver eingeschleust, z.B. über eine SQL-Injection oder in ein Gästebuch. Da er nun persistent auf dem Webserver gespeichert ist, wird der Code bei jedem Aufruf der Seite dem Browser übermittelt und dort ausgeführt. Diese Methode ist gefährlicher als reflektiertes XSS, da keine Benutzeraktion für eine Kompromittierung mehr notwendig ist.
- *DOM-basiertes oder lokales XSS*: Hier wird der Schadcode nicht zum Zeitpunkt des Seitenaufrufs, sondern erst später über eine client-seite JavaScript-Funktion, z.B. via AJAX, eingeschleust und automatisch ausgeführt.

Cross-Site Request Forgery (CSRF)
Grundsätzlich geht ein Server davon aus, dass alle an ihn gestellten Anfragen von einem authentifizierten Nutzer stammen und dieser die Anfragen vorsätzlich durchgeführt hat.[8] Dieses

[4] Vgl. Wussow (2007), S. 53.
[5] Vgl Eilers (2008), S. 23.
[6] Vgl. Shah (2008), S. 121f.
[7] Vgl Eilers (2008), S. 33f.
[8] Vgl. Wussow (2007), S. 69f.

Vertrauen des Servers wird bei Cross-Site Request Forgery ausgenutzt. Als Beispiel wird in der Demobox ein Angriff auf das Online Banking simuliert, bei dem sich der Nutzer zunächst anmeldet und dann parallel in einem anderen Browser-Fenster im Internet surft. Dabei gerät er auf eine Seite mit einem präparierten Link, auf den er klickt. Da die Online-Banking-Applikation zur Freigabe der Überweisung nur nach einem bestehenden Session-Cookie sucht, kann sich der Angreifer in diesem Moment einen Betrag auf sein Konto überweisen.

4.2 Sektion 2: Advanced Attacks using AJAX

Kompromittierung von Webanwendungen mit Hilfe von AJAX und XSS
Hier werden eine Webseite durch Einschleusen von AJAX-Code über eine XSS-Schwachstelle komplett von einem Angreifer übernommen und dabei jegliche Benutzeraktivitäten sowie Formulardaten an ihn übermittelt.[9]

AJAX Prototype Hijacking
Dieser Angriff nutzt die Eigenschaften von JavaScript als sogenannte „Prototype-based language". Eine ausführliche Beschreibung des Angriffs und der Umsetzung in der Demobox wurde bereits in Kapitel 3 vorgenommen.

Race Conditions
Die parallele Abarbeitung von AJAX-Requests bringt besonders durch sog. „Race Conditions" Probleme bei der Asynchronität der Anwendung mit sich.[10] Wenn mehrere Ajax-Elemente hintereinander Anfragen an den Server senden, ist somit nicht ohne weiteres gewährleistet, dass die Antworten in der gleichen Reihenfolge kommen. Sind derartige Vorkehrungen nicht getroffen, kann es einem Angreifer gelingen, durch zeitgenaue Aufrufe der einzelnen Funktionen diesen Prozess zu seinen Gunsten umzusortieren.

HTTP Request Splitting
Über HTTP Request Splitting ist es möglich, fremde Inhalte unter einer beliebig anderen URL anzuzeigen und somit jegliche Sicherheitsprüfungen, wie beispielsweise über Zertifikate, zu bestehen.[11] Request Splitting macht sich hierbei die Asynchronität mehrerer Anfragen und die Arbeitsweise eines Proxy zu Nutze.

Lautloses Cross-Site Request Forgery
Ist es bei der Ausführung eines normalen CSRF-Angriffs nötig, dass der Nutzer auf einen präparierten Link klickt, so reicht bei CSRF unter AJAX das alleinige Besuchen einer Webseite. Der Nutzer muss sich in der Demobox lediglich im Online Banking anmelden und im Anschluss daran eine speziell präparierte Seite besuchen. In diesem Moment werden automatisch 100 Euro von dem Konto abgebucht.

[9] Vgl. Agarwal (2006).
[10] Vgl. Hoffman/Sullivan (2008), S.20.
[11] Vgl. Di Paola/Fedon (2006), S. 5.

4.3 Sektion 3: Tools

Portscanner
Diese Demonstration zeigt die Vielfalt an Einsatzmöglichkeiten von JavaScript. Es ist hierbei möglich, eine Domain oder IP-Adresse zu spezifizieren sowie die zu scannenden Ports einzugeben. Als Ergebnis wird auf der Seite angezeigt, ob die entsprechenden Ports der Seite offen oder geschlossen sind.

Zwischenablage
In dieser Demo wird gezeigt, wie über JavaScript der Zugriff auf die Zwischenablage möglich ist. Das Kopieren in und das Einfügen aus der Zwischenablage kann simuliert werden.

Browser Historie
Bei JavaScript ist es zwar nicht möglich, den Verlauf des Browsers anzuzeigen, jedoch kann auf einzelne Seiten hin überprüft werden, ob der Nutzer diese bereits besucht hat. So ist – wenn auch indirekt – nachvollziehbar, wo der Nutzer bereits gewesen ist.

Same Origin Policy umgehen
Ein wichtiges Sicherheitsmerkmal im Zusammenhang mit JavaScript und AJAX ist die sog. „Same Origin Policy". Danach sollte es nicht möglich sein, eine Verbindung zu einem anderen als dem Webserver aufzubauen, von dem die Webseite geladen wurde, wobei die Kombination aus Domain, Protokoll und Port ausschlaggebend ist. Es lässt sich jedoch zeigen, dass diese „Same Origin Policy" mit wenigen Zeilen JavaScript umgangen werden kann. Die in der Demobox umgesetzte Demonstration hebelt diese Sicherheitsvorrichtung aus und übermittelt über JavaScript respektive AJAX gestohlene Daten an einen fremden Server.[12]

5 Fazit und Ausblick

Mit der Demobox wurde ein System entwickelt, das die Unterstützung der praktischen Wissensvermittlung im Bereich Angriffstechnologien auf Web-Applikationen zum Gegenstand hat. Dabei wurde die Entwicklung konsequent auf die in Abschnitt 2.1 aufgestellten Anforderungen ausgerichtet.

So konnte das System durch den Einsatz von Virtualisierungstechnologien in Form einer Closed-Box-Lösung in hohem Maße portabel gestaltet werden und ist z.B. von einem USB-Stick lauffähig. Mit der Realisierung von Desktop, Online-Banking-Applikation und Angreifer-Tool wird die Anforderung der Multiperspektivität erfüllt. Der Nutzer des Systems kann jede Angriffstechnik von der grundsätzlichen Beschreibung der Funktionsweise über die Demonstration der Durchführung und deren Folgen auf Seiten des Angreifers bis hin zur Einsicht des Codes und des Kommunikationsverlaufs detailliert verfolgen.

Die Aufbereitung der einzelnen Anwendungsperspektiven erfolgte dabei unter Berücksichtigung der Verständlichkeit bei der Vermittlung der Vorgänge und Inhalte sowie der Benutzerfreundlichkeit hinsichtlich der Bedienerführung. Beides wurde während der Entwicklung regelmäßig unter Hinzunahme von potenziellen Nutzern getestet.

[12] Vgl. Di Paola/Fedon (2006), S. 4.

Schließlich wurde die Architektur der Anwendungen wie auch der sonstigen software-seitigen Infrastruktur so gestaltet, dass Erweiterungen mit möglichst geringem Aufwand vorgenommen werden können. So können problemlos weitere Systeme auf der Server- wie auch auf der Client-Seite hinzugefügt werden. Die drei Anwendungen Desktop, Online-Banking-Applikation und Angreifer-Tool sind so modular und offen aufgebaut, dass weitere Angriffstechniken mit vertretbarem Aufwand hinzugefügt werden können.

Mit den implementierten Angriffstechniken werden die derzeit relevanten Bedrohungen auf Web-Applikationen weitgehend abgedeckt. Es liegt in der Natur dieses Bereichs, dass dies nur von kurzer Dauer sein wird, da einerseits die Web-Technologien und andererseits auch die Angriffstechnologien ständig weiterentwickelt werden. Somit sollte auch die Weiterentwicklung der Demobox ein dynamischer Vorgang sein. Für die nächste Entwicklungsstufe ist dabei die Erweiterung auf eine Online-Update-Funktionalität geplant, so dass das lokale Exemplar des Systems immer auf dem aktuellsten Stand gehalten werden kann.

Literatur

Agarwal, A. (2006): Ajax Worm - Proof of Concept, http://myappsecurity.blogspot.com/2006/12/ajax-worm-proof-of-concept.html

Di Paola, S.; Fedon, G. (2006): Subverting Ajax, 23rd Chaos Communication Congress, Berlin, http://events.ccc.de/congress/2006/Fahrplan/attachments/1158-Subverting_Ajax.pdf.

Eilers, C. (2008): AJAX Security – Sichere Web 2.0 Anwendungen, 1. Auflage, Unterhachingen.

Heng, S.; Meyer, T.; Stobbe, A. (2008): Finanzdienstleistung und das Web 2.0: Wandel im Informationsverhalten bereitet Chancen, in: Information Management & Consulting, 23. Jg., Heft 4, S. 87-92.

Hoffman,B.; Sullivan, B. (2008): AJAX Security, 1. Auflage, Boston.

Shah, S. (2008): Web 2.0 Security: Defending AJAX, RIA, and SOA, 1. Auflage, Boston.

Steyer, R. (2006b): JavaScript. Einstieg für Anspruchsvolle, 1. Auflage, München.

Wussow, A. (2007): Sichere Webanwendungen, 1. Auflage, Frankfurt.

FRANKFURT SCHOOL / HFB – WORKING PAPER SERIES

No.	Author/Title	Year
115.	Herrmann-Pillath, Carsten Kulturelle Hybridisierung und Wirtschaftstransformation in China	2009
114.	Schalast, Christoph: Staatsfonds – „neue“ Akteure an den Finanzmärkten?	2009
113.	Schalast, Christoph / Alram, Johannes Konstruktion einer Anleihe mit hypothekarischer Besicherung	2009
112.	Schalast, Christoph / Bolder, Markus / Radünz, Claus / Siepmann, Stephanie / Weber, Thorsten Transaktionen und Servicing in der Finanzkrise: Berichte und Referate des Frankfurt School NPL Forums 2008	2009
111.	Werner, Karl / Moormann, Jürgen Efficiency and Profitability of European Banks – How Important Is Operational Efficiency?	2009
110.	Herrmann-Pillath, Carsten Moralische Gefühle als Grundlage einer wohlstandschaffenden Wettbewerbsordnung: Ein neuer Ansatz zur erforschung von Sozialkapital und seine Anwendung auf China	2009
109.	Heidorn, Thomas / Kaiser, Dieter G. / Roder, Christoph Empirische Analyse der Drawdowns von Dach-Hedgefonds	2009
108.	Herrmann-Pillath, Carsten Neuroeconomics, Naturalism and Language	2008
107.	Schalast, Christoph / Benita, Barten Private Equity und Familienunternehmen – eine Untersuchung unter besonderer Berücksichtigung deutscher Maschinen- und Anlagenbauunternehmen	2008
106.	Bannier, Christina E. / Grote, Michael H. Equity Gap? – Which Equity Gap? On the Financing Structure of Germany’s Mittelstand	2008
105.	Herrmann-Pillath, Carsten The Naturalistic Turn in Economics: Implications for the Theory of Finance	2008
104.	Schalast, Christoph (Hrgs.) / Schanz, Kay-Michael / Scholl, Wolfgang Aktionärsschutz in der AG falsch verstanden? Die Leica-Entscheidung des LG Frankfurt am Main	2008
103.	Bannier, Christina / Müsch, Stefan Die Auswirkungen der Subprime-Krise auf den deutschen LBO-Markt für Small- und MidCaps	2008
102.	Cremers, Heinz / Vetter, Michael Das IRB-Modell des Kreditrisikos im Vergleich zum Modell einer logarithmisch normalverteilten Verlustfunktion	2008
101.	Heidorn, Thomas / Pleißner, Mathias Determinanten Europäischer CMBS Spreads. Ein empirisches Modell zur Bestimmung der Risikoaufschläge von Commercial Mortgage-Backed Securities (CMBS)	2008
100.	Schalast, Christoph (Hrsg.) / Schanz, Kay-Michael Schaeffler KG/Continental AG im Lichte der CSX Corp.-Entscheidung des US District Court for the Southern District of New York	2008
99.	Hölscher, Luise / Haug, Michael / Schweinberger, Andreas Analyse von Steueramnestiedaten	2008
98.	Heimer, Thomas / Arend, Sebastian The Genesis of the Black-Scholes Option Pricing Formula	2008
97.	Heimer, Thomas / Hölscher, Luise / Werner, Matthias Ralf Access to Finance and Venture Capital for Industrial SMEs	2008
96.	Böttger, Marc / Guthoff, Anja / Heidorn, Thomas Loss Given Default Modelle zur Schätzung von Recovery Rates	2008
95.	Almer, Thomas / Heidorn, Thomas / Schmaltz, Christian The Dynamics of Short- and Long-Term CDS-spreads of Banks	2008
94.	Barthel, Erich / Wollersheim, Jutta Kulturunterschiede bei Mergers & Acquisitions: Entwicklung eines Konzeptes zur Durchführung einer Cultural Due Diligence	2008
93.	Heidorn, Thomas / Kunze, Wolfgang / Schmaltz, Christian Liquiditätsmodellierung von Kreditzusagen (Term Facilities and Revolver)	2008
92.	Burger, Andreas Produktivität und Effizienz in Banken – Terminologie, Methoden und Status quo	2008
91.	Löchel, Horst / Pecher, Florian The Strategic Value of Investments in Chinese Banks by Foreign Financial Insitutions	2008

90.	Schalast, Christoph / Morgenschweis, Bernd / Sprengetter, Hans Otto / Ockens, Klaas / Stachuletz, Rainer / Safran, Robert Der deutsche NPL Markt 2007: Aktuelle Entwicklungen, Verkauf und Bewertung – Berichte und Referate des NPL Forums 2007	2008
89.	Schalast, Christoph / Stralkowski, Ingo 10 Jahre deutsche Buyouts	2008
88.	Bannier, Christina / Hirsch, Christian The Economics of Rating Watchlists: Evidence from Rating Changes	2007
87.	Demidova-Menzel, Nadeshda / Heidorn, Thomas Gold in the Investment Portfolio	2007
86.	Hölscher, Luise / Rosenthal, Johannes Leistungsmessung der Internen Revision	2007
85.	Bannier, Christina / Hänsel, Dennis Determinants of banks' engagement in loan securitization	2007
84.	Bannier, Christina “Smoothing“ versus “Timeliness“ - Wann sind stabile Ratings optimal und welche Anforderungen sind an optimale Berichtsregeln zu stellen?	2007
83.	Bannier, Christina Heterogeneous Multiple Bank Financing: Does it Reduce Inefficient Credit-Renegotiation Incidences?	2007
82.	Cremers, Heinz / Löhr, Andreas Deskription und Bewertung strukturierter Produkte unter besonderer Berücksichtigung verschiedener Marktszenarien	2007
81.	Demidova-Menzel, Nadeshda / Heidorn, Thomas Commodities in Asset Management	2007
80.	Cremers, Heinz / Walzner, Jens Risikosteuerung mit Kreditderivaten unter besonderer Berücksichtigung von Credit Default Swaps	2007
79.	Cremers, Heinz / Traughber, Patrick Handlungsalternativen einer Genossenschaftsbank im Investmentprozess unter Berücksichtigung der Risikotragfähigkeit	2007
78.	Gerdesmeier, Dieter / Roffia, Barbara Monetary Analysis: A VAR Perspective	2007
77.	Heidorn, Thomas / Kaiser, Dieter G. / Muschiol, Andrea Portfoliooptimierung mit Hedgefonds unter Berücksichtigung höherer Momente der Verteilung	2007
76.	Jobe, Clemens J. / Ockens, Klaas / Safran, Robert / Schalast, Christoph Work-Out und Servicing von notleidenden Krediten – Berichte und Referate des HfB-NPL Servicing Forums 2006	2006
75.	Abrar, Kamyar / Schalast, Christoph Fusionskontrolle in dynamischen Netzsektoren am Beispiel des Breitbandkabelsektors	2006
74.	Schalast, Christoph / Schanz, Kay-Michael Wertpapierprospekte: Markteinführungspublizität nach EU-Prospektverordnung und Wertpapierprospektgesetz 2005	2006
73.	Dickler, Robert A. / Schalast, Christoph Distressed Debt in Germany: What´s Next? Possible Innovative Exit Strategies	2006
72.	Belke, Ansgar / Polleit, Thorsten How the ECB and the US Fed set interest rates	2006
71.	Heidorn, Thomas / Hoppe, Christian / Kaiser, Dieter G. Heterogenität von Hedgefondsindizes	2006
70.	Baumann, Stefan / Löchel, Horst The Endogeneity Approach of the Theory of Optimum Currency Areas - What does it mean for ASEAN + 3?	2006
69.	Heidorn, Thomas / Trautmann, Alexandra Niederschlagsderivate	2005
68.	Heidorn, Thomas / Hoppe, Christian / Kaiser, Dieter G. Möglichkeiten der Strukturierung von Hedgefondsportfolios	2005
67.	Belke, Ansgar / Polleit, Thorsten (How) Do Stock Market Returns React to Monetary Policy ? An ARDL Cointegration Analysis for Germany	2005
66.	Daynes, Christian / Schalast, Christoph Aktuelle Rechtsfragen des Bank- und Kapitalmarktsrechts II: Distressed Debt - Investing in Deutschland	2005
65.	Gerdesmeier, Dieter / Polleit, Thorsten Measures of excess liquidity	2005
64.	Becker, Gernot M. / Harding, Perham / Hölscher, Luise Financing the Embedded Value of Life Insurance Portfolios	2005

63..	Schalast, Christoph Modernisierung der Wasserwirtschaft im Spannungsfeld von Umweltschutz und Wettbewerb – Braucht Deutschland eine Rechtsgrundlage für die Vergabe von Wasserversorgungskonzessionen? –	2005
62.	Bayer, Marcus / Cremers, Heinz / Kluß, Norbert Wertsicherungsstrategien für das Asset Management	2005
61.	Löchel, Horst / Polleit, Thorsten A case for money in the ECB monetary policy strategy	2005
60.	Richard, Jörg / Schalast, Christoph / Schanz, Kay-Michael Unternehmen im Prime Standard - „Staying Public“ oder „Going Private“? - Nutzenanalyse der Börsennotiz -	2004
59.	Heun, Michael / Schlink, Torsten Early Warning Systems of Financial Crises - Implementation of a currency crisis model for Uganda	2004
58.	Heimer, Thomas / Köhler, Thomas Auswirkungen des Basel II Akkords auf österreichische KMU	2004
57.	Heidorn, Thomas / Meyer, Bernd / Pietrowiak, Alexander Performanceeffekte nach Directors´Dealings in Deutschland, Italien und den Niederlanden	2004
56.	Gerdesmeier, Dieter / Roffia, Barbara The Relevance of real-time data in estimating reaction functions for the euro area	2004
55.	Barthel, Erich / Gierig, Rauno / Kühn, Ilmhart-Wolfram Unterschiedliche Ansätze zur Messung des Humankapitals	2004
54.	Anders, Dietmar / Binder, Andreas / Hesdahl, Ralf / Schalast, Christoph / Thöne, Thomas Aktuelle Rechtsfragen des Bank- und Kapitalmarktrechts I : Non-Performing-Loans / Faule Kredite - Handel, Work-Out, Outsourcing und Securitisation	2004
53.	Polleit, Thorsten The Slowdown in German Bank Lending – Revisited	2004
52.	Heidorn, Thomas / Siragusano, Tindaro Die Anwendbarkeit der Behavioral Finance im Devisenmarkt	2004
51.	Schütze, Daniel / Schalast, Christoph (Hrsg.) Wider die Verschleuderung von Unternehmen durch Pfandversteigerung	2004
50.	Gerhold, Mirko / Heidorn, Thomas Investitionen und Emissionen von Convertible Bonds (Wandelanleihen)	2004
49.	Chevalier, Pierre / Heidorn, Thomas / Krieger, Christian Temperaturderivate zur strategischen Absicherung von Beschaffungs- und Absatzrisiken	2003
48.	Becker, Gernot M. / Seeger, Norbert Internationale Cash Flow-Rechnungen aus Eigner- und Gläubigersicht	2003
47.	Boenkost, Wolfram / Schmidt, Wolfgang M. Notes on convexity and quanto adjustments for interest rates and related options	2003
46.	Hess, Dieter Determinants of the relative price impact of unanticipated Information in U.S. macroeconomic releases	2003
45.	Cremers, Heinz / Kluß, Norbert / König, Markus Incentive Fees. Erfolgsabhängige Vergütungsmodelle deutscher Publikumsfonds	2003
44.	Heidorn, Thomas / König, Lars Investitionen in Collateralized Debt Obligations	2003
43.	Kahlert, Holger / Seeger, Norbert Bilanzierung von Unternehmenszusammenschlüssen nach US-GAAP	2003
42.	Beiträge von Studierenden des Studiengangs BBA 012 unter Begleitung von Prof. Dr. Norbert Seeger Rechnungslegung im Umbruch - HGB-Bilanzierung im Wettbewerb mit den internationalen Standards nach IAS und US-GAAP	2003
41.	Overbeck, Ludger / Schmidt, Wolfgang Modeling Default Dependence with Threshold Models	2003
40.	Balthasar, Daniel / Cremers, Heinz / Schmidt, Michael Portfoliooptimierung mit Hedge Fonds unter besonderer Berücksichtigung der Risikokomponente	2002
39.	Heidorn, Thomas / Kantwill, Jens Eine empirische Analyse der Spreadunterschiede von Festsatzanleihen zu Floatern im Euroraum und deren Zusammenhang zum Preis eines Credit Default Swaps	2002
38.	Böttcher, Henner / Seeger, Norbert Bilanzierung von Finanzderivaten nach HGB, EstG, IAS und US-GAAP	2003
37.	Moormann, Jürgen Terminologie und Glossar der Bankinformatik	2002

07.	Moormann, Jürgen Stand und Perspektiven der Informationsverarbeitung in Banken	1998
06.	Heidorn, Thomas / Schmidt, Wolfgang LIBOR in Arrears	1998
05.	Jahresbericht 1997	1998
04.	Ecker, Thomas / Moormann, Jürgen Die Bank als Betreiberin einer elektronischen Shopping-Mall	1997
03.	Jahresbericht 1996	1997
02.	Cremers, Heinz / Schwarz, Willi Interpolation of Discount Factors	1996
01.	Moormann, Jürgen Lean Reporting und Führungsinformationssysteme bei deutschen Finanzdienstleistern	1995

FRANKFURT SCHOOL / HFB – WORKING PAPER SERIES
CENTRE FOR PRACTICAL QUANTITATIVE FINANCE

No.	Author/Title	Year
18.	Keller-Ressel, Martin / Kilin, Fiodar Forward-Start Options in the Barndorff-Nielsen-Shephard Model	2008
17.	Griebsch, Susanne / Wystup, Uwe On the Valuation of Fader and Discrete Barrier Options in Heston's Stochastic Volatility Model	2008
16.	Veiga, Carlos / Wystup, Uwe Closed Formula for Options with Discrete Dividends and its Derivatives	2008
15.	Packham, Natalie / Schmidt, Wolfgang Latin hypercube sampling with dependence and applications in finance	2008
14.	Hakala, Jürgen / Wystup, Uwe FX Basket Options	2008
13.	Weber, Andreas / Wystup, Uwe Vergleich von Anlagestrategien bei Riesterrenten ohne Berücksichtigung von Gebühren. Eine Simulationsstudie zur Verteilung der Renditen	2008
12.	Weber, Andreas / Wystup, Uwe Riesterrente im Vergleich. Eine Simulationsstudie zur Verteilung der Renditen	2008
11.	Wystup, Uwe Vanna-Volga Pricing	2008
10.	Wystup, Uwe Foreign Exchange Quanto Options	2008
09.	Wystup, Uwe Foreign Exchange Symmetries	2008
08.	Becker, Christoph / Wystup, Uwe Was kostet eine Garantie? Ein statistischer Vergleich der Rendite von langfristigen Anlagen	2008
07.	Schmidt, Wolfgang Default Swaps and Hedging Credit Baskets	2007
06.	Kilin, Fiodor Accelerating the Calibration of Stochastic Volatility Models	2007
05.	Griebsch, Susanne/ Kühn, Christoph / Wystup, Uwe Instalment Options: A Closed-Form Solution and the Limiting Case	2007
04.	Boenkost, Wolfram / Schmidt, Wolfgang M. Interest Rate Convexity and the Volatility Smile	2006
03.	Becker, Christoph/ Wystup, Uwe On the Cost of Delayed Currency Fixing	2005
02.	Boenkost, Wolfram / Schmidt, Wolfgang M. Cross currency swap valuation	2004
01.	Wallner, Christian / Wystup, Uwe Efficient Computation of Option Price Sensitivities for Options of American Style	2004

HFB – SONDERARBEITSBERICHTE DER HFB - BUSINESS SCHOOL OF FINANCE & MANAGEMENT

No.	Author/Title	Year
01.	Nicole Kahmer / Jürgen Moormann Studie zur Ausrichtung von Banken an Kundenprozessen am Beispiel des Internet (Preis: € 120,--)	2003

Printed edition: € 25.00 + € 2.50 shipping

Download:
Working Paper: http://www.frankfurt-school.de/content/de/research/Publications/list_of_publication0.html
CPQF: http://www.frankfurt-school.de/content/de/research/quantitative_Finance/research_publications.html

Order address / contact
Frankfurt School of Finance & Management
Sonnemannstr. 9 – 11 ▪ D – 60314 Frankfurt/M. ▪ Germany
Phone: +49 (0) 69 154 008 – 734 ▪ Fax: +49 (0) 69 154 008 – 728
eMail: m.biemer@frankfurt-school.de
Further information about Frankfurt School of Finance & Management
may be obtained at: http://www.frankfurt-school.de