



Konzeption und Entwicklung des Confluence-Add-ons Mail2Blog

Sebastian Gellweiler (SG), dm-drogerie markt GmbH + Co. KG

18. September 2017
v1.0

Mail2Blog



Quellcode: <https://github.com/dm-drogeriemarkt/Mail2Blog>

Marketplace: <https://marketplace.atlassian.com/plugins/de.dm.mail2blog.mail2blog>

Hinweis

Bei diesem Dokument handelt es sich um eine für die Online-Publikation revidierte Version, der Projektarbeit „Konzeption und Einsatz eines Prozesses zur Erweiterung von Atlassian-Produkten durch Eigenentwicklungen“, die am 18.09.2017 bei der dualen Hochschule Baden-Württemberg in Karlsruhe als Prüfungsleistung des Autors eingereicht wurde. Diese Veröffentlichung ist nicht identisch mit der eingereichten Arbeit. Eine Liste der Änderungen befindet sich am Ende des Dokuments.



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung 4.0 International Lizenz.
<https://creativecommons.org/licenses/by/4.0/legalcode>

Abstract This paper describes the development process of an Atlassian add-on. The add-on Mail2Blog for Confluence was developed and published open-source in the context of this work. Mail2Blog converts emails into blog posts. The theoretical part describes Confluence and email, and analyses the security context of these technologies. The practical part describes the implementation and publication of the add-on, including implemented security measurements.

Kurzreferat Die Arbeit beschreibt den Entwicklungsprozess eines Atlassian Add-ons. Im Rahmen der Arbeit wurde das Add-on Mail2Blog für Confluence entwickelt und open-source veröffentlicht. Mail2Blog erstellt aus E-Mails automatisiert Blog-Posts. Im theoretischen Teil wird Confluence und E-Mail beschrieben und das Sicherheitsumfeld dieser Technologien analysiert. Der praktische Teil beschreibt die Implementierung und Veröffentlichung des Add-ons, inklusive getroffener Sicherheitsmaßnahmen.

Inhaltsverzeichnis

Abstract	iii
Inhaltsverzeichnis	iv
Akürzungsverzeichnis	vi
Glossar	viii
Listings	xvi
Abbildungsverzeichnis	xvii
1 Einführung	1
1.1 Firmenvorstellung	1
1.2 Arbeitsumfeld	1
1.3 Motivation und Ziele der Arbeit	1
2 Theorie	3
2.1 Wiki	3
2.2 Confluence	5
2.2.1 Seiten und Blog-Posts	5
2.2.2 Spaces	6
2.2.3 Speicher-Format	7
2.2.4 Add-ons	9
2.2.4.1 Zeitgesteuerte Jobs	10
2.2.4.2 Konfigurations-Seiten	11
2.2.4.2.1 Beispiel Add-On	13
2.3 E-Mail	17
2.3.1 E-Mail-Adresse	17
2.3.2 Format	18
2.3.3 Multipurpose Internet Mail Extensions	19
2.3.3.1 Multipart-Nachrichten	20
2.3.4 IMAP and POP3	23

Inhaltsverzeichnis

2.4	Sicherheitsanalyse	24
2.4.1	Lauschangriffe	24
2.4.2	TLS/SSL	25
2.4.3	Man-in-the-Middle-Angriff	27
2.4.4	Cross-Site-Scripting	28
2.4.5	Phishing	29
2.4.5.1	Malware	32
3	Praxis	33
3.1	Anforderungen	33
3.2	Marktanalyse	34
3.2.1	Send EMail to Page Plugin	34
3.2.2	Mail2News	34
3.2.3	Fazit der Marktanalyse	35
3.3	Struktur	36
3.3.1	Ablauf	38
3.3.2	Konfigurationssystem	39
3.3.3	Mailbox	41
3.3.4	Space-Bestimmung	42
3.3.5	Inhaltsverarbeitung	43
3.3.5.1	Parsing	43
3.3.5.2	Anhänge	45
3.3.5.3	Blog-Post-Erstellung	45
3.4	Konfigurations-UI	49
3.5	Sicherheitsmaßnahmen	52
3.5.1	HTML-Filter	52
3.5.1.1	Angriffsszenario	53
3.5.2	Dateinamen-Filter	55
3.5.2.1	Angriffsszenario	55
3.5.3	Dateityp-Filter	57
3.5.3.1	Angriffsszenario	60
3.5.4	Absender-Filter	62
3.6	Tests	63
3.6.1	Unit-Tests	63
3.6.2	Integrations-Test	64
3.7	Veröffentlichung	65
4	Fazit und Ausblick	66
Literaturverzeichnis		66
Versionshistorie		74

Abkürzungsverzeichnis

ASCII American Standard Code for Information Interchange	18
CEO Chief executive officer	31
CSV Comma-separated values	ix
CSS Cascading Style Sheets	30
DNS Domain Name System	17
FBI Federal Bureau of Investigation	31
GUID Globally Unique Identifier	45
HTML Hypertext Markup Language	7
HTTP Hypertext Transfer Protocol	27
HTTPS Hypertext Transfer Protocol Secure	27
IMAP Internet Message Access Protocol	23
ISP Internet Service Provider	23
KIT Karlsruher Institut für Technologie	36
POP Post Office Protocol	23

Akürzungsverzeichnis

RTF Rich Text Format	7
SDK Software Development Kit.....	9
SGML Standard Generalized Markup Language.....	xiii
SSL Secure Sockets Layer.....	25
TCP Transmission Control Protocol	24
TLS Transport Layer Security	25
UDP User Datagram Protocol	24
VERP Variable envelope return path.....	17
MVC Model View Controller	11
WYSIWYG What You See Is What You Get	5
XHTML Extensible Hypertext Markup Language.....	7
XML Extensible Markup Language	xv
XSS Cross-site scripting	28

Glossar

Apache Software Foundation Die Apache Software Foundation ist eine non-profit Organisation, die sich um Entwicklung und Pflege von open-source Software kümmert. Das größte und namensgebende Projekt der Organisation ist der Apache-Webserver.

Archiv Ein Archiv packt mehrere Dateien zu einer Datei zusammen. Bekannte Archivtypen sind z. B. ZIP und TAR.

ARP-Spoofing Bei diesem Angriff versucht ein Angreifer die MAC-Adresse seiner Netzwerkkarte (oder eine gefälschte) für die IP-Adresse eines Opfers zu registrieren und so anstelle dessen alle TCP- und UDP-Pakete zu erhalten. Danach kann er die, potenziell modifizierten, Pakete weiter an die MAC-Adresse des Opfers versenden, sodass das Opfer vom Angriff nichts mitbekommt.

ASCII Bei ASCII handelt es sich um eine Zeichencodierung, die allen im englischen verwendeten Ziffern, Sonderzeichen, Klein- und Großbuchstaben, sowie den wichtigsten Steuerzeichen (für Drucker und textbasierte Anzeigegeräte), einem 7 Bit breiten Wert zuweist.

Bit Eine einzelne Stelle im dualen Zahlensystem, die entweder den Wert 0 oder 1 annehmen kann.

Build-Management-Tool Eine Software, die den Kompilierungsvorgang steuert und Abhängigkeiten zu Bibliotheken verwaltet und auflöst.

Byte Eine Gruppe von 8 Bit.

Cachen Bezeichnet einen Vorgang, bei dem berechnete oder aufwendig zu ladende Werte, in einem einfach zu erreichenden Speicher (z. B. dem Hauptspeicher) zwischengespeichert werden.

Checkbox Ein HTML-Element, das einen Schalter simuliert und entweder an- oder ausgeschaltet sein kann.

Glossar

Client Eine Software, welche die Dienste eines Servers in Anspruch nimmt. Beim Client-Server-Prinzip wird die Verbindung immer vom Client aus aufgebaut. Der Client kann selber nicht direkt erreicht werden.

Cron Ein Dienst unter unixartigen Betriebssystemen, der zeitgesteuert, in Konfigurationsdateien hinterlegte, Jobs ausführt.

CSS Cascading Style Sheets (CSS) ist eine Sprache mit der die Darstellung von Dokument gesteuert werden kann.

CSV Comma-separated values (CSV) ist ein simples Format um tabulare Daten textbasiert abzubilden. Dabei werden Felder klassischerweise mit Komma getrennt, stattdessen kann aber auch jedes andere Zeichen als Seperator verwendet werden.

E-Mail-Provider Stellt einen E-Mail-Dienst für Kunden bereit. Bekannte E-Mail-Provider sind z. B. Web.de, GMX, Google (GMail), etc.

E-Mail-Reader Eine Software die E-Mails anzeigt.

Deserialization Bezeichnet den Vorgang, bei dem aus einem vorher in einen String serialisiertem Objekt wieder ein Objekt gemacht wird.

Domainname Ein hierarchisch aufgebauter Bezeichner, der einen Rechner oder ein Netzwerk von Rechnern identifiziert. Teil des DNS-Systems. Z. B. weist der Domainname www.dm.de auf einen Webserver von DM.

DNS DNS ist ein hierarchisch organisiertes Namenssystem im Internet. Rechner können eine DNS-Anfrage an DNS-Server senden, die dann den zu einem Domänenamen gehörenden Rechner versuchen zu identifizieren.

Encodierung Bildet, nicht in einem Zeichensatz darstellbare, Daten auf kompatible Zeichen ab. Siehe Kapitel 2.3.3 für Beispiele.

Fork Ein Fork ist eine Abspaltung eines bestehenden Projektes. Das neue Projekt wird eigenständig und unabhängig weiter entwickelt. Z. B. ist das Betriebssystem OpenBSD ein auf maximale Sicherheit ausgerichteter Fork von FreeBSD.

Getter In Java ist es gängige Praxis die Eigenschaften einer Klasse über Getter- und Setter-Methoden erreichbar zu machen, anstelle direkt auf Eigenschaften zuzugreifen. Auf diese Weise kann z. B. eine Eigenschaft ausschließlich lesbar oder schreibbar gemacht werden. In einer Getter-Methode können Eigenschaften z. B. über lazy-loading nachgeladen werden, oder beim Zugriff berechnet werden (magische Eigenschaften).

Glossar

Git Ein verbreitetes Versionsverwaltungssystem für Software.

Github Ein bekannter webbasierter Git-Hoster, der für Open-Source-Projekte kostenlos ist.

Gottesklasse Ein Anti-Pattern. Eine Gottesklasse ist eine Klasse, die zu viel tut und zu viel weiß. Als zentrale Klasse eines Programms erledigt eine Gottesklasse alle/viele wichtige Aufgaben, anstelle dass Funktionen in eigene Module ausgelagert werden. Steht im Widerspruch zum Grundgedanken der strukturierten Programmierung, bei der Probleme in Teilprobleme zerlegt und dann gelöst werden.

GUID Ein Globally Unique Identifier (GUID) ist eine generierte ID, die weltweit nur einmal vorkommen darf. Um das sicherzustellen, wird neben einem sehr genauen Zeitstempel in der Regel eine Mac-Adresse oder eine sehr große Zufallszahl für die Generierung verwendet.

Hex-Wert Ein Zahlenwert dargestellt im hexadezimalen (Basis 16) Zahlensystem. Für die Ziffern 0-9 werden die gleichen Zeichen wie im dezimalen Zahlensystem verwendet, die Ziffern 10-15 werden durch die Buchstaben A-F dargestellt.

HTML Hypertext Markup Language (HTML) ist eine Auszeichnungssprache, die standardmäßig im Web verwendet wird. HTML-Dokumente werden von Webbrowsern dargestellt. HTML basiert auf SGML.

HTML-Img-Tag Ein HTML-Tag mit dem ein Bild eingebunden werden kann.

HTTP Das Hypertext Transfer Protocol (HTTP) wird verwendet um Dateien im Web zu übertragen. Das Protokoll ist ähnlich wie E-Mail aufgebaut.

Integrations-Test Ein Integrations-Test testet die Interaktion zwischen Komponenten.

IP-Adresse Mit einer IP-Adresse werden Rechner im Internet und lokalen Netzwerken identifiziert.

ISP Ein Internet Service Provider (ISP) verbindet Endkunden und Firmen, z. B. über DSL oder Mobilfunk, mit dem Internet. Bekannte ISPs sind z. B. die Telekom, 1&1, Verizon, etc.

Java Java ist eine objektorientierte Programmiersprache, die 1995 veröffentlicht wurde. Java-Programme werden in Java-Bytecode übersetzt, der von einer virtuellen Maschine (Java Virtual Machine) ausgeführt wird. Java wurde von Sun Microsystems entwickelt, die 2010 von Oracle aufgekauft wurden.

Glossar

Java-Annotation Eine spezielle Anweisung, die im Quellcode über einer Java-Klasse, einer Methode oder einer Code-Zeilen steht und die während des Kompiliervorgangs ausgewertet wird.

JavaBean Eine JavaBean ist ein Objekt, das ausschließlich Eigenschaften speichert, aber keine Funktionalität implementiert (abgesehen von Gettern- und Settern).

Java-Interface Definiert öffentlich aufrufbare Methoden, die eine erbende Java-Klasse implementieren muss.

Java-Klasse Stellt eine Bauanleitung für ein Objekt da. Definiert Eigenschaften und implementiert Methoden eines Objektes.

Java-Paket Ein Namensraum in Java, der durch einen Java-Paket-Namen bezeichnet wird.

Java-Paket-Name Ein Bezeichner für ein Java-Paket. Sollte in der Regel global eindeutig sein, um das zu erreichen werden in der Regel Domänennamen zur Erstellung eines Namens verwendet. Z. B. org.example.myaddon.

JavaScript Eine Skriptsprache, die im Webbrowser ausgeführt wird. Anders als der Name vermuten lässt, hat die Sprache nichts mit Java zu tun.

lazy-loading Englisch für faules laden. Bei zu berechnenden und zu ladenden Werten (z. B. aus einer Datenbank), kann es Sinn machen die Werte beim ersten Zugriff zu berechnen/laden und danach zu cachen, anstelle die Werte schon beim Erstellen des Objekts zu ermitteln.

Mailbox Digitaler Briefkasten für E-Mails.

Markdown Eine einfache textbasierte Auszeichnungssprache, die für Menschen einfach zu lesen/schreiben ist und für Maschinen einfach in HTML und andere Auszeichnungssprachen gewandelt werden kann. Markdown ist die standardmäßig auf Github verwendete Auszeichnungssprache für Dokumentation.

Meta-Informationen Informationen über Daten, aber nicht die Daten selber. Z. B. sind das Erstellungsdatum und der Name des Autors Meta-Informationen eines Zeitungsartikels.

Mx-Record Ein DNS-Eintrag, der auf einen E-Mail-Server verweist.

Open-Source Software bei welcher der Quellcode öffentlich zugänglich gemacht wird.

Glossar

Parser Ein Parser führt eine syntaktische Analyse eines Daten-Streams nach den Regeln einer Grammatik durch und überführt das Ergebnis in ein zur Weiterverarbeitung geeignetes Format.

Payload Der eigentliche Teil einer Nachricht ohne Meta-Informationen und Übertragungs-informationen. Bei Malware der eigentliche Schadcode.

Plain-Text Text ohne Formatierung.

Primitive Datentypen Ein primitiver Datentyp speichert einen einzelnen Wert. Komplexe Datentypen (z. B. Objekte und Arrays) setzen sich aus den primitiven Typen zusammen. In Java gibt es Wahrheitswerte (boolean), Ganzzahlwerte (char, byte, int, long) und Fließkommazahlen (float, double).

Pseudocode Programmiercode, optional in einer Phantasiesprache, der zur Verdeutlichung eines Konzept geschrieben wird und nur von Menschen lesbar ist, nicht aber von einer Maschine ausgeführt werden kann.

Refactoring Bezeichnet einen Vorgang, bei dem veralteter und schwer wartbarer, Code überarbeitet und modernisiert wird.

RTF Rich Text Format (RTF) ist eine, in Textverarbeitungsprogrammen verbreitete, 1987 von Microsoft eingeführte, Auszeichnungssprache für Dokumente.

SDK Ein Software Development Kit (SDK) ist eine Sammlung von Tools zum Erstellen von Software.

Serialisation Bezeichnet einen Vorgang, bei dem ein Objekt auf einen String abgebildet wird, z. B. um ihn abzuspeichern oder zu übertragen. Bei der Deserialisation kann aus dem String dann wieder ein Objekt erstellt werden.

Sessioncookie HTTP ist ein zustandloses Protokoll. Um einen Nutzer trotzdem über mehrere Anfragen identifizieren zu können, kann der Browser angewiesen werden einen kleinen, von der Webanwendung übermittelten, Textwert (Cookie) bei jeder Anfrage mitzusenden. In diesem Cookie wird eine zufällige einzigartige generierte ID gespeichert, mit welcher der Nutzer identifiziert wird. Auf diese Weise muss auch das Passwort des Nutzers nur einmal zur Authentifizierung übermittelt werden, danach reicht es den Sessioncookie zu überprüfen. Wenn ein Angreifer in Besitz des Sessioncookies gelangt, kann er sich als das Opfer gegenüber der Anwendung ausgeben (Session hijacking).

Glossar

Setter In Java ist es gängige Praxis die Eigenschaften einer Klasse über Getter- und Setter-Methoden erreichbar zu machen, anstelle direkt auf Eigenschaften zuzugreifen. Auf diese Weise kann z. B. eine Eigenschaft ausschließlich lesbar oder schreibbar gemacht werden. In einer Setter-Methode kann z. B. der übergebene Wert validiert werden oder ein abhängiger Cache aktualisiert werden.

Server Stellt einen Dienst in einem Netzwerk zur Verfügung. Ein Client baut die Verbindung zum Server auf.

SGML Standard Generalized Markup Language (SGML) ist eine textbasierte Sprache um Daten hierarchisch strukturiert darzustellen. XML und HTML basieren auf SGML.

Skriptsprache Eine Skriptsprache ist eine Programmiersprache, die nicht kompiliert wird, sondern von einem Interpreter ausgeführt wird.

Singleton Ein Singleton ist eine Klasse, von der es nur genau eine Instanz geben kann. Die Instanz wird in der Klasse selber gespeichert und beim ersten Zugriff erzeugt. Bei allen weiteren Zugriffen wird das bestehende Objekt zurückgegeben.

SPAM Unerwünschte, dem Nutzer unverlangt zugestellte, Nachrichten. Wird in der Regel massenhaft versendet. SPAM (für Spiced Hamm) war ursprünglich der Markennamen eines günstigen Dosenfleischs. Die heutige Verwendung des Begriffs resultiert aus einem Monty Python Sketch, indem fortlaufend das Wort SPAM verwendet wird.

Strategiepattern Ein Entwurfsmuster, bei dem einzelne Strategien austauschbar implementiert werden. Z. B. wäre eine Klasse Pferd denkbar, die ein Pferd modelliert, dass sich sowohl galoppierend als auch trabend fortbewegen kann. Die Strategien Galopp und Trab würden dann in eigenen Klassen implementiert werden, die vom Interface Fortbewegen (mit der Methode fortbewegen) erben. Die Klasse Pferd enthält dann eine Eigenschaft fortbewegungsmethode die nach Bedarf auf eine Instanz der Klassen Galop oder Trab geändert werden kann. Wird dann pferd.fortbewegen() aufgerufen, wird die Aufgabe an fortbewegungsmethode.fortbewegen() delegiert.

Social Engineering Methoden mit denen Angreifer versuchen Opfer durch zwischenmenschliche Beeinflussung zu manipulieren und zum Handeln im Sinne des Angreifers zu verleiten.

TCP Das Transmission Control Protocol (TCP) wird verwendet um Daten-Pakete im Internet zu versenden. Im Gegensatz zu UDP wird eine Fehlerkontrolle durchgeführt, Pakete werden überprüft und falsch übertragene Pakete werden erneut übermittelt. Pakete eines TCP-Streams werden nummeriert, von der Gegenstelle wird ein TCP-

Glossar

Stream garantiert wieder in der gleichen Reihenfolge, wie er versendet wurde, aufgebaut.

Token-Streams Ein Programmcode wird beim Kompilieren in der Regel vor dem Parsen von einem Tokenizer in einen einfacher auszuwertenden Token-Stream übersetzt. Ein Token besteht aus einem Bezeichner und einem Wert. Tokens können z. B. Bezeichner, Schlüsselwörter, Operatoren etc. sein.

Travis-Ci Ein für Open-Source-Projekte kostenloser Buildserver.

UDP Das User Datagram Protocol (UDP) wird verwendet um Daten-Pakete im Internet zu versenden. Im Gegensatz zu TCP findet keine Fehlerkontrolle statt. Pakete werden in der Reihenfolge verarbeitet, in der sie beim Empfänger ankommen. Bei der Übertragung von z. B. Video- und Audiostreams sind vereinzelte Fehler im Datenstream in der Regel weniger störend als Latenzen, die durch den zusätzlichen Aufwand der Fehlerkontrolle entstehen würden. Daher wird hier in der Regel UDP anstelle von TCP verwendet.

Unit-Test Ein Unit-Test testet automatisiert eine einzelne Komponente einer Software. Nicht aber die Interaktion zwischen Komponenten.

Web Bezeichnet den Teil des Internets, der mit einem Web-Browser erreichbar ist. Im Web werden Dateien per HTTP übertragen.

Web-Template-Sprache Mit einer Web-Template-Sprache kann Platzhalter-Code in einem HTML-Dokument (Template) eingefügt werden, der dann von einer Template-Engine ausgeführt wird, das dabei entstehende HTML wird dann an entsprechender Stelle eingefügt. Typischerweise unterstützen Web-Template-Sprachen Variablen, einfache Konditionen und Schleifen. Komplexerer Code sollte außerhalb des Templates implementiert werden.

Whitelistfilter Ein Filter der Begriffe gegen eine Liste von erlaubten Werten prüft. Im Gegensatz zu einem Blacklistfilter, der gegen verbotene Werte prüft. Whitelistfilter sind schwerer zu umgehen als Blacklistfilter und daher in der Regel sicherer, schränken aber auch den Nutzer stärker ein.

WikiMarkup Eine textbasierte Auszeichnungssprache, die speziell für den Einsatz in Wikis konzipiert wurde.

Wurm Eine sich selbst replizierende und verbreitende Malware.

Glossar

WYSIWYG-Editor Ein Editor indem Inhalte so bearbeitet werden können, wie sie auch angezeigt werden. WYSIWYG steht für What You See Is What You Get (Was du siehst, ist das, was du bekommst).

XHTML Extensible Hypertext Markup Language (XHTML) ist eine auf HTML basierende Auszeichnungssprache zur Darstellung von Dokumenten. Im Gegensatz zu HTML ist XHTML nicht nur mit SGML kompatibel, sondern auch mit XML.

XML Extensible Markup Language (XML) ist eine vereinfachte Variante (mit strengeren Regeln) des SGML-Standards. Wurde entwickelt, um einfachere und effizientere Parser zu schreiben.

Listings

2.1	Confluence Speicher-Format	8
2.2	atlassian-plugin.xml – Meta-Informationen	10
2.3	atlassian-plugin.xml – Job-Modul	11
2.4	Java-Klasse – Job	11
2.5	Beispiel Add-On – atlassian-plugin.xml	14
2.6	Beispiel Add-On – Java-Klasse – Xwork-Kontroller	15
2.7	Beispiel Add-On – Velocity-Template – age.vm	16
2.8	Beispiel Add-On – Velocity-Template – age-success.vm	16
2.9	Beispiel Add-On – Velocity-Template – age-error.vm	16
2.10	Beispiel für eine Email	19
2.11	Beispiel für eine Multipart-E-Mail	22
3.1	Algorithmus zum E-Mail-Parsing in Pseudocode	44
3.2	HTML-Makro – E-Mail-Exploit für Send EMail to Pages	54
3.3	HTML-Makro – E-Mail-Exploit für Mail to News	54
3.4	Java-Code, der Dateinamen bereinigt	55
3.5	Dateinamen – E-Mail-Exploit	56
3.6	Malware – Virus-Test-Skript getarnt als image/png – E-Mail	59
3.7	Malware – Virus-Test-Skript getarnt als image/png – HTTP	59

Abbildungsverzeichnis

2.1	Hypothetische Größe der englischsprachigen Wikipedia in ausgedruckter Form	4
2.2	Confluence WYSIWYG-Editor	6
2.3	Confluence Blog-Historie	6
2.4	Confluence Page-Tree	6
2.5	Confluence Space	7
2.6	Beispiel Add-On – Dateibaum	13
2.7	Beispiel Add-On – UI	13
2.8	Beispiel für die Hierarchie einer Multipart-Nachricht	21
2.9	Der Package-Sniffer Wireshark	25
2.10	TLS/SSL Verbindung	26
2.11	TLS/SSL Chain-Of-Trust	27
2.12	Man-in-the-Middle-Angriff	28
2.13	Eine Phishing E-Mail	30
2.14	Eine Phishing Webseite	31
3.1	Mail2News-Klassendiagramm mit Gottesklasse Mail2NewsJob	36
3.2	Mail2Blog-Klassendiagramm	37
3.3	Mail2Blog-Datenfluss während des Abarbeitungsprozesses	38
3.4	Klassendiagramm des Konfigurationssystems	39
3.5	Klassendiagramm des Mailboxsystems	41
3.6	Klassendiagramm der Space-Bestimmung	42
3.7	Klassendiagramm der Inhaltsverarbeitung	43
3.8	Generierter Blog-Post	46
3.9	Aktivitätsdiagramm der Anhang-Verarbeitung	47
3.10	Aktivitätsdiagramm der E-Mail zu Blog-Post-Verarbeitung	48
3.11	Konfigurations-UI	50
3.12	Klassendiagramm der Konfigurationsseite	51
3.13	HTML-Filter Einstellungen	53
3.14	HTML-Makro – Send EMail to Pages	55
3.15	HTML-Makro – Mail to News	55
3.16	HTML-Makro – Mail2Blog	55
3.17	Dateinamen – Verarbeitete Anhänge	56

Abbildungsverzeichnis

3.18 Dateinamen – Suche Send EMail to Pages & Mail2News	57
3.19 Dateinamen – Suche Mail2Blog	57
3.20 Einstellungen des Dateityp-Filters	58
3.21 Malware – Virus-Test-Skript getarnt als image/png – OS X	60
3.22 Malware – Virus-Test-Skript getarnt als image/png – Confluence	60
3.23 Malware – Goldeneye – E-Mail	61
3.24 Malware – Goldeneye – Excel-Datei mit Schadcode	61
3.25 Malware – Goldeneye – Blog-Eintrag ohne Excel-Datei	62
3.26 Einstellungen des Absender-Filters	62
3.27 Aktivitätsdiagramm des Integrationstest	64
3.28 Github	65
3.29 Travis-CI	65

1 Einführung

1.1 Firmenvorstellung

dm-drogierie markt GmbH + Co. KG ist mit fast 10 Mrd. € Umsatz und über 55.000 Mitarbeitern die größte Drogeremarktkette Europas (Stand 2017) und verfügt über ein dichtes Filialnetz in Deutschland, Österreich und Südosteuropa[1]. Für Entwicklung und Betrieb der Software in den Filialen, der Zentrale und den Verteilerzentren ist die FILIADATA GmbH als Tochter von DM zuständig.

1.2 Arbeitsumfeld

Der Bereich Marketing und E-Commerce (MEC) innerhalb von FILIADATA kümmert sich um die Entwicklung der Internetpräsenz und Online-Marketing-Werkzeuge des Unternehmens. Das Web- und Softwareengineering (Web-Se) Team ist im MEC-Bereich angesiedelt und kümmert sich um die Entwicklung des Onlineshops und anderer auf Java basierender Software. Die vorliegende Projektarbeit ist im Rahmen einer Praxisphase im Web-Team entstanden.

1.3 Motivation und Ziele der Arbeit

Bei FILIADATA werden die Projektmanagement- und Entwicklertools Jira (Fehlererfassung und -verfolgung), Confluence (Dokumentation), Bitbucket (Softwareverwaltung)

Einführung – Motivation und Ziele der Arbeit

und Crucible (Code-Reviewing) der Firma Atlassian verwendet. Immer wieder gibt es im Unternehmen von diesen Systemen nicht abgedeckte Anforderungen, für die es auch keine fertigen Lösungen im Atlassian-Marketplace gibt. In Zukunft sollen hausintern Add-ons für diese Systeme entwickelt werden, die diese Aufgaben übernehmen. Zusätzlich sollen die Add-ons, sofern sie allgemein nützlich sind, auch open-source veröffentlicht werden, um von der Rückmeldung und Mitarbeiter anderer Firmen zu profitieren und das öffentliche Image der IT von DM zu stärken.

Aufgabe der Arbeit ist es: die Entwicklung eines Add-ons bis zur Veröffentlichung anhand eines konkreten Beispiels zu dokumentieren. Dazu wurde in der Findungsphase der Arbeit das im folgenden beschriebene Problem als Musterbeispiel ausgewählt.

Im Unternehmen wurden regelmäßig relevante Presseartikel gesammelt und per E-Mail an alle Mitarbeiter versendet. Das führte dazu, dass die Postfächer der Mitarbeiter mit Nachrichten vollliefen und sich Beschwerden über die entstehende E-Mail-Flut häuften. Daher wurde entschieden, die Presseartikel stattdessen zentral, in einer Art digitalen Pressemappe, in dem firmenintern verwendeten Wiki Confluence zu sammeln. Dafür bot sich der Blog-Bereich von Confluence an. Da die Nachrichten ursprünglich per E-Mail eingehen und die für die Verteilung verantwortlichen nicht gezwungen werden sollten die Blog-Posts per Hand zu erstellen, sollte stattdessen ein Add-on für Confluence entwickelt werden, dass E-Mails in Blog-Posts wandelt.

2 Theorie

2.1 Wiki

Ein Wiki (hawaiianisch für schnell[2]) ist eine Webseite, auf der kollaborativ Texte mit Bildern, Zitaten und anderen Medien gelesen und bearbeitet werden können. Der Grundgedanke von Wikis ist in der Regel, dass jeder Artikel verfassen und bearbeiten darf. Diese Regelung erlaubt es Wikis schnell zu wachsen und eine große Abdeckung eines oder mehrerer Themen zu erreichen, kann aber auch zu Qualitätsproblemen, wie veralteten Inhalten, subjektiver Betrachtung und inhaltlichen Fehlern führen. In erfolgreich geführten Wikis gibt es in der Regel eine Gruppe von vertrauenswürdigen und erfahrenen Nutzern, die sich um die Pflege eines Bereichs kümmern. [3]

Das bekannteste Beispiel für ein Wiki ist Wikipedia. Die englischsprachige Wikipedia besteht derzeit aus fast 5,5 Millionen Artikel mit einer durchschnittlichen Länge von 615 Wörtern[4]. Die deutschsprachige Wikipedia besteht aus fast 1,7 Millionen Artikeln mit einer durchschnittlichen Länge von 583 Wörtern pro Artikel[5]. Wikipedia ist derzeit die meistgenutzte Webseite der Welt nach 2 Suchmaschinen, Youtube und Facebook[6].

Neben im Internet frei verfügbaren Wikis, wie Wikipedia, werden Wikis auch für die Dokumentation innerhalb von Firmen verwendet. Wikis erlauben es das verstreute Wissen der gesamten Mitarbeiter zu bündeln und Abläufe und Projekte zu dokumentieren. Die flache Hierarchie in Wikis erlaubt eine unkomplizierte Sammlung von informellem Wissen und Notizen. Ein gepflegtes und umfangreiches Firmenwiki kann die Anzahl der persönlichen und schriftlichen Rückfragen drastisch reduzieren und die Produktivität erhöhen. [7]

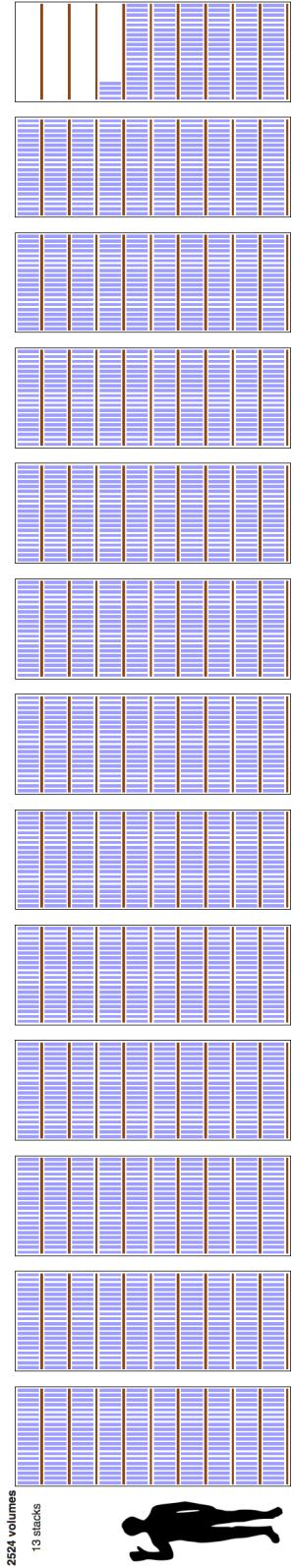


Abbildung 2.1: Hypothetische Größe der englischsprachigen Wikipedia in ausgedruckter Form

Stand 05.09.17, ausschließlich Text, Referenzgröße: Encyclopaedia Britannica

© https://en.wikipedia.org/wiki/Wikipedia:Size_in_volumes
/ GPLv2 (<https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>)

2.2 Confluence

Confluence ist ein Wiki (siehe Kapitel 2.1) der Firma Atlassian, das primär als Firmenwiki konzipiert ist. Derzeit nutzen über 35.000 Firmen Confluence und es wurden insgesamt über 100 Millionen Seiten in Confluence-Systemen angelegt[8]. Confluence ist in Java geschrieben und legt einen Fokus auf die technische Dokumentation von Softwareprojekten[9]. Es integriert sich gut mit den anderen Entwicklertools von Atlassian, wie Jira (Ticket-System), Bitbucket (Software-Verwaltung) und Crucible (Software-Reviewing)[10]. Die aktuelle Confluence-Version ist 6.3 (veröffentlicht am 12.07.17)[11].

2.2.1 Seiten und Blog-Posts

Seiten und Blog-Posts sind in der Confluence Dokumentation[12] beschrieben. Die Informationen in diesem Abschnitt richten sich nach dieser Quelle.

Confluence kennt zwei primäre Konzepte um Inhalte abzubilden: Seiten und Blog-Posts. Sowohl eine Seite als auch ein Blog-Post besteht aus von Usern gepflegtem und formatiertem Text, mit eingebetteten Bildern, Videos und anderen Medien, der mit Hilfe eines What You See Is What You Get (WYSIWYG)-Editor bearbeitet werden kann. Beide Inhaltstypen unterstützen zusätzlich Datei-Anhänge. Beiträge können über Verlinkungen miteinander verbunden werden.

Seiten dokumentieren in der Regel Zustände, können nachträglich aktualisiert werden und sind in einer hierarchischen Struktur angeordnet. Blog-Beiträge sind zu einem Datum aktuelle Mitteilungen und werden in einem Blog chronologisch sortiert. Seiten sind quasi die moderne Variante eines Aktenordners, während ein Blog eher die Form eines Logbuch oder eines Tagebuchs annimmt.

Theorie – Confluence

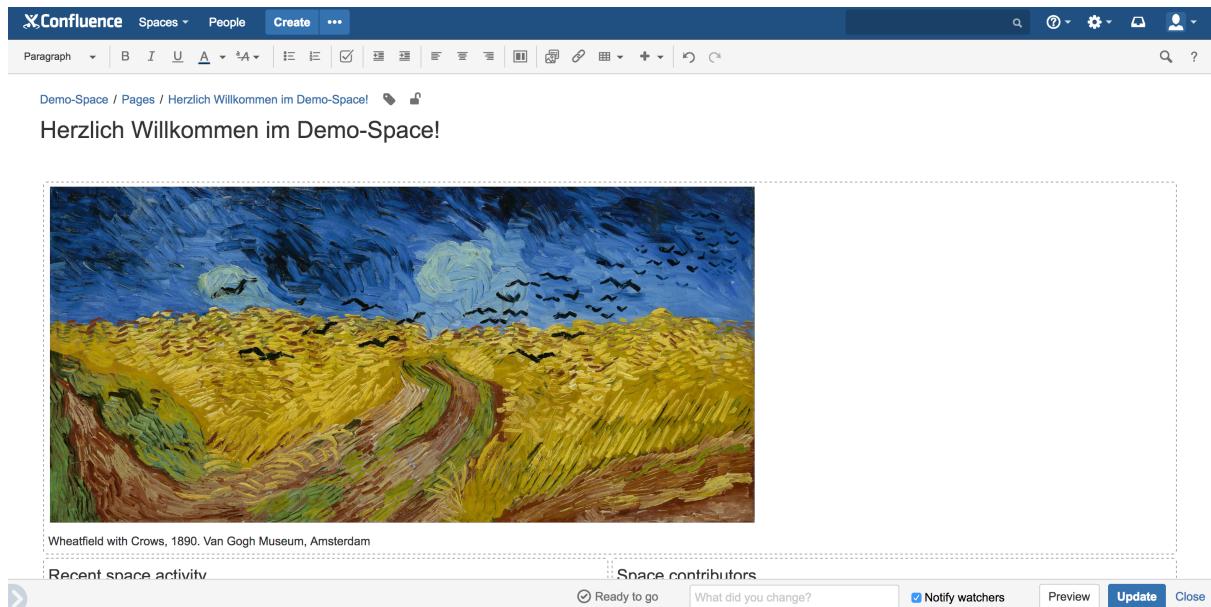


Abbildung 2.2: Confluence WYSIWYG-Editor

A screenshot of the Confluence blog history page. It shows a sidebar with a "BLOG" icon and a "RSS" icon. Below that, a tree view of blog posts: "2017" has a child node "August" which contains three entries: "Welcome to Confluence", "Passed: dm-drogeriemarkt/M...", and "Passed: dm-drogeriemarkt/M...".

Abbildung 2.3: Confluence Blog-Historie

A screenshot of the Confluence page tree. On the left, it says "PAGE TREE". To the right, there's a hierarchical list: "Agile Coaches - Scrum Master Space" has a child node "Anleitungsartikel". "MEC-*-SE" has children "Amazon Web Services", "Developer Days", "dmTech Talks", and "Docker-Container und -Deployments-Primer".

Abbildung 2.4: Confluence Page-Tree

2.2.2 Spaces

Spaces sind in der Confluence Dokumentation[13] beschrieben. Die Informationen in diesem Abschnitt richten sich nach dieser Quelle.

Seiten und Blog-Einträge sind in Spaces (Bereichen) angeordnet. Globale Spaces sind für alle verfügbar und durchsuchbar. Jeder Nutzer kann einen persönlichen Space anlegen, trotz des Namens sind die Inhalte unterhalb persönlicher Spaces standardmäßig für alle sichtbar und durchsuchbar.

Theorie – Confluence

Jeder Space besitzt eine gestaltbare Home-Seite, die beim Besuchen des Spaces angezeigt wird. Einen Seiten-Baum, der die Seiten-Hierarchie darstellt, und einem Blog-Bereich in dem Blog-Posts angezeigt werden.

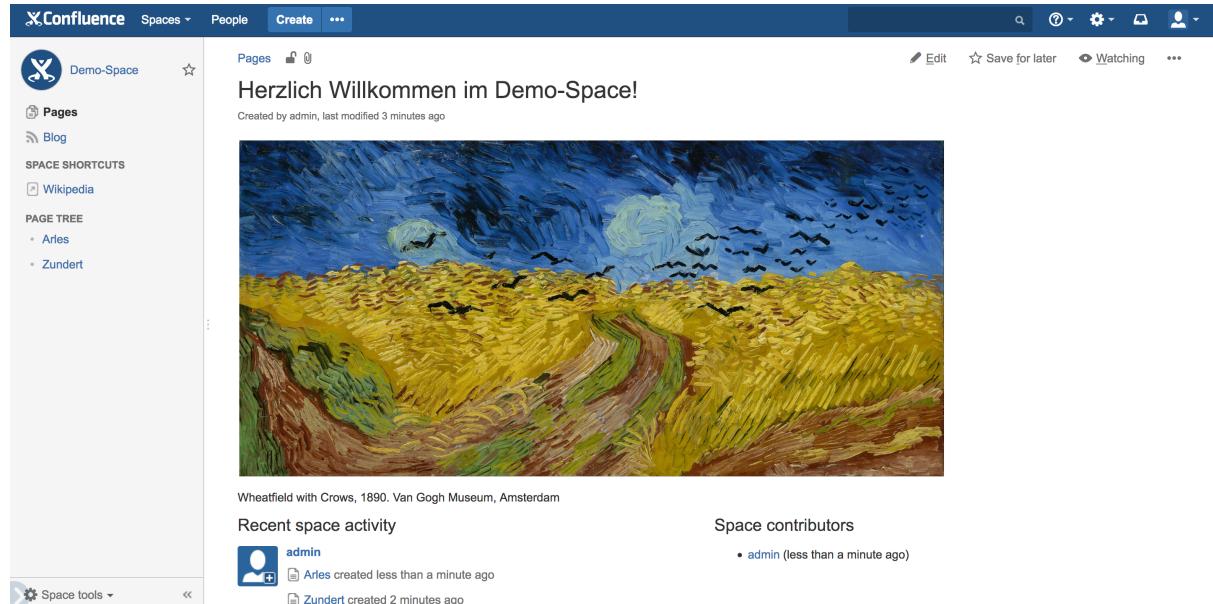


Abbildung 2.5: Confluence Space

2.2.3 Speicher-Format

Confluence speichert den Inhalt von Seite und Blog-Beiträgen in einem Extensible Hypertext Markup Language (XHTML)-Dialekt. Die Basis-Formatierung ist dem XHTML-Standard entliehen, für spezielle Elemente, wie z. B. To-do-Listen, interne Verlinkungen, etc., wurden neue Tags eingefügt die mit ac: anfangen. Komplexe Elemente, wie z. B. Codeblöcke, Diagramme, etc., können als Makros eingefügt werden. [14]

Vor Confluence 4 wurde WikiMarkup als Speicherformat verwendet. Zum Bearbeiten von Inhalten gab es einen WYSIWYG-Editor, der intern mit Rich Text Format (RTF) arbeitete und einem Quellcode-Editor. Die Umwandlung von RTF in WikiMarkup stellte sich als fehlerträchtig heraus, es kam zu Darstellungsfehlern im Editor und in der Anzeige. Seit dem Wechsel auf das neue Speicherformat wird TinyMCE als Hypertext Markup

Theorie – Confluence

Language (HTML)-WYSIWYG-Editor verwendet, sodass jetzt Editor, Speicherformat und die Anzeige alle mit XHTML arbeiten. [15][16]

Listing 2.1: Confluence Speicher-Format

```
1 <p>
2   <ac:image ac:height="250">
3     <ri:attachment ri:filename="Still-Life-with-Absinthe.jpg" />
4   </ac:image>
5 </p>
6 <p>
7   Lorem<strong>ipsum</strong> dolor <s>sit</s> <em>amet</em>,
8   <u>consectetur</u> adipiscing <sub>elit</sub>.
9 </p>
10 <ul>
11   <li>Alpha</li>
12   <li>Bravo</li>
13 </ul>
14 <ac:task-list>
15   <ac:task>
16     <ac:task-id>8</ac:task-id>
17     <ac:task-status>incomplete</ac:task-status>
18     <ac:task-body>Task1</ac:task-body>
19   </ac:task>
20   <ac:task>
21     <ac:task-id>9</ac:task-id>
22     <ac:task-status>incomplete</ac:task-status>
23     <ac:task-body>Task2</ac:task-body>
24   </ac:task>
25 </ac:task-list>
26 <ac:structured-macro
27   ac:name="code"
28   ac:schema-version="1"
29   ac:macro-id="4b291ec9-cf5d-44df-be6f-10a086ad3564"
30 >
31   <ac:parameter ac:name="language">bash</ac:parameter>
32   <ac:plain-text-body>
33     <![CDATA[echo "Hello World"]]>
34   </ac:plain-text-body>
35 </ac:structured-macro>
36 <p class="auto-cursor-target"><br /></p>
```

2.2.4 Add-ons

Confluence kann um zusätzliche Funktionen mit Hilfe von Add-ons erweitert werden. Add-ons werden über den Atlassian-Marketplace vertrieben. Derzeit gibt es über 800 Add-ons für Confluence[17].

Ein Confluence Add-on ist eine JAR-Datei und besteht aus einer Konfigurationsdatei (atlassian-plugin.xml) und alle für die Ausführung nötigen Java-Klassen und Ressourcen[18]. Die Add-ons werden mit dem Build-Management-Tool Maven gebaut, für die Entwicklung stellt Atlassian ein Software Development Kit (SDK) bereit, dass Maven und alle benötigten Tools bereitstellt[19]. Der Maven-Build wird, wie bei allen Maven-Projekten über die Konfigurationsdatei pom.xml gesteuert[20]. Ein einfaches Beispiel für ein komplettes Add-on findet sich in Kapitel 2.2.4.2.1.

Die atlassian-plugin.xml enthält Meta-Informationen über das Add-on und dessen Entwickler. Diese Information wird im Administrations-Interface und im Marketplace angezeigt. Folgendes kann festgehalten werden:[21]

- Ein eindeutiger Schlüssel, der das Add-on identifiziert und der wie ein Java-Paketname aufgebaut ist.
- Ein für Menschen lesbarer Name für das Add-on
- Beschreibender Text zum Add-on
- Versions-Nummer des Add-on
- Name der Firma/Person, die das Add-on verkauft/entwickelt
- Url zur Webseite der Firma/Person, die das Add-on verkauft/entwickelt
- Ein Bild/Logo des Add-ons

Listing 2.2: atlassian-plugin.xml – Meta-Informationen

```
1 <atlassian-plugin key="org.example.myaddon" name="MyAddon"
2   → plugins-version="2">
3   <plugin-info>
4     <description>Beispiel Add-On</description>
5     <version>1.0</version>
6     <vendor name="Beispiel GmbH" url="http://example.org" />
7     <param name="plugin-icon">images/pluginIcon.png</param>
8     <param name="plugin-logo">images/pluginLogo.png</param>
9   </plugin-info>
  </atlassian-plugin>
```

Neben Meta-Informationen werden in der atlassian-plugin.xml zusätzliche Module genutzt, um z. B. Abhängigkeiten zu anderen Confluence-Komponenten oder Add-Ons zu definieren, zeitgesteuerte Jobs einzurichten (siehe Kapitel 2.2.4.1) oder um Konfigurationsseiten anzulegen (siehe Kapitel 2.2.4.2).

2.2.4.1 Zeitgesteuerte Jobs

Das Job-Config-Modul kann in der atlassian-plugin.xml verwendet werden, um periodisch auszuführende Jobs einzurichten. Der eigentliche Job wird von einer Java-Klasse ausgeführt, die das Java-Interface JobRunner (mit der Methode runJob()) implementiert, diese Klasse muss in die atlassian-plugin.xml eingetragen werden. Das Intervall indem der Job ausgeführt wird, kann in Sekunden oder als Cron-Ausdruck angegeben werden. Jobs werden automatisch von Confluence ausgeführt oder können von Administratoren im Administrations-Interface manuell gestartet werden[22].

Theorie – Confluence

Listing 2.3: atlassian-plugin.xml – Job-Modul

```
1 <component key="myJobRunner" class="my.myJobRunner"/>
2
3 <job-config name="My job" key="myJobId">
4   <job key="myJobRunner" perClusterJob="true" clusteredOnly="true"/>
5   <schedule cron-expression="0 * * * * ?" jitterSecs="10"/>
6   <managed editable="true" keepingHistory="true" canRunAdhoc="true"
    ↳ canDisable="true"/>
7 </job-config>
```

Listing 2.4: Java-Klasse – Job

```
1 package my;
2
3 import com.atlassian.scheduler.JobRunnerRequest;
4 import com.atlassian.scheduler.JobRunnerResponse;
5 import com.atlassian.scheduler.JobRunner
6
7 public class myJobRunner implements JobRunner {
8   public JobRunnerResponse runJob(JobRunnerRequest jobRunnerRequest) {
9     // Code hier einfügen
10    return JobRunnerResponse.success();
11  }
12 }
```

2.2.4.2 Konfigurations-Seiten

Häufig bieten Add-Ons Konfigurationswebseiten an, auf denen Administratoren Einstellungen vornehmen können. Dabei handelt es sich um einen klassischen Model View Controller (MVC)-Anwendungsfall: Ein Model speichert die Add-On Daten, z. B. in einer Datenbank und ein Web-Template (View) zeigt diese Daten dem Nutzer an und bietet ein Formular um die Daten zu bearbeiten; eine Kontroller-Klasse vermittelt zwischen Model und View, bereitet Daten für die Anzeige vor, validiert Nutzereingaben und trifft Entscheidungen[23].

In Confluence-Add-Ons werden MVC-Seiten mit Hilfe des Xwork/WebWork Framework und der Web-Template-Sprache Velocity realisiert[24]. Velocity wird von der Apache

Theorie – Confluence

Software Foundation entwickelt und unterstützt Variablen, Konditionen und Schleifen[25]. WebWork wurde 2007 mit dem Apache Struts Projekts der Apache Software Foundation vereinigt und ist seit dem Teil von Struts2[26]. Ein komplettes Beispiel für eine einfache MVC-Anwendung in einem Confluence-Add-On befindet sich im anschließenden Abschnitt 2.2.4.2.1.

Das Xwork-Modul kann in der atlassian-plugin.xml verwendet werden um einzelnen Webseiten zu registrieren. Dabei können mehrere Seiten zu einem Package mit gemeinsamer Basis-Url (Namespace) innerhalb eines Xwork-Containers zusammengefasst werden. Eine Seite wird als Action mit einem Bezeichner (Key) und der Klasse des Kontrollers angelegt. Die Url der Seite ergibt sich aus der Confluence-Basis-Url, dem Namespace des Packages und dem Key der Action gefolgt von der Endung .action. Lautet die Confluence-Basis-Url z. B. <http://confluence.example.org>, der Namespace des Packages plugins/myaddon und der Key der Action myaction, dann ergibt sich daraus die URL <http://confluence.example.org/plugins/myaddon/myaction.action>. [27]

Jeder Action wird in der atlassian-plugin.xml eine Kontroller-Klasse zugewiesen. Die Kontroller-Klasse muss die Klasse ConfluenceActionSupport erweitern. Zusätzlich kann die Methode am Kontroller angegeben werden die aufgerufen werden soll, ansonsten wird beim ersten Besuchen der Seite die Methode doDefault() ausgelöst und nachdem Daten übermittelt wurden, die Methode execute(); diese Methoden geben einen String zurück, für den in der atlassian-plugin.xml verschiedene Velocity-Templates angegeben werden können mit denen die Ausgabe erzeugt wird. [27]

Seit Confluence 4.0 werden in Velocity-Templates enthaltene Variablen automatisch vor der Ausgabe encodiert[28]. Velocity-Variablen werden automatisch in die passenden Getter-/Setter der Kontroller-Klasse übersetzt. Beim lesen der Velocity-Variable \$configuration.test wird z. B. getConfiguration.getTest() aufgerufen. Nach Absenden des Formulars wird für alle im Formular enthaltene Variablen der passende Setter aufgerufen, für configuration.test=123 wird z. B. getConfiguration().setTest(123) aufgerufen. Um zu verhindern, dass auf diese Weise ein Angreifer über Getter einer Kontroller-Klasse Zugriff auf sicherheitskritische Objekte erlangt, wurde die Java-Annotation @ParameterSafe eingeführt, mit der alle Klasse, die auf diese Art und Weise bearbeitbar sein sollen, markiert werden müssen. [29][30]

Theorie – Confluence

2.2.4.2.1 Beispiel Add-On

Die folgende minimale Beispiel-Anwendung fragt den Nutzer unter dem Pfad /admin/plugins/myaddon/age.action nach dem Alter und prüft, ob das eingegebene Alter unter oder über 18 Jahren ist. Anschließend wird dem Nutzer das Ergebnis der Prüfung mitgeteilt.

Abbildung 2.6: Beispiel Add-On – Dateibaum

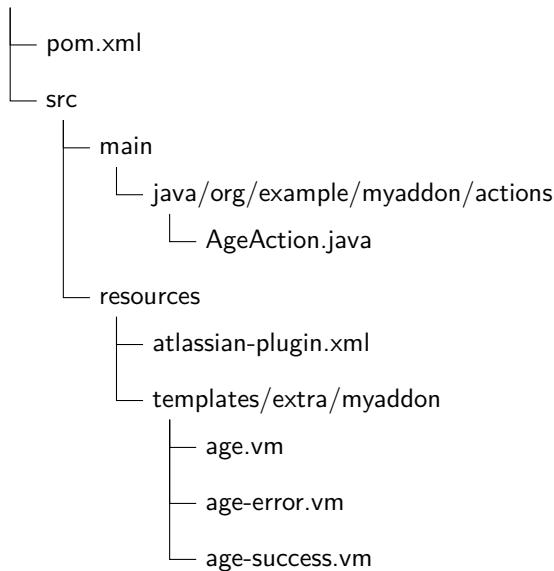
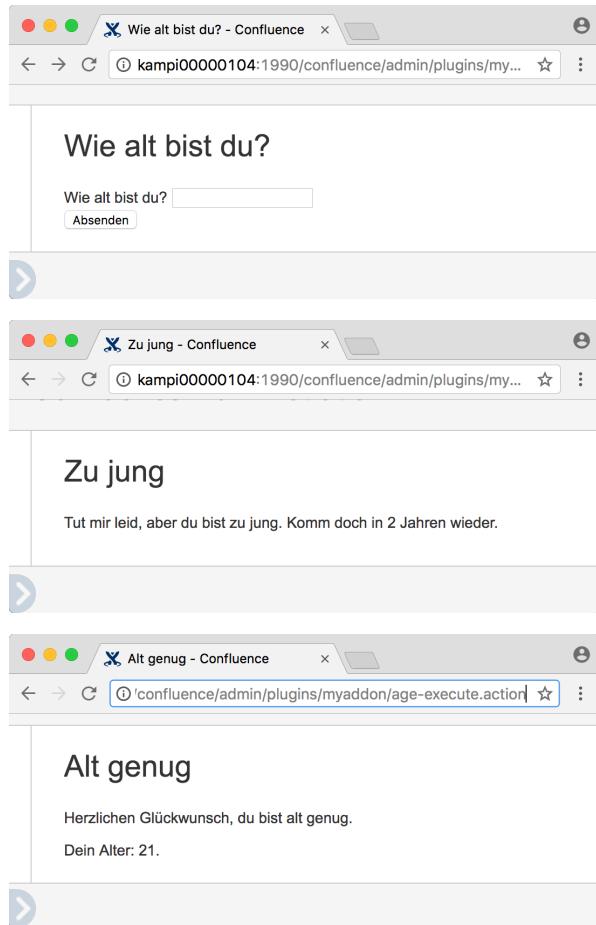


Abbildung 2.7: Beispiel Add-On – UI



Theorie – Confluence

Listing 2.5: Beispiel Add-On – atlassian-plugin.xml

```
1 <atlassian-plugin key="org.example.myaddon" name="MyAddon" plugins-version="2">
2   <plugin-info>
3     <description>Beispiel Add-On</description>
4     <version>1.0</version>
5     <vendor name="Beispiel GmbH" url="http://example.org" />
6     <param name="plugin-icon">images/pluginIcon.png</param>
7     <param name="plugin-logo">images/pluginLogo.png</param>
8   </plugin-info>
9
10  <web-item key="myaddon-alter-link" name="MyAddOn-Alter"
11    ↳ section="system.admin/configuration" weight="1000">
12    <description>MyAddon - Alter</description>
13    <label>MyAddOn - Alter</label>
14    <link linkId="devhelper-admin-link">/admin/plugins/myaddon/age.action</link>
15  </web-item>
16
16  <xwork name="MyAddon UI" key="myaddon">
17    <description>User-Interface von myAddon</description>
18
19    <package name="myaddon" extends="default" namespace="/admin/plugins/myaddon">
20      <default-interceptor-ref name="defaultStack" />
21
22      <action name="age" class="org.example.myaddon.actions.AgeAction" method="input">
23        <result name="input" type="velocity">/templates/extra/myaddon/age.vm</result>
24        <result name="success"
25          ↳ type="velocity">/templates/extra/myaddon/age-success.vm</result>
26        <result name="error"
27          ↳ type="velocity">/templates/extra/myaddon/age-error.vm</result>
28      </action>
29
30      <action name="age-execute" class="org.example.myaddon.actions.AgeAction"
31        ↳ method="execute">
32        <result name="success"
33          ↳ type="velocity">/templates/extra/myaddon/age-success.vm</result>
34        <result name="error"
          ↳ type="velocity">/templates/extra/myaddon/age-error.vm</result>
35      </action>
36    </package>
37  </xwork>
38 </atlassian-plugin>
```

Theorie – Confluence

Listing 2.6: Beispiel Add-On – Java-Klasse – Xwork-Kontroller

```
1 package org.example.myaddon.actions;
2
3 import com.atlassian.confluence.core.ConfluenceActionSupport;
4
5 public class AgeAction extends ConfluenceActionSupport {
6     private int age = 0;
7     private String state = "input";
8
9     public String input() {
10         return "input";
11     }
12
13     public String execute() {
14         if (age < 18) {
15             return "error";
16         } else {
17             return "success";
18         }
19     }
20
21     public void setAge(int age) {
22         this.age = age;
23     }
24
25     public int getAge() {
26         return age;
27     }
28 }
```

Theorie – Confluence

Listing 2.7: Beispiel Add-On – Velocity-Template – age.vm

```
1 <html>
2   <head><title>Wie alt bist du?</title></head>
3   <body>
4     <form class="aui" name="age_form" method="POST"
5       ↳ action="age-execute.action">
6       <p>Wie alt bist du? <input min="0" type="number" name="age" /></p>
7       <input type="submit" value="Absenden" />
8       #form_xsrfToken()
9     </form>
10    </body>
11  </html>
```

Listing 2.8: Beispiel Add-On – Velocity-Template – age-success.vm

```
1 <html>
2   <head><title>Alt genug</title></head>
3   <body>
4     <p>Herzlichen Glückwunsch, du bist alt genug.</p>
5     <p>Dein Alter: $age. </p>
6   </body>
7 </html>
```

Listing 2.9: Beispiel Add-On – Velocity-Template – age-error.vm

```
1 <html>
2   <head><title>Zu jung</title></head>
3   <body>
4     #set($wait = 18 - $age)
5     <p>Tut mir leid, aber du bist zu jung. Komm doch in $wait Jahren
6       ↳ wieder.<p>
7   </body>
8 </html>
```

2.3 E-Mail

E-Mail ist in den Zeiten des ARPANET (dem Vorläufer des Internet) entstanden und war lange der meist genutzte Dienst des Internets[31]. Nach wie vor wächst die Zahl der versendeten E-Mails, 2016 wurden in Deutschland 625,8 Milliarden E-Mails (ohne SPAM) versendet[32].

Dieses Kapitel beschäftigt sich mit den Protokollen und Formaten, die relevant für die Aufgabe sind.

2.3.1 E-Mail-Adresse

Eine E-Mail-Adresse identifiziert einen Empfänger/Sender eindeutig. Sie setzt sich zusammen aus einem lokalen Teil (in der Regel der Name des Postfachs) und einem Domainnamen getrennt durch „@“. Beim Senden einer Nachricht stellt der Absender eine Domain Name System (DNS)-Abfrage für den Domainnamen und sendet die E-Mail an den im MX-Record angegebenen Server, der dann den lokalen Teil einer Mailbox zuordnet und die E-Mail ausliefert. [33]

Um unerreichbare E-Mail-Adressen auf Mailinglisten zu identifizieren, wurde das Variable envelope return path (VERP)-Verfahren entwickelt. Dabei werden für verschiedene Empfänger individuelle Absenderadressen verwendet, sodass eingehende Fehler-Meldungen eindeutig einem Empfänger zuzuordnen sind (Meldungen sind nicht standardisiert)[34]. Damit alle Nachrichten in einem Postfach landen, unterstützen einige E-Mail-Provider ein mit „+“ abgetrenntes Suffix im lokalen Teil der E-Mail-Adresse; so werden z. B. Nachrichten an bob+VERP@gmail.com in das gleiche Postfach wie bob@gmail.com geliefert[35].

2.3.2 Format

Das Format einer E-Mail ist in RFC 5322 definiert[33]. Die Informationen in diesem Abschnitt richten sich nach dieser Quelle.

Eine E-Mail besteht aus einem Kopf (Header) und einem Nachrichten-Körper (Body). Im Header befinden sich Meta-Informationen (Betreff, Datum, Inhaltstyp, Encodierung, etc.) und Informationen zum Absender und Empfänger. Üblicherweise hinterlässt jeder Mailserver, den die Nachricht durchläuft, seinen Hostnamen, seine IP-Adresse und einen Zeitstempel im Header, damit die Route, die eine E-Mail nimmt, nachvollzogen werden kann. Im Body befindet sich die eigentlich Nachricht bestehend aus 7 Bit breiten American Standard Code for Information Interchange (ASCII)-Zeichen. Header und Body werden durch eine Leerzeile (\CR\LF\CR\LF) voneinander getrennt.

Jede Information im Header besteht aus einem Feldnamen und einem Wert, die durch einen Doppelpunkt getrennt in einer Zeile stehen. Die, nach Auffassung des Autors, wichtigsten Felder sind:

To Der Empfänger, z. B. bob@example.org. Neben der Adresse kann auch zusätzlich der Name angegeben werden, z. B. Bob <bob@example.org>. Mehrere Einträge können mit Komma getrennt werden.

CC Eine Liste von Empfängern die eine Kopie der E-Mail erhalten sollen, an die die Nachricht aber nicht direkt gerichtet ist. Das Format ist das gleiche wie bei To.

From Der Absender. Z. B. Alice <alice@example.org>.

Subject Betreff der E-Mail.

Date Datum (und optional Zeit) an dem die E-Mail versendet wurde.

Message-Id Global eindeutige ID, die eine Nachricht identifiziert.

Listing 2.10: Beispiel für eine Email

```
1 From: "Alice" <alice@example.org>
2 Subject: Hallo Welt
3 Message-ID: <DBFA944D-CFFA-445E-B5B9-853319F4E093@example.org>
4 Date: Wed, 23 Aug 2017 11:02:11 +0200
5 To: Bob <bob@example.org>, Charlie <charlie@example.org>
6 CC: Dave <dave@example.org>
7
8 Hallo Welt
```

2.3.3 Multipurpose Internet Mail Extensions

Der MIME-Standard ist in den RFCs 2046[36], 2047[37], 4289[38] und 2049[39] definiert. Die Informationen in diesem Abschnitt richten sich nach diesen Quellen.

Der MIME-Standard erweitert E-Mails um die Möglichkeit mehr als nur 7 Bit breiten ASCII-Text zu transportieren. Dabei ist auch der Transport von mehreren zusammenhängenden Dateien möglich. Eine Mime-Mail wird mit dem Header MIME-Version: 1.0 markiert.

Um den Inhaltstyp einer Nachricht zu klassifizieren, wurde der Header Content-Type eingeführt. Der Inhaltstyp wird als Mime-Type angegeben. Mime-Types werden zentral vergeben und geben E-Mail-Readern Aufschluss darüber, wie sie mit einer Datei umgehen sollen. Ein Mime-Type besteht aus einem Haupttyp und einem Untertyp, die mit „/“ voneinander getrennt werden. Die Haupttypen sind, neben den zwei verfahrenstechnisch verwendeten Typen multipart und message:

text Für unformatierten und formatierten Text. Z. B. text/plain für Plain-Text oder text/html für HTML.

image Für Bilddateien. Z. B. image/jpeg für Jpeg-Bilder.

audio Für Audiodateien. Z. B. audio/mpeg für Mpeg-Audio-Dateien.

video Für Videodateien. Z. B. video/MP2P für Mpeg2-Video-Dateien.

application Für Anwendungsdateien. Z B. application/pdf für PDF-Dateien. Der Mime-Type application/octet-stream kann generisch für alles verwendet werden, was keinem Mime-Type zugeordnet ist und wird von E-Mail-Readern in der Regel als Anhang dargestellt.

Da 7 Bit breiter ASCII-Text ausschließlich für englischen Klartext geeignet ist, nicht aber für andere Sprachen oder Binärdaten, wurden verschiedene Encodierungen eingeführt, die im Header Content-Transfer-Encoding angegeben werden:

7bit Der ursprüngliche Standard. Keine Codierung findet statt. Die Daten bestehen aus 7 Bit breiten ASCII-Zeichen.

QUOTED-PRINTABLE Alle Bytes, die nicht den druckbaren ASCII-Zeichen entsprechen, werden durch = gefolgt vom Hex-Wert des Zeichens ersetzt.

BASE64 Immer 6 Bit der Daten werden durch ein ASCII-Zeichen dargestellt. Vor allen Dingen bei Binärdaten nützlich. Durch den Codierungsvorgang wächst die Länge auf $\sim 4/3$ der ursprünglichen Länge an.

8Bit Keine Codierung findet statt. Potenziell problematisch, da nicht sichergestellt ist, dass alle vermittelnde Stationen in der Lage sind 8 Bit breite Daten zu verarbeiten, der ursprüngliche Standard sieht nur 7 Bit vor (siehe Kapitel 2.3.2). Wenn eine E-Mail z. B. über eine, auf 7 Bit breite Zeichen konfigurierte, serielle Schnittstelle gesendet wird, werden alle nicht ASCII konformen Zeichen falsch übertragen.

2.3.3.1 Multipart-Nachrichten

Der Mime-Standard eröffnet die Möglichkeit mehrere Dateien/Dokumente in einer E-Mail zu transportieren. Dazu wurden die multipart/* Mime-Types eingeführt. Die einzelnen Teile werden dabei von einer Begrenzer Zeichenkette getrennt und bestehen ihrerseits wieder aus einem Header und einem Body. Der Begrenzer wird dem Mime-Type im Content-Type-Header angehängt. Dabei können die einzelnen Teile unter Verwendung der verschiedenen Multipart-MIME-Types hierarchisch angeordnet werden. Die Multipart-Mime-Types lauten:

Theorie – E-Mail

multipart/mixed Die einzelnen untergeordneten Teile sollen unabhängig voneinander im Mail-Reader oder als Anhang dargestellt werden.

multipart/digest Wird verwendet um mehrere E-Mails zu einer zusammenzufassen.

multipart/alternative Der E-Mail-Reader soll zwischen den untergeordneten Teilen einen wählen. In den meisten Fällen kann der Reader zwischen einer Text- und einer Html-Version wählen.

multipart/related Die untergeordneten Teile ergeben nur zusammen Sinn. Z. B. kann ein HTML-Dokument auf Bilder aus einem der anderen Teile verweisen. Besonders relevant ist dieser Mime-Type in der Kombination mit einem übergeordneten Teil mit dem Mime-Type multipart/alternative.

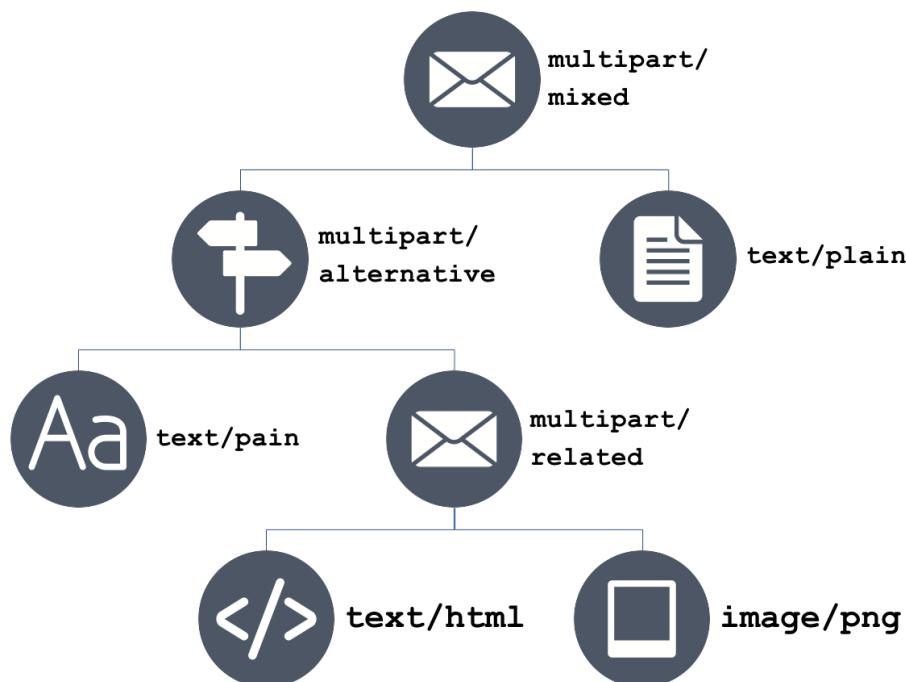


Abbildung 2.8: Beispiel für die Hierarchie einer Multipart-Nachricht
Icons © Elegant Themes / <https://www.elegantthemes.com/blog/freebie-of-the-week/beautiful-flat-icons-for-free>
/ GPLv2 (<https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>)

Theorie – E-Mail

Listing 2.11: Beispiel für eine Multipart-E-Mail

```
1 Subject: Beispiel für eine Multipart-Nachricht
2 MIME-Version: 1.0
3 From: "Alice" <alice@example.org>
4 To: Bob <bob@example.org>
5 Message-ID: <>132a9252-7ace-4e3d-91f6-11cc71f32cf4@example.org>
6 Date: Wed, 23 Aug 2017 15:09:03 +0200
7 Content-Type: multipart/mixed; boundary=LEVEL0_BOUNDARY
8
9 --LEVEL0_BOUNDARY
10 Content-Type: multipart/alternative; boundary=LEVEL1_BOUNDARY
11
12 --LEVEL1_BOUNDARY
13 Content-Type: text/plain; charset="utf-8"
14 Content-Transfer-Encoding: 8bit
15
16 Zwölf große Boxkämpfer jagen Viktor quer über den Sylter Deich
17 --LEVEL1_BOUNDARY
18 Content-Type: multipart/related; boundary=LEVEL2_BOUNDARY
19
20 --LEVEL2_BOUNDARY
21 Content-Type: text/html; charset="utf-8"
22 Content-Transfer-Encoding: QUOTED-PRINTABLE
23
24 <html><body>
25     <h1>Zw=F6lf gro=Df Fe Boxk=E4mpfer jagen Viktor ...</h1>
26     
27 </body></html>
28 --LEVEL2_BOUNDARY
29 Content-Type: image/png
30 Content-Disposition: inline
31 Content-Transfer-Encoding: base64
32 Content-ID: <image>
33
34 iVBORw0KGgoAAAANSUhEUgAAAAAMAAAADCAIAAADZSiLoAAAACXBIWXMAAAsTAAAL
35 EwEApmpwYAAAAB3RJTUUH4QgXEA8EcKE9xgAAABxJREFUCNc9yKEBAAAIwKDp/z9j
36 k0gKaqOZ608A048N9+ctgaEAAAASUVORK5CYII=
37
38 --LEVEL2_BOUNDARY--
39 --LEVEL1_BOUNDARY--
40 --LEVEL0_BOUNDARY
41 Content-Type: text/plain; charset="ascii"; name="test.txt"
42 Content-Transfer-Encoding: 7bit
43 Content-Disposition: attachment; filename="test.txt"
44
45 Anhang
46 --LEVEL0_BOUNDARY--
```

2.3.4 IMAP and POP3

Derzeit gibt es zwei verbreitete offene Protokolle um E-Mails von einem E-Mail-Server abzurufen: Das Post Office Protocol (POP) in der Version 3, das in RFC 1939[40] beschrieben ist, und das Internet Message Access Protocol (IMAP) in der Version 4, das in RFC 1730[41] beschrieben ist. Die Informationen aus diesem Absatz richten sich nach diesen Quellen.

POP-Clients holen E-Mail, über eine TCP-Verbindung (Port 110), vom Server ab und speichern sie lokal, in der Regel löschen sie die Nachrichten danach vom Server. Mehrere Clients synchronisieren sich nicht untereinander. Eine E-Mail, die z. B. auf dem Rechner gelöscht wird, wird nicht auch auf dem Handy gelöscht. Wenn die Mail-Clients so konfiguriert sind, dass sie die Nachrichten nach dem Herunterladen vom Server löschen, dann gibt es ein „Wettrennen“ zwischen den Clients und die Nachrichten kommen nur auf einem Gerät an. POP unterstützt keine Ordner.

IMAP wurde als Alternative zu POP entwickelt und unterstützt auch Ordner. IMAP-Clients synchronisieren, über eine TCP-Verbindung (Port 143), den lokalen Stand mit dem Stand auf dem Server. Wird eine Nachricht auf einem Gerät gelöscht oder verschoben, wird die Änderung auch auf dem Server durchgeführt und bei der nächsten Synchronisation auch auf allen anderen Clients.

Beide Protokolle sind textbasiert und können über eine TLS-Verbindung (siehe Kapitel 2.4.2) auch verschlüsselt genutzt werden. Dabei kann die TLS-Verbindung über einen neuen Port aufgebaut werden POP3s (Port 995) und IMAPs (Port 993), oder mit dem STARTTLS-Kommando eine bestehende Verbindung verschlüsselt werden. Beide Vorgehensweisen sind in RFC 2595[42] beschrieben. Während der Autor des RFC 2595, STARTTLS über POP3s und IMAPs bevorzugt, hat sich gezeigt, dass STARTTLS in der Praxis unsicherer ist, weil ein Man-in-the-Middle-Angrifer (siehe Kapitel 2.4.3) den STARTTLS-Befehl beim Verbindungsauflauf entfernen kann, in der Regel fallen Server und Client dann auf eine Klartextverbindung zurück[43]. 2014 haben z. B. zwei große thailändische Internet Service Provider (ISPs) die E-Mail-Kommunikation ihrer Kunden auf diese Weise aufgebrochen[44].

2.4 Sicherheitsanalyse

Durch die Verwendung von E-Mail und Confluence als Webanwendung entstehen einige potenzielle Sicherheitsprobleme, die in diesem Kapitel erläutert werden.

Das Internet hat sich zu einem Geschäftsfeld für Kriminelle und zu einem signifikanten Sicherheitsproblem für Unternehmen, Regierungen und Privatpersonen entwickelt: McAfee schätzte 2014 den jährlichen globalen finanziellen Schaden durch Internetkriminalität auf mehr als 400 Milliarden US-Dollar und den Verlust für Deutschland auf fast 50 Milliarden Euro[45].

2.4.1 Lauschangriffe

Der einfachste Angriff auf eine Anwendung und die Basis für viele komplexere Angriffe stellt ein einfaches Abhören der Kommunikation zwischen Opfer und Anwendung da. Insbesondere in lokalen Netzwerken werden User Datagram Protocol (UDP)- und Transmission Control Protocol (TCP)-Pakete häufig an alle Teilnehmer versendet, sodass der Angreifer die Kommunikation leicht abhören kann. Viele Anwendungen (z. B. Webanwendungen) und Protokolle (z. B. SMTP, POP3, IMAP) tauschen Passwörter, Tokens und sensible Daten in Klartext aus und ein Lauschangriff stellt eine einfache, verdeckte und effiziente Methode da, ein System zu kompromittieren. [46]

Sogenannte Packet-Sniffer (Sniffing Englisch für Schnüffeln), wie z. B. Wireshark, die eine Vielzahl von gängigen Protokollen unterstützen, machen es Angreifern einfach UDP- und TCP-Datenströmen zu analysieren und relevante Daten zu extrahieren[47].

Theorie – Sicherheitsanalyse

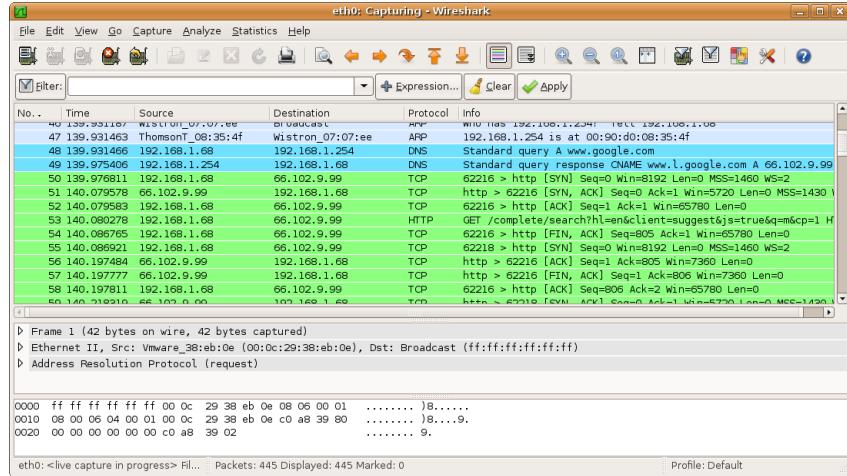


Abbildung 2.9: Der Package-Sniffer Wireshark

© https://upload.wikimedia.org/wikipedia/commons/0/03/Wireshark_screenshot.png
 / GPLv2 (<https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>)

2.4.2 TLS/SSL

Ein verlässlicher Schutz vor Lauschangriffen stellt nur die Ende-zu-Ende-Verschlüsselung der Kommunikation da. Zu diesem Zweck wurde Ende des letzten Jahrtausends der Transport Layer Security (TLS)/Secure Sockets Layer (SSL)-Standard entwickelt. Die eigentliche Verbindung wird dabei mit einem effizienten symmetrischen Verschlüsselungsverfahren (gleicher Schlüssel für die Ver- und Entschlüsselung) verschlüsselt. Dabei wird für jede Verbindung ein neuer Schlüssel zufällig generiert. Der Schlüsselaustausch findet mit einem langsameren asymmetrischen Verschlüsselungsverfahren statt (öffentlicher Schlüssel zur Verschlüsselung, privater Schlüssel zur Entschlüsselung). [48]

Ein Client initiiert zunächst eine unverschlüsselte Verbindung zu einem Server, der Server antwortet unverschlüsselt mit seinem öffentlichen Schlüssel zusammen mit einigen Meta-Informationen über den Dienst (Zertifikat). Der Client generiert einen Schlüssel für die Kommunikation, verschlüsselt ihn mit dem öffentlichen Schlüssel und sendet ihn an den Server. Danach läuft die Kommunikation komplett verschlüsselt ab. [48]

Theorie – Sicherheitsanalyse

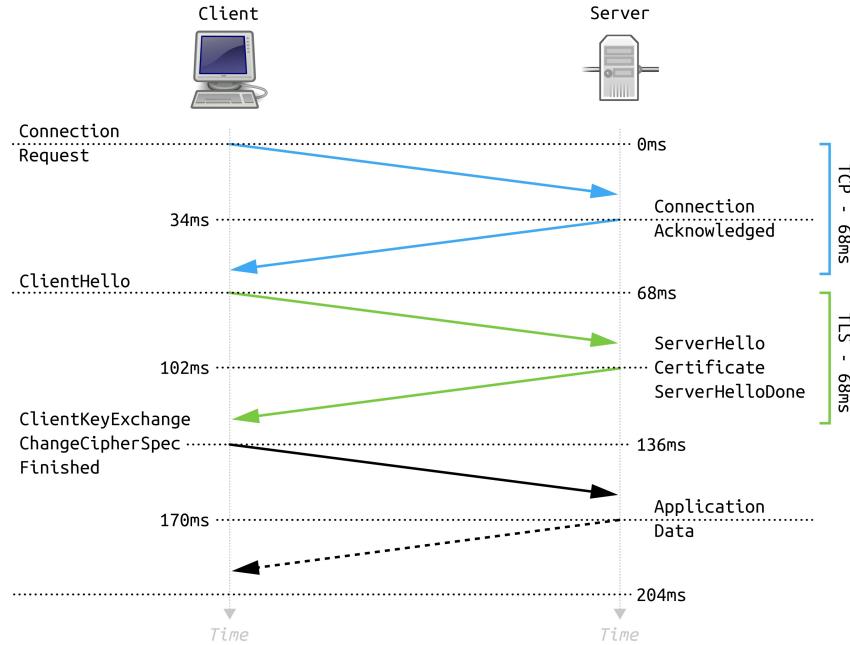


Abbildung 2.10: TLS/SSL Verbindung

© https://commons.wikimedia.org/wiki/File:Abbreviated_TLS_1.2_Handshake.svg / Gemeinfrei

Für einen ausschließlich lauschenden Angreifer ist es nach Erfahrung des Autors praktisch unmöglich den Inhalt einer so verschlüsselten Konversation, zu ermitteln.

Um gegen einen im nächsten Abschnitt beschriebenen MITM-Angriff gewappnet zu sein und die Validität eines Zertifikates zu überprüfen wird das Zertifikat wieder mit einem Zertifikat signiert. Dadurch ergibt sich eine Zertifizierungskette, die mehrere Ebenen aufweisen kann, das oberste Zertifikat signiert sich selbst. Anwendungen und Betriebssysteme führen eine Liste mit vertrauten Zertifikaten, gegen die Zertifikate geprüft werden. Für die interne Verwendung in Unternehmen gibt es häufig eine eigene Stelle die Zertifikate signiert; für die Kommunikation im Internet gibt es allgemein anerkannte Vergabestellen, deren Zertifikate in Anwendungen und Betriebssystemen bereits eingebaut sind. [49]

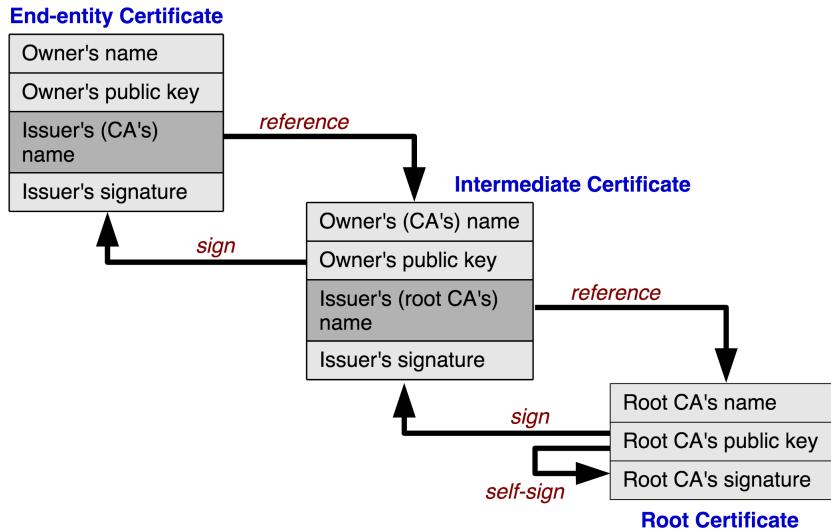


Abbildung 2.11: TLS/SSL Chain-Of-Trust

© Yanpas / https://commons.wikimedia.org/wiki/File:Chain_of_trust.svg
 / CC-BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/legalcode>)

TLS/SSL kann prinzipiell verwendet werden, um jede TCP-Verbindung zu verschlüsseln. Ursprünglich entwickelt wurde es für Hypertext Transfer Protocol Secure (HTTPS), wobei es sich um mit TLS/SSL verschlüsseltes Hypertext Transfer Protocol (HTTP), das Übertragungsprotokoll des WWW, handelt. Aber auch z. B. SMTP, POP3 und IMAP können so verschlüsselt werden. [50]

2.4.3 Man-in-the-Middle-Angriff

Als Man-in-the-Middle-Angriff bezeichnet man einen Angriff, bei dem der Angreifer die Möglichkeit hat, die Kommunikation zwischen zwei Opfern zu lesen und zu modifizieren. Dabei nutzt der Angreifer entweder die Tatsache aus, dass die Kommunikation physikalisch an ihm vorbei läuft (Wlan-Hotspot, Router, ISP, Proxy-Server), oder er lenkt die Kommunikation logisch durch ihn um, indem er z. B. in einem lokalen Netzwerk ARP-Spoofing betreibt oder Antworten auf DNS-Anfragen fälscht. [51]

Ein MITM-Angriff kann in vielen Situationen dazu führen, dass Verschlüsselung wirkungslos wird. Z. B. kann der Angreifer, die für die Kommunikation verwendeten Schlüssel durch

Theorie – Sicherheitsanalyse

eigene Schlüssel ersetzen; daher ist es wichtig, dass die verwendeten Schlüssel von einer vertrauenswürdigen Stelle signiert werden und die Signaturen überprüft werden (siehe Kapitel 2.4.2)[52]. Oder es kann der Wechsel von einer unverschlüsselten Kommunikation auf eine verschlüsselte Kommunikation verhindert werden[53].

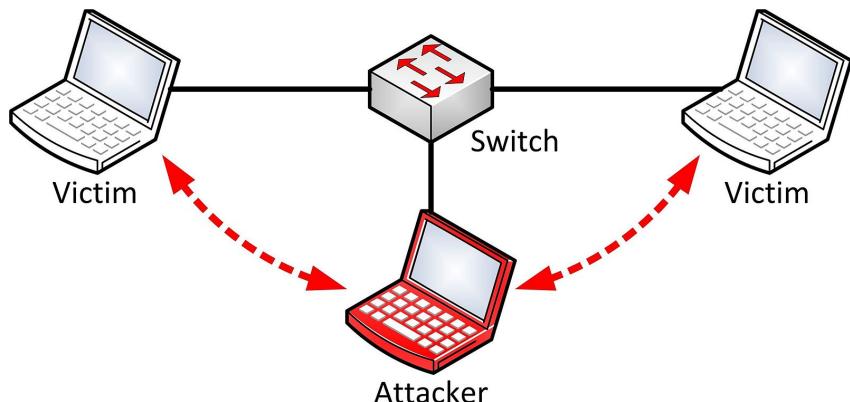


Abbildung 2.12: Man-in-the-Middle-Angriff

© Nasanbuyn / <https://commons.wikimedia.org/wiki/File:%D0%A5%D0%BO%D0%BB%D0%B4%D0%BB%D0%BO%D0%B3%D0%BO.jpg>
/ CC-BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/legalcode>)

2.4.4 Cross-Site-Scripting

Bei Webanwendungen sind neben Angriffen auf die Übertragung der Daten, auch Angriffe auf die eigentliche Anwendung möglich. Eine häufige Angriffsfläche stellen dabei nicht richtig validierte und encodierte Nutzereingaben dar[54]. Exemplarisch wird für diese Art der Angriffe hier Cross-site scripting (XSS) behandelt, da es, nach Meinung des Autors, für die Aufgabe besonders relevant ist.

XSS beschreibt einen Angriff auf eine Webanwendung, bei der der Angreifer eine Sicherheitslücke ausnutzt, um Schadcode permanent oder temporär in eine Webseite einzubauen, sodass er im gleichen Kontext ausgeführt wird. Der Code hat dann die gleichen Rechte wie das Opfer, das die Webseite besucht, und kann z. B. die Sessioncookies des Opfers stehlen und an den Angreifer senden, der sich damit gegenüber der Webanwendung authentifizieren kann. [55]

Theorie – Sicherheitsanalyse

Der Angreifer schleust den Schadcode dabei über Benutzereingaben, die gar nicht oder nicht richtig encodiert werden in die Webseite ein. Wenn die Eingabe z. B. in einem URL-Parameter steht und daher vom Client aufgerufen werden muss, spricht man von reflektiertem XSS, dabei kann der Link-Aufruf auch im Hintergrund z. B. aus einem Html-IMG-Tag heraus erfolgen, ohne dass es für das Opfer ersichtlich ist. Bei persistentem XSS wird der Schadcode dauerhaft in der Webseite gespeichert, z. B. in einem Blog-Post. [56]

Da JavaScript die meistgenutzte und am besten unterstützte client-seitige Sprache ist, verwenden die meisten Angriffe auch JavaScript. XSS funktioniert aber mit allen client-seitigen Scriptssprachen, wie z. B. auch ActionScript und VBScript. [57]

Die Verbreitung des Problems zeigt eine Analyse von WhiteHat Security für das Jahr 2016 auf: In 15.000 Webanwendungen und mehr als 65.000 mobilen Webanwendungen wurde in einem Drittel aller Fälle eine XSS-Lücke gefunden[58]. Acunetix hat im Zeitraum April 2015 bis März 2016, über 61.000 Webseiten untersucht und kommt zu dem gleichen Ergebnis[59].

Wie gravierend eine XSS-Lücke sein kann, zeigt ein Beispiel aus dem Jahr 2005. Eine persistente XSS-Lücke in MySpace machte es möglich, dass der XSS-Wurm Samy innerhalb von 24 Stunden eine Millionen My-Space Profile infizieren konnte, bevor MySpace die Situation unter Kontrolle bekommen hatte. In dem Fall war der Zweck des Wurms harmlos und diente nur dazu den Angreifer bekannt zu machen, genauso gut hätte er aber auch, nach Analyse des Autors, die Zugangsdaten aller Opfer stehlen können. [60]

2.4.5 Phishing

Aufgrund der Popularität die E-Mail genießt (siehe Kapitel 2.3) und der Tatsache, dass ein Angreifer aktiv gezielt sein Opfer anschreiben kann, hat sich E-Mail, nach Analyse des Autors, zu einem beliebten Medium für Angreifer entwickelt. In Deutschland wurden 2016 jeden Tag über 100 Millionen SPAM-E-Mails versendet[61].

Die Ziele von SPAM-Kampagnen sind unterschiedlich und können unter anderem der Installation von Malware (siehe Kapitel 2.4.5.1) dienen, versuchen den Empfänger dazu

Theorie – Sicherheitsanalyse

zu bringen vertrauliche Informationen wie Bankdaten, Passwörter oder Firmeninterna preiszugeben oder das Opfer dazu verleiten dem Täter Geld zu überweisen.

Die Angreifer verfolgen dabei zwar unterschiedliche Ziele, bedienen sich aber der gleichen Techniken des Social Engineerings um die Opfer dazu zu bringen vertrauliche Informationen preiszugeben (Benutzername, Passwörter, Bankinformationen, etc.), einen Anhang zu öffnen oder auf einen Link zu klicken. Dabei gibt sich der Angreifer dem Opfer gegenüber als Person oder Organisation aus, in die das Opfer vertrauen hat, oder tarnt den Inhalt als harmlose Anfrage. E-Mails sehen z. B. aus wie legitime Mitteilung bekannter Banken und Firmen. [62]



Abbildung 2.13: Eine Phishing E-Mail

© <https://upload.wikimedia.org/wikipedia/commons/9/9f/Phishing.gif> / Gemeinfrei

Häufig enthalten E-Mails Links auf nachgebaute Webseiten von Banken, Sozialen Netzwerken oder anderen Organisationen, die Anmelddaten oder andere Informationen abgreifen. Dabei erstellen die Angreifer häufig 1:1 Kopien des HTMLs und Cascading Style Sheets (CSS) echter Webseiten, registrieren seriös wirkende Domains und SSL-Zertifikate (siehe Kapitel 2.4.2), modifizieren die Adresszeile des Browsers mit JavaScript, etc., sodass es für Opfer schwer ist, die Seite als Fälschung zu erkennen. [63][64][65]

Diese Methoden werden als Phishing bezeichnet. Das Wort kommt vom englischen Wort fishing (Fischen), da die Opfer ähnliche wie Fische mit einem Köder gelockt werden und dazu gebracht werden sollen, im Sinne des Angreifers zu handeln. Das ph leitet sich von

Theorie – Sicherheitsanalyse

The screenshot shows a web page with a blue header containing the Deutsche Bank logo. The main content area has a light gray background. At the top left, there's a sidebar with links for 'db OnlineBanking' (including 'Demokonto testen', 'Konto eröffnen', 'Konto für Online- und Telefonbanking freischalten'), 'Hilfe' (including 'Häufig gestellte Fragen', 'BLZ-Suche', 'Download-Center', 'Nutzeranleitung', 'Kontakt', 'Sicherheit', 'Basisinformationen für Vermögensanlagen'), and a 'Herzlich willkommen!' message.

The main form area contains instructions: 'Füllen Sie bitte den Fragebogen für die Bestätigung Ihrer Bankdaten aus. Alle Felder sind Pflichtfelder'. It includes gender selection ('Frau' or 'Herr'), input fields for 'Vorname' and 'Name', and a large text area for entering a TAN code ('Tasten Sie in das gegebene Feld 10 ungenutzte TAN ein (falls es sie weniger übrigblieb, so setzen Sie die bleibenden ein)'). Below this are fields for 'Filiale' (3-stellig), 'Konto' (7-stellig), 'Unterkonto' (2-stellig), and 'PIN' (5-stellig). An 'E-mail' field is also present. A red 'Anmelden' button is at the bottom right.

Abbildung 2.14: Eine Phishing Webseite

© https://commons.wikimedia.org/wiki/File:Fishingwebseite_deutschebank.jpg / Gemeinfrei

phreaking ab, einer Wortneuschöpfung aus phone (Telefon) und breaking (Brechen), die durch frühe Hacking-Angriffe mit Telefonen entstanden ist. [66]

Neben breit angelegten SPAM-Kampagnen, die auf die schiere Masse der potenziellen Opfer zielen, gibt es auch direkt auf ein einzelnes Opfer abgestimmte Angriffe, die als Spear-Pishing bezeichnet werden[67]. So sind z. B. 2016 beim Federal Bureau of Investigation (FBI) Meldungen über 12.005 Fälle eingegangen, bei denen über 360 Millionen US-Dollar, nach Aufforderungen per E-Mail, an Täter überwiesen wurden; dabei haben die Täter sich gezielt Firmen ausgesucht, die häufig große Überweisungen an ausländische Konten durchführen und haben die Anweisungen, mit dem Namen eines Chief executive officer (CEO) als Absender, personalisiert an Mitarbeiter aus den Finanzabteilungen gesendet[68].

2.4.5.1 Malware

Ein Ziel, das die Urheber von Phishing E-Mails verfolgen können, ist die Verbreitung von Malware. Malware ist ein Sammelbegriff für Software, die gegen den Willen des Opfers auf einem Computer installiert wird und im Dienst des Angreifers arbeitet[69].

Der Verizon „Data Breach Investigations Report“ für das Jahr 2017, hat 42.068 erfolgreiche Angriffe, aller Art, auf Unternehmen analysiert und kategorisiert. Die Datenlage wurde von Verizon und über 100 weiteren Netzprovidern, Sicherheitsfirmen und anderen IT-Firmen zusammengetragen. In 51% aller Fälle wurde dabei Malware verwendet, die zu 66% von Mitarbeitern aus E-Mail Anhängen installiert wurde. [70]

Angreifer tarnen die Malware häufig in harmlos wirkenden Dateitypen, um keinen Verdacht bei Opfern und SPAM-Filtern zu erwecken, dabei nutzen sie die Menge an Skriptsprachen und Archivtypen, die Windows, MS-Office und andere Anwendungen unterstützen und die gefährlichen Code enthalten können aus [71]. 2016 wurde z. B. Malware als JavaScript-Dateien versendet, die auf PCs von Windows Script Host ohne Nachfrage ausgeführt wurde[72].

Durch geschickte Tarnung und Social Engineering können sehr viele Opfer dazu bewegt werden, einen bösartigen Anhang zu öffnen. PhishMe, ein Hersteller von Anti-Phishing-Training-Software, hat in einem Bericht die Ergebnisse aus über 40 Millionen simulierten Phishing-E-mails, die in über 1000 echten Unternehmen versendet wurden, im Zeitraum Januar 2015 bis Juli 2016, zusammengetragen. Unter anderem wurden die E-Mail Kampagnen echter Malware an Mitarbeiter versendet, dabei öffneten im Schnitt 17% der Mitarbeiter den Anhang, die effizienteste Kampagne erreichte eine Klickrate von 21,5%. [73]

Anstelle angehängter Dateien können in der E-Mail auch Http-Links enthalten sein, die auf Malware verweisen. [70]

3 Praxis

3.1 Anforderungen

Zu Beginn der Umsetzung wurden zusammen mit dem in Kapitel 1.2 erwähntem Team die Anforderungen an das Add-on, vor dem Hintergrund der in Kapitel 2 erläuterten Theorie, festgehalten:

Kompatibilität Der MIME-Standard (siehe Kapitel 2.3.3), die gängigen Formate Klartext und HTML, sowie Dateianhänge sollen unterstützt werden.

Sicherheit Das System soll gegen XSS-Angriffe (siehe Kapitel 2.4.4) gewappnet sein, die Verbreitung von Malware (siehe Kapitel 2.4.5.1) durch das System verhindern und Maßnahmen gegen SPAM ergreifen.

Konfigurierbarkeit Das System soll eine einfach zu bedienende Web-Schnittstelle (siehe Kapitel 2.2.4.2) bieten, mit der alle Einstellungen getroffen werden können.

Dokumentation Die Dokumentation des Add-ons soll in Englisch verfasst werden und zusammen mit dem Quellcode veröffentlicht werden. Sie soll aus einem Teil für Endanwender und einem für Entwickler bestehen. Zusätzlich soll diese Arbeit dem Quellcode beigelegt werden.

Lizenzen Es dürfen keine proprietären Bibliotheken und Schnittstellen, außer der Confluence-API, verwendet werden, um Lizenzprobleme zu vermeiden.

Tests Das Add-on soll sowohl mit Unit- als auch mit Integrations-Tests automatisiert getestet werden.

3.2 Marktanalyse

Zu Beginn der Arbeit wurden bestehende Lösungen für Confluence analysiert und mit den gesetzten Anforderungen verglichen (siehe Kapitel 3.1). Dabei sollte überprüft werden, ob es bereits ein Add-on gibt, das die Anforderungen erfüllt, und ob es eventuell eine Lösung gibt, die als Basis für eine Eigenentwicklung dienen kann. Im Atlassian-Marketplace sind die folgenden zwei Add-ons zu finden, die aus E-Mails Beiträge für Confluence erzeugen können.

3.2.1 Send EMail to Page Plugin

Das kostenpflichtige Confluence Add-on Send EMail To Page Plugin erzeugt aus E-Mails Wiki-Seiten oder Blog-Posts. Es kann mit HTML-, Textmails und Dateianhängen umgehen[74]. Bei der in Kapitel 3.5 beschriebenen Analyse des Autors stellte sich jedoch heraus, dass das Add-on eine Reihe von Sicherheitslücken besitzt. Der Preis für Atlassian-Add-ons hängt von der Anzahl der lizenzierten Nutzer ab, bei 2000 Lizenzen, kostet das erste Jahr 1600 € und jedes weitere Jahr 800 €[75].

3.2.2 Mail2News

Das kostenlose Add-on Mail2News erzeugt aus E-Mails Blögeinträge, allerdings ist es mit aktuellen Confluence-Versionen nicht mehr kompatibel[76]. Nach Analyse des Autors verarbeitet es Text-E-mails und Dateianhänge, kann aber nicht mit HTML-Mails umgehen. Für den Einsatz wird ein VERP-fähiger Mailserver benötigt (siehe Kapitel 2.3.1), da die Confluence-Spaces in der Empfängeradresse angegeben werden[76]. Bei der Analyse wurden die gleichen Sicherheitsprobleme wie bei Send EMail to Page gefunden (siehe Kapitel 3.5). Das Add-on ist open-source auf GitHub, unter der 3-Klausel-BSD-Lizenz veröffentlicht worden[77].

3.2.3 Fazit der Marktanalyse

Gegen den Einsatz beider Add-ons im Unternehmen sprachen die gefundenen Sicherheitsprobleme (siehe Kapitel 3.5). Bei Mail2News kommt noch hinzu, dass das Add-on nicht mit der im Unternehmen aktuellen Confluence-Version kompatibel ist und der im Unternehmen verwendete Mailserver (MS Exchange) kein VERP unterstützt. Auch die Anforderung HTML-E-Mails zu unterstützen, erfüllt Mail2News nicht. Damit wurde die Entscheidung ein eigenes Add-on zu entwickeln noch einmal erhärtet. Um den Entwicklungsvorgang zu beschleunigen, wurde der Quellcode von Mail2News als Basis für das eigenentwickelte Add-on Mail2Blog verwendet.

3.3 Struktur

Das, im Rahmen dieser Arbeit entwickelte, Add-on Mail2Blog ist ursprünglich als Fork des Projekts Mail2News (siehe Kapitel 3.2.2) entstanden. Der Code ist jedoch so stark refactored worden, dass es keine gemeinsame Codebasis mehr gibt. Um das nachzuprüfen, wurden beide Projekte mit dem vom Karlsruher Institut für Technologie (KIT) entwickeltem Tool JPlag¹ verglichen. JPlag wurde geschrieben um Softwareplagiate von Studenten zu finden, es wandelt die zu analysierenden Quellcodes, in Token-Streams um und vergleicht dann die beiden Streams, dadurch wird nur die Codestruktur, nicht aber Kommentare, Variablennamen oder Konstanten beim Vergleichen berücksichtigt[78]. JPlag findet keine Gemeinsamkeiten mehr zwischen den Projekten jenseits von ein paar trivialen Try-Catch-Blöcken und Import-Statements.

Ein extremes Refactoring ist nötig gewesen, da die Logik von Mail2News um eine große Gottesklasse herum organisiert ist (siehe Abbildung 3.1) und daher, nach Auffassung des Autors, schwer zu lesen und testen ist. Trotzdem wurde durch die Verwendung des Mail2News-Quellcode Zeit gespart, da von Anfang an der grundlegende Ablauf geklärt war und es, in einer älteren Confluence-Version funktionierenden, Beispielcode gab.

Während des Entwicklungsprozesses wurde die Gottesklasse in viele einzelne testbare und klar abgegrenzte Komponenten zerlegt (siehe Abbildung 3.2). Es wurden Unit- und Integrationstests (siehe Kapitel 3.6.1), Sicherheitsmaßnahmen (siehe Kapitel 3.5), neue Features und Dokumentation (siehe Kapitel 3.7) hinzugefügt. Der grundlegende Ablauf (siehe Kapitel 3.3.1), mit dem E-Mails abgearbeitet werden, ist ähnlich geblieben.

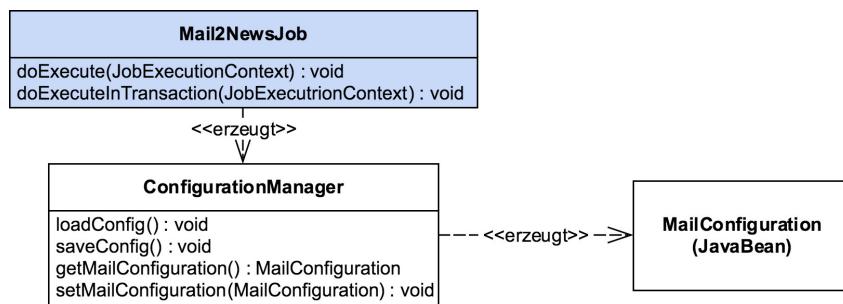


Abbildung 3.1: Mail2News-Klassendiagramm mit Gottesklasse Mail2NewsJob

¹<https://github.com/jplag/jplag>

Praxis – Struktur

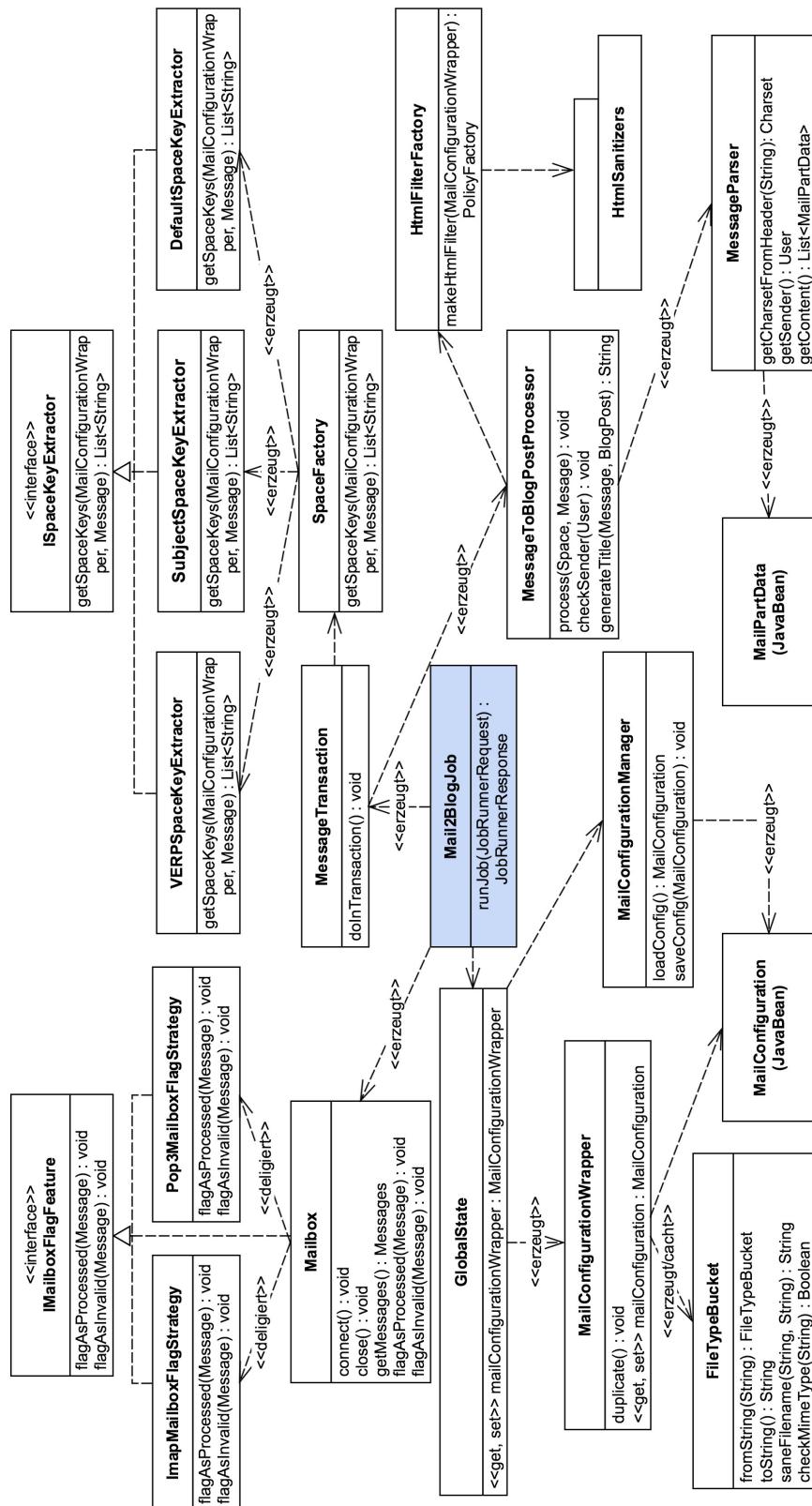
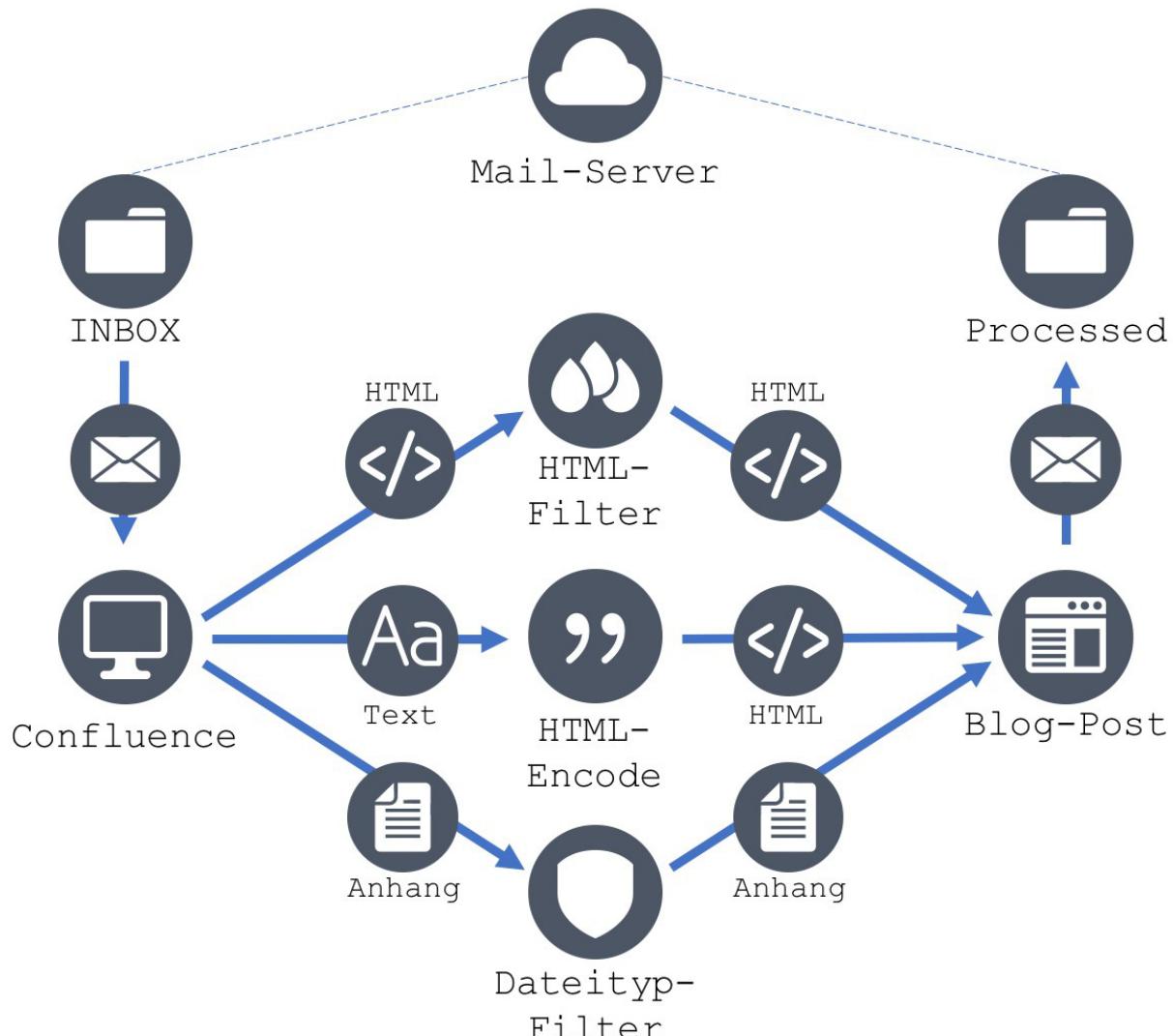


Abbildung 3.2: Mail2Blog-Klassendiagramm

3.3.1 Ablauf



© Icons by Elegant Themes / <https://www.elegantthemes.com/blog/freebie-of-the-week/beautiful-flat-icons-for-free/> / GPLv2

Abbildung 3.3: Mail2Blog-Datenfluss während des Abarbeitungsprozesses

Neue E-Mails im Postfach werden von einem zeitgesteuerten Job (siehe Kapitel 2.2.4.1) regelmäßig abgearbeitet. Der Job verbindet sich mit dem, in den Einstellungen hinterlegtem, Mailserver und iteriert dann über alle E-Mails, die sich im Posteingang befinden. Dabei wird die älteste E-Mail zuerst abgearbeitet, damit die neuste Nachricht als Erstes im Blog auftaucht. Jede Nachricht wird in einer neuen Transaktion behandelt, damit im

Fehlerfall keine unvollständigen Blog-Posts zurückbleiben. Zunächst wird der Confluence-Space (siehe Kapitel 2.2.2) bestimmt, indem die Nachricht veröffentlicht werden soll (siehe Kapitel 3.3.4). Danach wird der Inhalt der E-Mail verarbeitet (siehe Kapitel 3.3.5): In der E-Mail erhaltenes HTML wird gefiltert (siehe Kapitel 3.5.1), Plain-Text als HTML encodiert und Anhänge überprüft (siehe Kapitel 3.5.3). Bei Verwendung des IMAP-Protokolls (siehe Kapitel 2.3.4) wird, nachdem der Vorgang erfolgreich beendet wurde, die Nachricht in den Ordner Processed verschoben; im Fehlerfall in den Ordner Invalid, wo sie von einem Administrator manuell analysiert und bearbeitet werden kann. Bei Verwendung des POP-Protokolls (siehe Kapitel 2.3.4), wird die E-Mail immer gelöscht, da POP3 nicht mit Ordnern umgehen kann.

3.3.2 Konfigurationssystem

Alle Einstellungen des Add-ons werden von einem zentralen Konfigurationssystem verwaltet. Administratoren können die Einstellungen im Administratoren-Interface mit einer grafischen Web-Schnittstelle bearbeiten (siehe Kapitel 3.4).

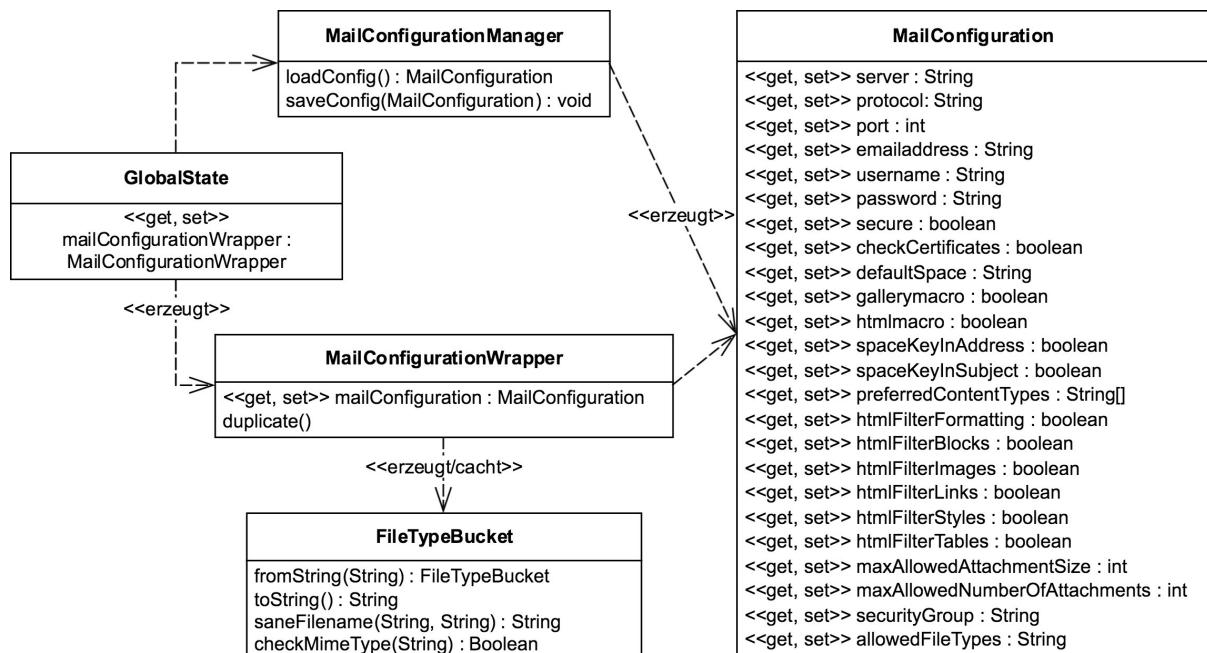


Abbildung 3.4: Klassendiagramm des Konfigurationssystems

Praxis – Struktur

Die Klasse MailConfiguration ist ein JavaBean, das sämtliche Einstellungen des Add-ons speichert. Alle Einstellungen sind als primitive Datentypen, Strings oder Arrays abgebildet worden, damit die Konfiguration einfach serialisiert/deserialisiert und geklont werden kann. Die Liste der erlaubten Dateitypen (siehe 3.5.3) wird als CSV-String abgespeichert, die Klasse FileTypeBucket deserialisiert das CSV in eine Hashtabelle, um Dateiendungen und Content-Types einfach und schnell zu prüfen. Damit der FileTypeBucket nicht bei jeder Verwendung neu generiert werden muss, wird er zusammen mit der passenden MailConfiguration in einem MailConfigurationWrapper gespeichert. Der MailConfigurationWrapper lazy-loaded den FileTypeBucket bei der ersten Verwendung und überwacht den Wert der CSV-Liste in der MailConfiguration, bei einer Änderung wird der FileTypeBucket neu erstellt. Dauerhaft werden die Einstellungen im PluginSettings-Speicher von Confluence gespeichert, von wo sie mit dem MailConfigurationManager geladen/gespeichert werden können. Das Singleton GlobalState speichert die globale aktuell gültige Konfiguration und benutzt den MailConfigurationManager um die Konfiguration lazy-zu-laden.

Durch diese Architektur, wird gewährleistet, dass mehrere Konfigurationen im Hintergrund verwendet werden können, die globale Konfiguration von überall abgerufen werden kann und Objekte nicht aufwendig neu deserialisiert werden. Eine extra Konfiguration wird z. B. benötigt, wenn neue Einstellungen zunächst getestet werden und die globale Konfiguration nicht überschrieben werden soll (siehe Kapitel 3.4). In Zukunft könnte dieses System so auch um die parallele Kommunikation mit mehreren Postfächern erweitert werden.

3.3.3 Mailbox

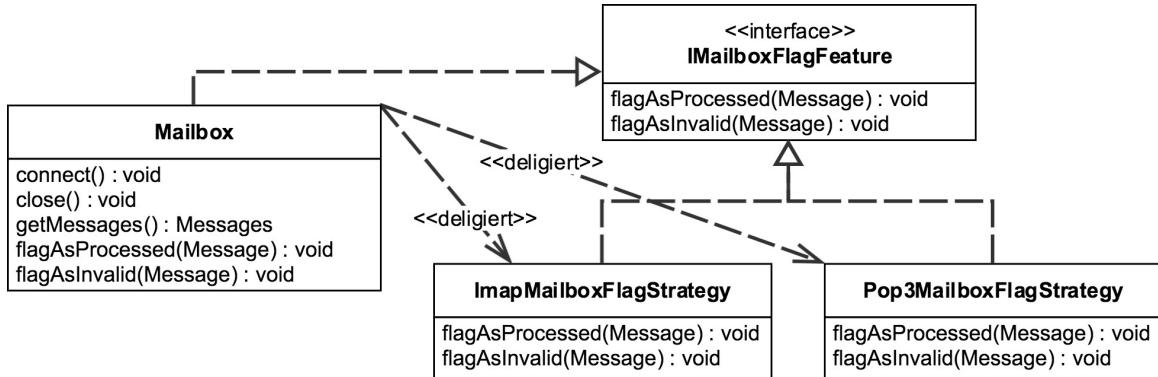


Abbildung 3.5: Klassendiagramm des Mailboxsystems

Die Kommunikation mit dem Mailserver wird durch die Klasse Mailbox gekapselt. Zur Kommunikation per IMAP (siehe Kapitel 2.3.4) und POP (siehe Kapitel 2.3.4) wird die open-source lizenzierte JavaMail² Bibliothek verwendet. Die Aufgaben der Mailbox sind: Verbindung zum Mailserver aufzubauen, neue Nachrichten abholen und eine Nachricht als erfolgreich oder nicht erfolgreich bearbeitet zu markieren. Wie bereits im Abschnitt 3.3.1 erwähnt, werden IMAP und POP Nachrichten unterschiedlich markiert. Diese konkreten Vorgehensweisen wurden dem Strategiepattern nach, in die Klassen IMAPMailboxFlagStrategy und POP3MailboxFlagStrategy ausgelagert.

²<https://javaee.github.io/javamail/>

3.3.4 Space-Bestimmung

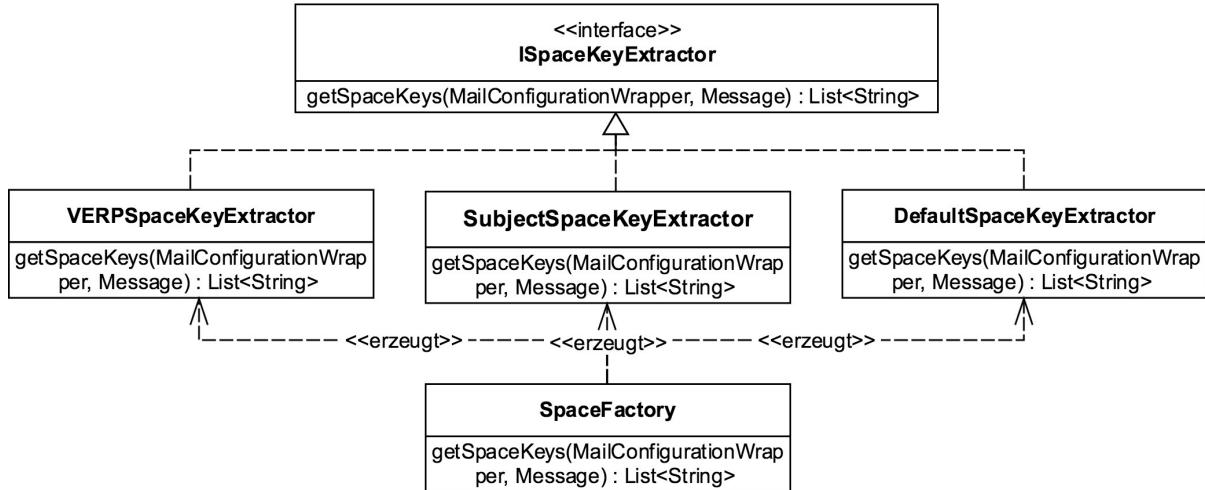


Abbildung 3.6: Klassendiagramm der Space-Bestimmung

Mail2Blog unterstützt verschiedene Strategien um den Confluence-Space (siehe Kapitel 2.2.2) zu bestimmen, in dem eine E-Mail gepostet wird. Die Factory SpaceFactory fragt alle aktivierten Strategien ab und bündelt das Ergebnis: Die Klasse VERPSpaceKeyExtractor versucht den Space-Key aus der Senderadresse zu extrahieren, SubjectSpaceKey sucht den Space-Key im Betreff der E-Mail, DefaultSpaceKeyExtractor gibt den, in der Konfiguration (siehe Kapitel 3.3.2) hinterlegten, Standard-Space-Key zurück.

3.3.5 Inhaltsverarbeitung

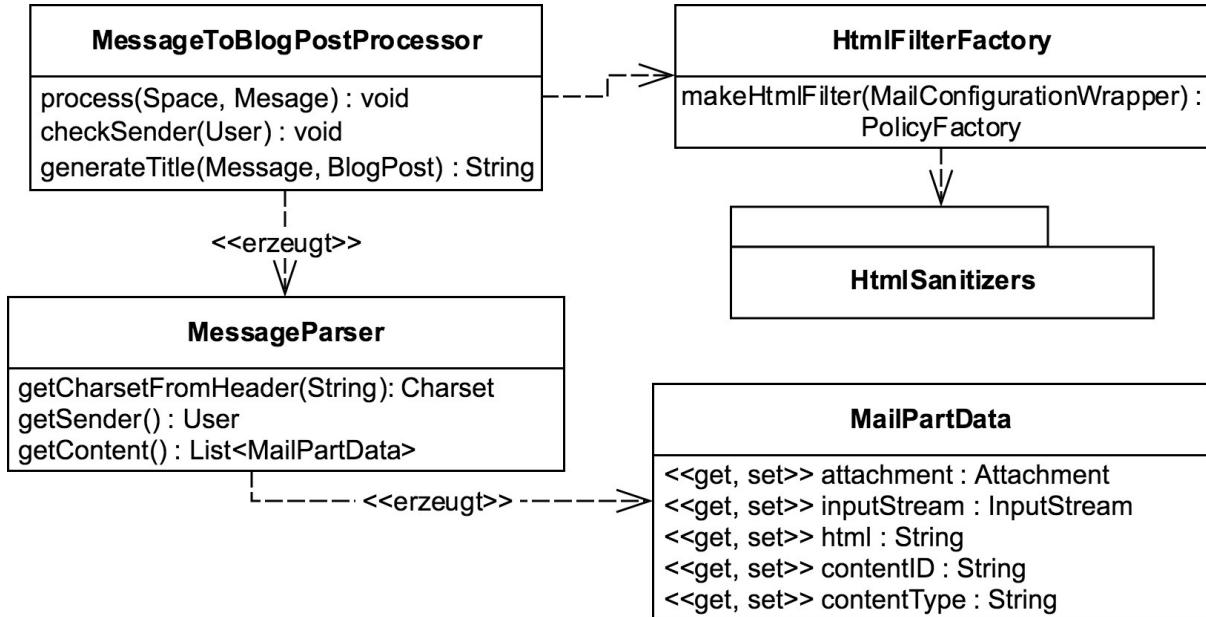


Abbildung 3.7: Klassendiagramm der Inhaltsverarbeitung

Die Klasse `MessageToBlogPostProcessor` wandelt E-Mails in Blogeinträge und postet diese dann in Confluence. Die Klasse `MessageParser` parst eine E-Mail und gibt den Inhalt als Liste von JavaBeans (`MailPartData`) zurück. Die Factory `HtmlFilterFactory` erzeugt einen HTML-Filter (siehe Kapitel 3.5.1) aus dem Paket `HtmlSanitziers`.

3.3.5.1 Parsing

Ein Blog-Eintrag (siehe Kapitel 2.2.1) in Confluence besteht aus HTML-Inhalten und Anhängen. JavaMail liefert den Inhalt einer Mime-Mail (siehe Kapitel 2.3.3) jedoch als Baum. Daher muss diese Struktur zunächst umgewandelt werden. Dazu wird der Baum von der Wurzel zu den Blättern geparsed. Trifft der Parser auf einen Knoten mit dem Content-Type `multipart/alternative`, versucht er in einem der Kindsknoten, den in den Einstellungen gespeicherten bevorzugten Content-Typ, zu finden und die Geschwisterknoten zu verwerfen. Inhalte mit dem Content-Typ `text/plain` werden in HTML gewandelt und dann wie `text/html` behandelt. Alle anderen Inhaltstypen werden als Anhänge behandelt. Der Parsing-Algorithmus ist als Pseudo-Code in Listing 3.1 beschrieben.

Praxis – Struktur

Listing 3.1: Algorithmus zum E-Mail-Parsing in Pseudocode

```
1 ; Hole den Inhalt einer E-Mail als Liste von Datensätzen
2 FUNCTION getContent(Nachricht)
3     RETURN extrahiere(Nachricht)
4
5 ; Erstelle eine Liste mit allen Datensätzen des Knoten.
6 FUNCTION extrahiere(Knoten)
7     IF Knoten hat Content-Type "multipart/*"
8         RETURN extrahiereMultipart(Knoten)
9     ELSE
10        RETURN LIST(extrahiereDaten(Knoten))
11
12 ; Erstelle eine Liste mit allen Datensätzen des Mutlipart.
13 ; Wenn es sich um multipart/alternative handelt:
14 ;   - nutze den ersten Teil mit dem bevorzugtem Inhaltstyp
15 ;   - nutze alle Teile, falls es keinen passenden Teil gibt
16 FUNCTION extrahiereMultipart(Knoten)
17     Alle_Inhalte := LIST()
18     Bevorzugte_Inhalte := LIST()
19     Bevorzugte_Gefunden := FALSE
20
21 FOR Kind IN Knoten
22     Inhalte := extrahiere(Kind)
23     Alle_Inhalte += Inhalte
24
25     IF Knoten hat Content-Type "multipart/alternative"
26     AND Bevorzugte_Gefunden = FALSE
27     AND Inhalte hat Element mit Content-Type = bevorzugter Typ
28         Bevorzugte_Gefunden := TRUE
29         Bevorzugte_Inhalte := Inhalte
30
31     IF Bevorzugte_Gefunden
32         RETURN Bevorzugte_Inhalte
33     ELSE
34         RETURN Alle_Inhalte
35
36 ; Erstelle einen Datensatz mit relevanten Daten. Stark vereinfacht.
37 FUNCTION extrahiereDaten(Knoten)
38     Datensatz := OBJECT()
39     Datensatz.Content-Type := Knoten.Content-Type
40     IF Knoten hat Content-Type "text/html"
41         Datensatz.html := Knoten.html
42     ELSE IF Knoten den Content-Type "text/plain" hat
43         Datensatz.html := escapeHtml(Knoten.text)
44     ELSE
45         Datensatz.attachment := Knoten.stream
46     RETURN Datensatz
```

3.3.5.2 Anhänge

Wie im Pseudocode für den Parsing-Algorithmus bereits angedeutet, verarbeitet der Mail-Parser auch E-Mail Anhänge und erzeugt daraus Confluence-Anhänge. Bevor der Anhang verarbeitet wird, wird überprüft, ob der angegebene Content-Type in den Einstellungen erlaubt ist und ein sicherer Dateiname erzeugt (siehe Kapitel 3.5.3 und Kapitel 3.5.2). Da Confluence beim Speichern der Anhänge die genaue Dateigröße in Bytes benötigt und die genaue Länge des Eingabestreams unbekannt ist, muss der Stream für jeden Anhang einmal komplett eingelesen werden. Dazu wird der Eingabestream in einer Schleife kilobyteweise in einen ByteArrayOutputStream eingelesen. Sollte dabei die in den Einstellungen konfigurierte maximale Anhanggröße überschritten, wird abgebrochen und der Anhang ignoriert. Für jeden Parsing-Vorgang gibt es einen Zähler, der die Anzahl der erfolgreich verarbeiteten Anhänge pro Blog-Eintrag zählt. Ist die in den Einstellungen gespeicherte maximale Anzahl an Anhängen pro Blog-Eintrag erreicht, werden alle weiteren Anhänge ignoriert. Der Prozess ist in der Abbildung 3.9 auf Seite 47 abgebildet.

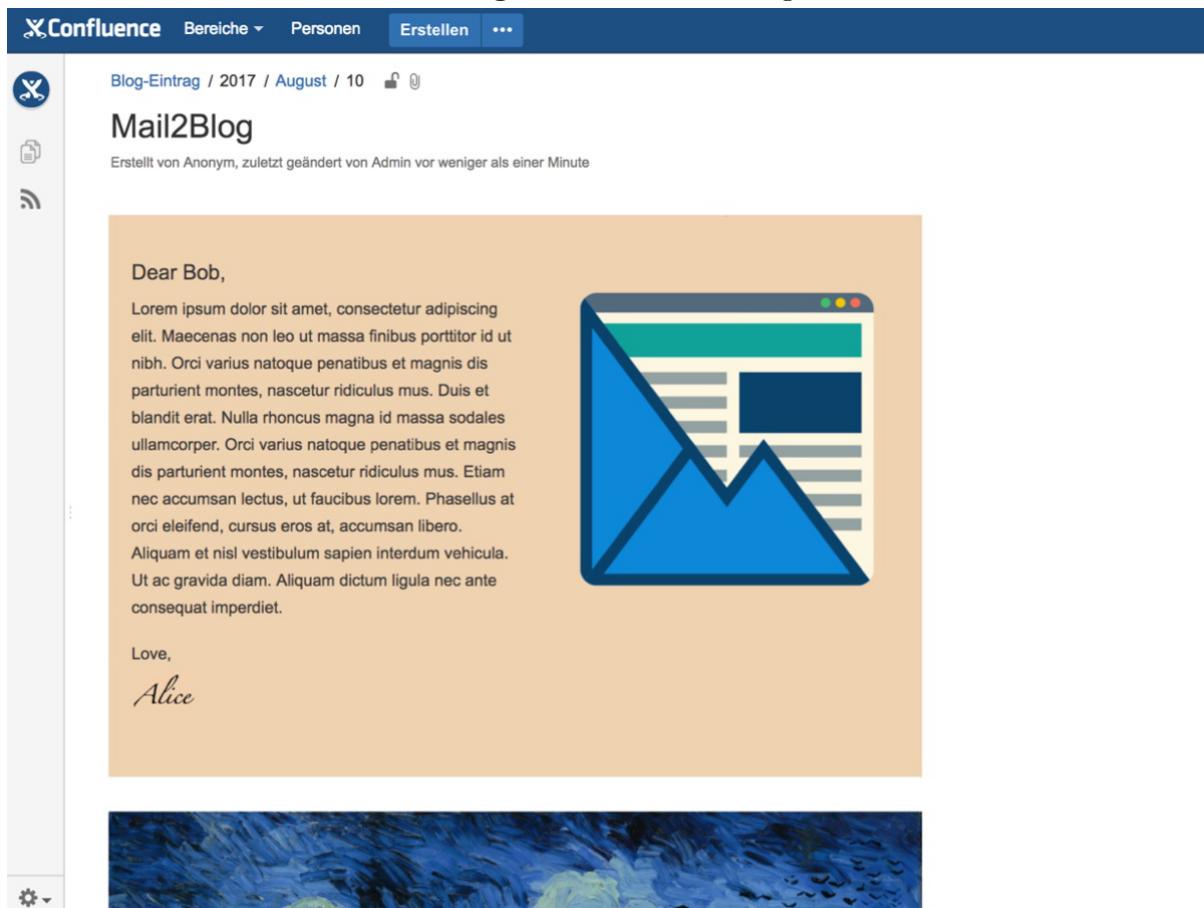
3.3.5.3 Blog-Post-Erstellung

Der Blog-Post wird von der Klasse MessageToBlogPostProcessor erzeugt. Als Erstes wird ein MessageParser erstellt, um die E-Mail zu analysieren. Anhand der Absenderadresse wird versucht ein Confluence-User zu finden, welcher der E-Mail-Adresse zugeordnet werden kann. Falls der Absender-Filter aktiviert ist (siehe Kapitel 3.5.4), wird überprüft, ob der Absender berechtigt ist, Blog-Einträge per E-Mail zu erstellen. Danach wird der Confluence-Blog-Beitrag erstellt und dem Beitrag Ersteller, Erstellungsdatum und Confluence-Space (siehe Kapitel 3.3.4) zugewiesen. Als Titel wird der Betreff der E-Mail verwendet. In jedem Confluence-Space muss der Titel eindeutig sein, daher wird überprüft, ob der Titel bereits vergeben ist. Bei Bedarf wird dem Titel bis zu dreimal ein „+“ angehängt, sollte der Titel dann immer noch vergeben sein, wird ein Globally Unique Identifier (GUID) an den Titel angehängt; um eine Endlosschleife bei einer fehlerhaften Programmierung zu verhindern, wird der Vorgang beim 5ten Durchlauf abgebrochen, obwohl dieser Fall aufgrund der Einzigartigkeit einer GUID nie vorkommen darf. Bevor Confluence-Anhänge gespeichert werden können, muss der zugehörige Beitrag schon existieren, daher wird der Beitrag zunächst ohne Inhalt gespeichert. Dann werden alle Anhänge, die der MessageParser liefert,

Praxis – Struktur

dem Blog-Post zu gewiesen und der Ersteller gesetzt. Alles HTML, das der MessageParser liefert wird konateniert. HTML-Links auf Anhänge werden repariert, indem für jeden Anhang CID://CONTENT-ID-DES-ANHANGS durch den passenden HTTP(s)-Link ersetzt wird. Danach wird das HTML durch den HTML-Filter gesendet (siehe 3.5.1). Ist das HTML-Makro in den Einstellungen aktiviert, wird der gesamte Inhalt in ein Confluence-HTML-Makro (siehe Kapitel 3.5.1.1) eingebettet, auf diese Weise werden auch komplexe HTML-Inhalte richtig dargestellt. Im Anschluss wird eine HTML-Liste mit allen Anhängen generiert und dem Beitrag angefügt. Ist das Gallery-Makro, das angehängte Bilder in einer Galerie anzeigt, in den Einstellungen aktiviert wird es angehängt. Am Ende wird dem Blog-Beitrag das generierte HTML als Inhalt zugewiesen. Der Prozess ist in der Abbildung 3.10 auf Seite 48 abgebildet.

Abbildung 3.8: Generierter Blog-Post



The screenshot shows a Confluence blog entry titled "Mail2Blog". The header includes navigation links for "Bereiche", "Personen", "Erstellen", and "...". Below the title, it says "Blog-Eintrag / 2017 / August / 10" and "Erstellt von Anonym, zuletzt geändert von Admin vor weniger als einer Minute". The main content area contains a message from Alice to Bob. The message starts with "Dear Bob," followed by a long block of placeholder text (Lorem ipsum). It ends with "Love," and the signature "Alice". To the right of the text, there is a graphic of a computer monitor displaying a blue triangle pointing downwards over a grid of horizontal bars. At the bottom of the page, there is a decorative footer image featuring a blue and green abstract pattern.

Praxis – Struktur

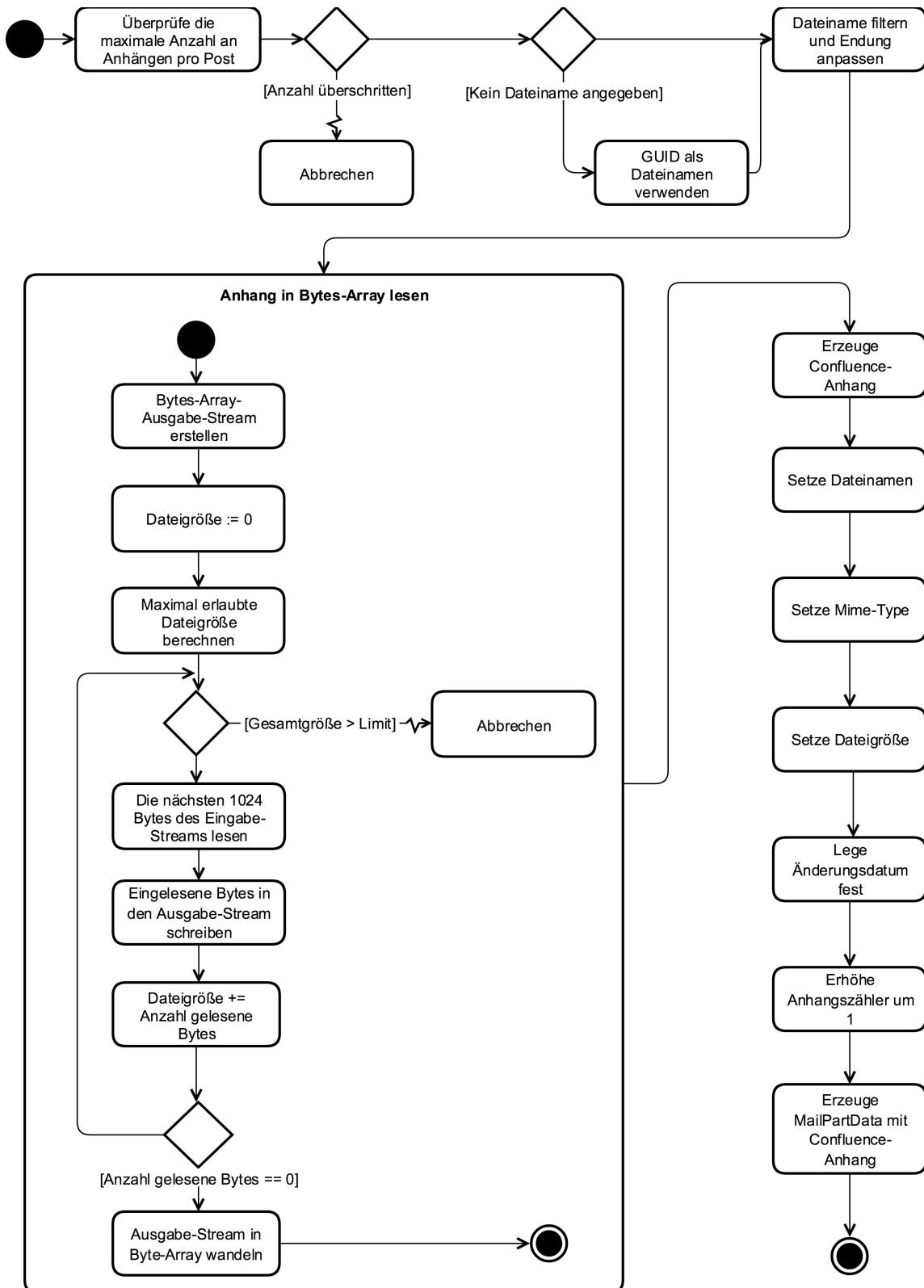


Abbildung 3.9: Aktivitätsdiagramm der Anhang-Verarbeitung

Praxis – Struktur

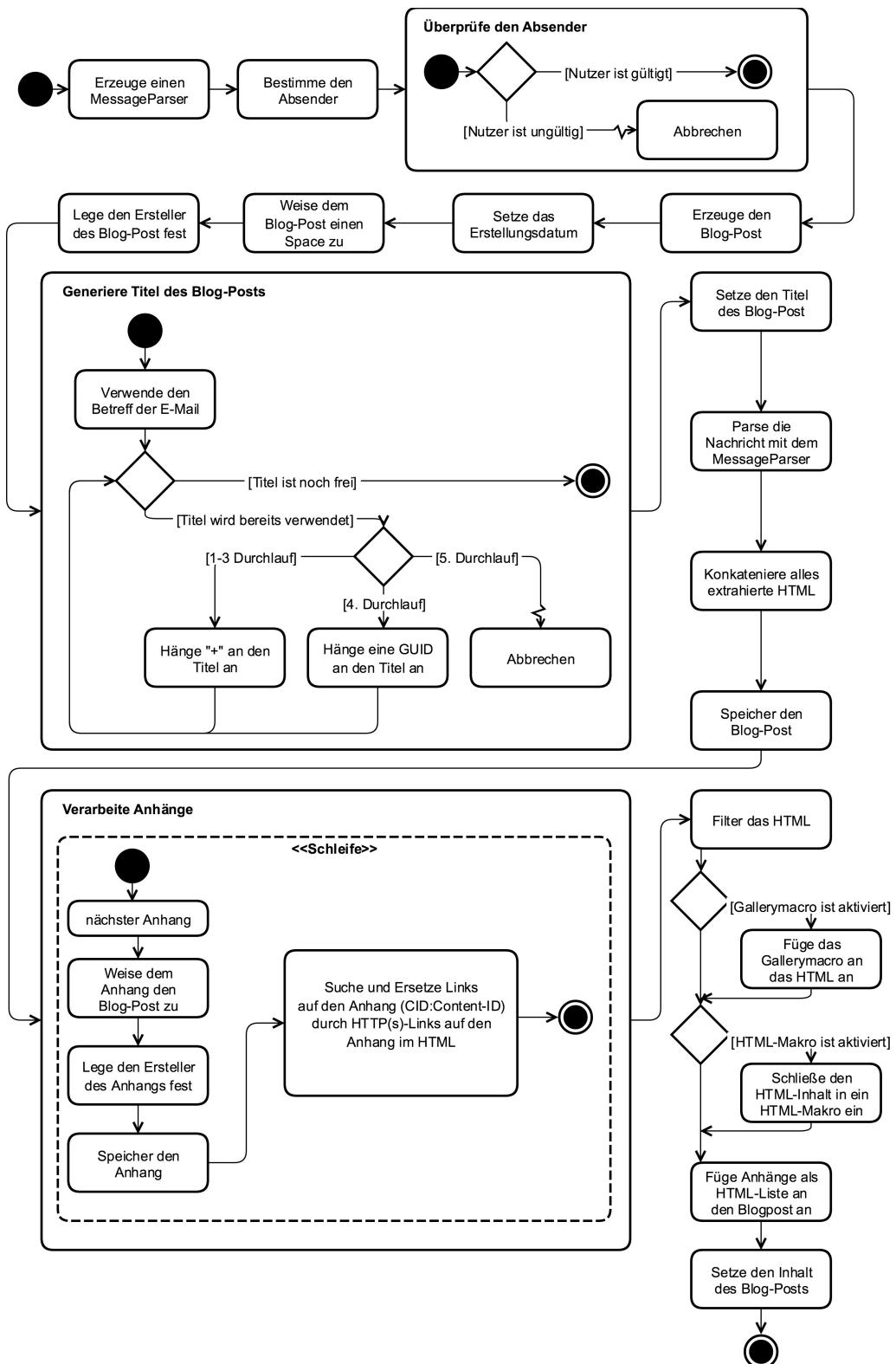


Abbildung 3.10: Aktivitätsdiagramm der E-Mail zu Blog-Post-Verarbeitung

3.4 Konfigurations-UI

Das Add-on Mail2Blog verfügt über eine Konfigurationsseite (siehe Kapitel 2.2.4.2), auf der alle Einstellungen die in der Klasse MailConfiguration gespeichert werden, von Administratoren vorgenommen werden können.

Die Klassen des UI-Systems sind in der Abbildung 3.12 abgebildet. Ähnlich wie das Singleton GlobalState die aktuell verwendete gloable Konfiguration speichert (siehe Kapitel 3.3.2), speichert ConfigurationActionState den aktuellen Zustand der Konfiguration auf der Konfigurationsseite, die aus dem GlobalState initialisiert wird. Wie im Kapitel 2.2.4.2 beschrieben, wird die Seite von einem Template aufgebaut. Beim Speichern wird zunächst die Methode validate() am Kontroller (ConfigurationActionState) aufgerufen, welche die Eingaben überprüft. Dabei wird versucht eine Verbindung zum Server aufzubauen und überprüft, ob die Werte in akzeptierten Bereichen liegen. Für den Fall, dass die Validierung erfolgreich ist, wird danach die Methode execute() am Kontroller aufgerufen, die mithilfe des MailConfigurationManager die Konfiguration abspeichert und dann sowohl den GlobalState als auch den ConfigurationActionState aktualisiert.

In HTML-Formularen wird nur der Wert von ausgewählten Checkboxen übertragen[79], Confluence ruft daher nur den Setter auf, wenn der Wert einer Checkbox gesetzt wird, nicht jedoch wenn die Checkbox abgewählt wird. Um trotzdem Checkboxen verwenden zu können, ist ein Workaround nötig. Die Klasse CheckboxTracker repliziert alle booleschen Felder der MailConfiguration. Vor jedem Aufruf der Konfigurationsseite werden alle Felder im Tracker auf FALSE gesetzt. Im HTML-Formular werden die Werte zwar aus der MailConfiguration geladen, aber in den Tracker geschrieben, sodass nicht übermittelte Checkboxen automatisch den Wert FALSE haben. Am Anfang der Validierungsphase werden die Werte aus dem Tracker dann in die MailConfiguration übertragen.

Die, für die Ausführung des Add-ons notwendigen, Einstellungen sind auf der Konfigurationsseite hervorgehoben, alle erweiterten Einstellungen sind unter Reitern aus dem Fokus genommen worden. Administratoren werden nicht daran gehindert potenziell problematische Sicherheitseinstellungen (siehe Kapitel 3.5) vorzunehmen, allerdings wurden hervorgehobene Warnhinweise unmittelbar neben diesen Einstellungen platziert, die Administratoren über Risiken aufklären. Fehler bei der Validierung werden sowohl neben

Praxis – Konfigurations-UI

den betroffenen Feldern als auch über dem Formular dargestellt. Insgesamt wurde versucht durch Hinweise und Gruppierungselemente das Formular so einfach wie möglich, zu gestalten.

Mail to Blog Configuration

The following error(s) occurred:

- Please enter a user name
- Failed to connect to mailbox: Incomplete mail configuration settings (at least one setting is null/empty).

① Help

- Documentation
- Source Code

① Warning: When using POP3 processed messages get deleted, because POP3 doesn't support folders.

① Warning: If you disable SSL, you're username, password and messages will be readable to an attacker.

Server*

Protocol

Use SSL Enabled

Port*

Email Address*

User name*
Please enter a value

Password

Default Space*

Default Space into which to post

Advanced Configuration

Spaces Formatting Attachments Senders Miscellaneous

Spaces

Confluence stores pages and blog posts in [spaces](#). All spaces are identified by their [space key](#). By default all posts land in the default space, but you can enable other strategies for determining the space into which to post.

Address Look for the space key in the mail address
Use spacekey+recipient@example.org to set the space key

Subject Look for space key in the subject line
Use "spacekey: Title" as subject to set the space key

Abbildung 3.11: Konfigurations-UI

Praxis – Konfigurations-UI

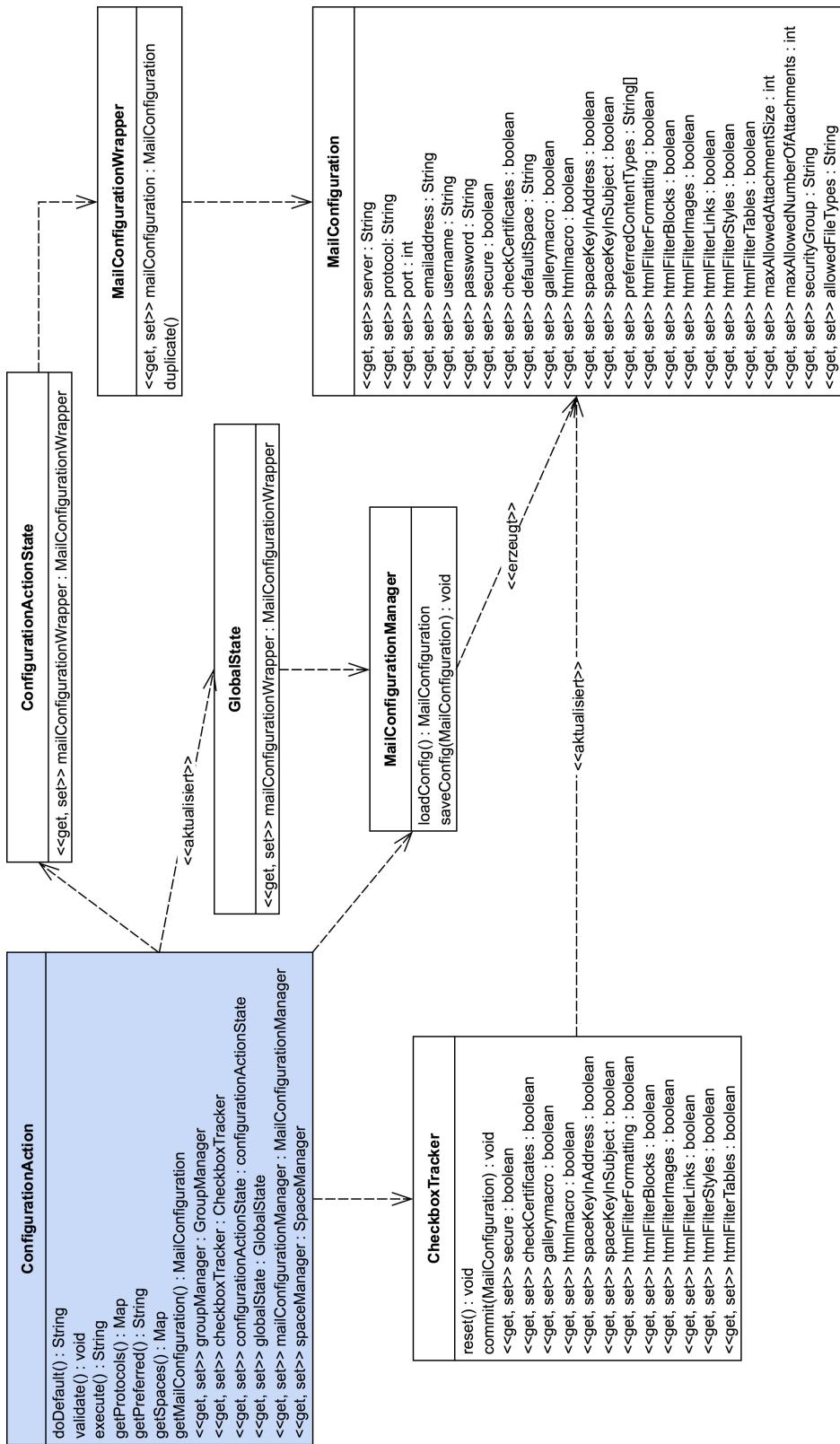


Abbildung 3.12: Klassendiagramm der Konfigurationsseite

3.5 Sicherheitsmaßnahmen

Das folgende Kapitel erläutert die Sicherheitsmaßnahmen, die getroffen wurden um sich gegen die im Kapitel 2.4 beschriebenen Gefahren zu wehren und die in Kapitel 3.1 beschriebenen Sicherheitsanforderungen einzuhalten. Dabei werden beispielhafte Angriffsszenarien gegen das neu entwickelte Add-on und die bestehenden Lösungen (siehe Kapitel 3.2) vorgestellt.

3.5.1 HTML-Filter

Beim HTML-Filter handelt es sich um einen Eingabefilter der XSS-Angriffe (siehe Kapitel 2.4.4) verhindert.

Das E-Mail to Page Plugin (siehe Kapitel 3.2.1) und Mail2News (siehe Kapitel 3.2.2) geben in E-Mails enthaltenes HTML ungefiltert an Confluence weiter. Confluence bereinigt die Eingaben vor dem Abspeichern und entfernt alle unbekannten Tags und Attribute und überführt HTML in XHTML, um mit dem in Kapitel 2.2.3 beschriebenen Speicherformat kompatibel zu sein[80]. Daher wird Javascript in Script-Tags und in Attributen (wie onerror, onmouseover, href) entfernt und einfache XSS-Angriffe schlagen fehl. Allerdings können in den E-Mails Confluence-Makros enthalten sein, die eine Reihe von Angriffsmöglichkeiten bieten, ein Beispiel dafür befindet sich im anschließenden Abschnitt 3.5.1.1.

Mail2Blog verhindert diese Angriffe, indem es das in E-Mails enthaltene HTML zunächst durch einen Whitelistfilter sendet und dadurch alle Confluence-Makros im Vorfeld entfernt. Mail2Blog greift dabei auf die, open-source lizenzierte, OWASP Java HTML Sanitizer Bibliothek³ zurück, um das HTML zu filtern. Die Bibliothek filtert HTML-Elemente, Attribute und die Inhalte von Attributen und unterbindet somit, nach Erfahrung des Autors, effektiv XSS-Angriffe. Die Bibliothek baut auf Code von Google aus dem Projekt Caja auf, der unter der Apache-Lizenz open-source veröffentlicht wurde[81]. Die Bibliothek liefert eine Reihe vordefinierter Regelsätzen zum Filtern von HTML[82]. Mail2Blog erlaubt

³https://www.owasp.org/index.php/OWASP_Java_HTML_Sanitizer_Project

standardmäßig alle Elemente aus diesen Regeln, in den Einstellungen können jedoch, einzelne Regelsätze aktiviert und deaktiviert werden.

HTML Filters

All HTML contained in mails is filtered, to remove dangerous or broken code, before getting posted in confluence. The filter uses a whitelist approach, you can fine control what is allowed in mails using the checkboxes below.

- | | |
|------------|---|
| Blocks | <input checked="" type="checkbox"/> Allow common block elements including <p>, <h1>, etc. |
| Formatting | <input checked="" type="checkbox"/> Allow common formatting elements including , <i>, etc. |
| Images | <input checked="" type="checkbox"/> Allow elements from HTTP, HTTPS, and relative sources |
| Links | <input checked="" type="checkbox"/> Allow HTTP, HTTPS, MAILTO, and relative links |
| Styles | <input checked="" type="checkbox"/> Allow certain safe CSS properties in style="..." attributes |
| Tables | <input checked="" type="checkbox"/> Allow html tables |

Abbildung 3.13: HTML-Filter Einstellungen

3.5.1.1 Angriffsszenario

Confluence bietet ein HTML-Makro und ein HTML-Include-Makro, welche die Möglichkeit bieten ungefiltertes HTML in einem Beitrag einzubinden. Das Makro ist standardmäßig deaktiviert. Wird es aktiviert, kann jeder Nutzer der Beiträge verfassen darf, auch JavaScript durch diese Makros ausführen. Daher warnt Atlassian auch vor der Aktivierung der Makros. [83]

Wird Confluence in einem geschlossenen Netzwerk, mit einer vertrauenswürdigen und überschaubaren Nutzerzahl verwendet, kann, nach Meinung des Autors, der Nutzen das Risiko überwiegen und das Makro wird eventuell aktiviert. Können dann aber Confluence-Makros in E-Mails enthalten sein, kann auch jeder, der eine E-Mail an das System senden kann, einen Angriff mit dem HTML-Makro durchführen.

Praxis – Sicherheitsmaßnahmen

Beim E-Mail to Page Plugin (siehe Kapitel 3.2.1) lässt sich der Payload in eine Html-Mail integrieren. Mail2News (siehe Kapitel 3.2.2) arbeitet nur Plain-Text-E-Mails (Content-Type text/plain) ab, encodiert den Inhalt aber nicht. Daher lässt sich der Payload in eine Plain-Text-E-Mail integrieren. Bei Mail2Blog lässt sich kein HTML-Macro einschleusen, da es vom HTML-Filter (siehe Kapitel 3.5.1) herausgefiltert wird.

Listing 3.2: HTML-Makro – E-Mail-Exploit für Send Email to Pages

```
1 From: attacker@example.org
2 To: victim@example.org
3 Subject: XSS
4 Content-Type: text/html; charset="utf-8"
5 Content-Transfer-Encoding: 8bit
6
7 <html>
8     <body>
9         <ac:structured-macro ac:name="html">
10            <ac:plain-text-body>
11                <![CDATA[
12                    <img src=bogus onerror=alert('XSS')>
13                ]]>
14            </ac:plain-text-body>
15        </ac:structured-macro>
16    </body>
17 </html>
```

Listing 3.3: HTML-Makro – E-Mail-Exploit für Mail to News

```
1 From: attacker@example.org
2 To: victim+SPACEKEY@example.org
3 Subject: XSS-Text
4 Content-Type: text/plain; charset="utf-8"
5 Content-Transfer-Encoding: 8bit
6
7 <ac:structured-macro ac:name="html"><ac:plain-text-body><![CDATA[ <img
→   src=bogus onerror=alert('XSS')>
→   ]]></ac:plain-text-body></ac:structured-macro>
```

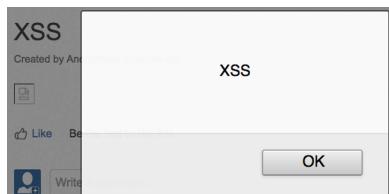


Abbildung 3.14: HTML-Makro
– Send EMail to Pages

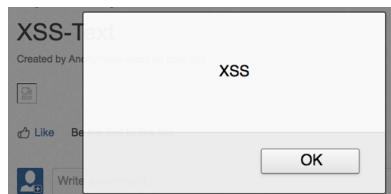


Abbildung 3.15: HTML-Makro
– Mail to News



Abbildung 3.16: HTML-Makro
– Mail2Blog

3.5.2 Dateinamen-Filter

Um sicherzugehen, dass Dateinamen keine Angriffsfläche bieten, ersetzt Mail2Blog alle Zeichen die nicht alphanumerisch oder Punkt(.) und Bindestrich(-) sind, vor dem Speichern durch einen Unterstrich(_).

Listing 3.4: Java-Code, der Dateinamen bereinigt

```
filename = filename.replaceAll("[^a-zA-Z0-9.\-\_]", "_");
```

3.5.2.1 Angriffsszenario

Eine Sicherheitslücke in Confluence < 5.10.6 (veröffentlicht am 23.09.2016[84]) sorgt dafür, dass JavaScript in Dateinamen beim Suchen nach Dateien ausgeführt wird[84]. Diese Sicherheitslücke lässt sich auch im Dateinamen von E-Mail Anhängen verwenden.

Praxis – Sicherheitsmaßnahmen

Listing 3.5: Dateinamen – E-Mail-Exploit

```
1 From: attacker@example.org
2 To: victim@example.org
3 Subject: XSS
4 Content-Type: multipart/mixed; boundary=BOUNDARY
5 Date: Wed, 2 Aug 2017 10:12:18 +0200
6 MIME-Version: 1.0
7
8 --BOUNDARY
9 Content-Transfer-Encoding: 7bit
10 Content-Type: text/plain; charset="us-ascii"
11
12 Hello World
13 --BOUNDARY
14 Content-Transfer-Encoding: base64
15 Content-Disposition: inline; filename=<script>alert('XSS');</script>.png"
16 Content-Type: image/png; x-unix-mode=0644; name=<script>alert('XSS');</script>.png"
17 Content-ID: <808E15E6-6623-4127-80DC-DD58C90A339F>
18
19 iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAAAAAA6fptVAAAACk1EQVR4nGP6DwABBQECz6AuzQAA
20 AABJRU5ErkJgg==
21 --BOUNDARY--
```

Name	Size	Creator	Creation Date	Comment
>  <script>alert('XSS');</script>.png	0.1 kB	Anonymous	Aug 02, 2017 10:33	added by the Send-EMail-To-Page plugin

Name	Size	Creator	Creation Date	Comment
>  <script>alert('XSS');</script>.png	0.1 kB	Anonymous	Aug 02, 2017 10:46	Attachment added by mail2news

Name	Size	Creator	Creation Date	Comment
>  _script_alert__xss____script__.png	0.1 kB	admin	Aug 02, 2017 09:27	

Abbildung 3.17: Dateinamen – Verarbeitete Anhänge
Send EMail to Pages (oben), Mail to News (mitte), Mail2Blog (unten)

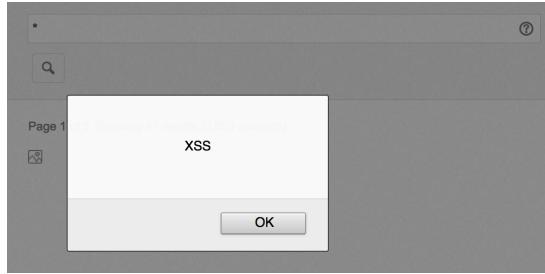


Abbildung 3.18: Dateinamen – Suche Send EMail to Pages & Mail2News

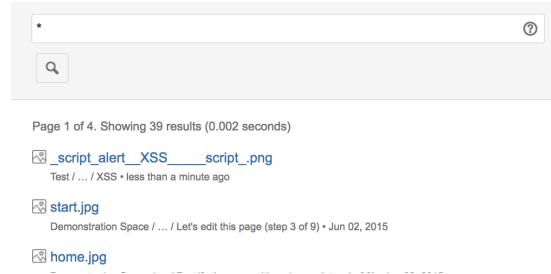


Abbildung 3.19: Dateinamen – Suche Mail2Blog

3.5.3 Dateityp-Filter

Beim Dateityp-Filter von Mail2Blog handelt es sich um einen Whitelistfilter, der in den Standardeinstellungen alle Anhänge, die ausführbaren Code enthalten können, blockiert. Mail2Blog hängt somit keine Malware (siehe Kapitel 2.4.5.1) an Confluence-Beiträge an. Möglich sind jedoch Angriffe auf Sicherheitslücken in Anwendungsprogrammen wie PDF-Reader, Image-Viewer, etc., welche an sich sichere Dateitypen öffnen. Confluence wird in der Regel als internes Wiki verwendet. Atlassian wirbt sogar damit, Confluence als Intranet zu verwenden[85]. Anwender vertrauen dem Intranet[86]. Daher steigt das Risiko, dass Mitarbeiter auf Phishing-Nachrichten (siehe Kapitel 2.4.5) hereinfallen, wenn sie in Confluence veröffentlicht werden.

Da Windows und OSX den Dateityp anhand der Dateiendung identifizieren, E-Mail und Web aber den überlieferten Content-Type (siehe Kapitel 2.3.3) verwenden, ist es wichtig, dass Content-Type und Dateiendung einer übermittelten Datei zueinanderpassen; nur wenn Content-Type und die Dateiendung überprüft werden, kann verhindert werden, dass der Filter umgangen werden kann[87][88]. Daher wird zunächst geschaut, ob der übermittelte Content-Type erlaubt ist, danach wird die Dateiendung überprüft. Für den Fall, dass die Endung nicht in der Whitelist gespeichert ist, wird eine passende Endung angehängt.

Die standardmäßig eingestellte Whitelist erlaubt die bekanntesten sicheren Bildtypen, Audiotypen, Videotypen, Textdateien und PDFs. Bei der Erstellung der Liste wurde sich an den Standardeinstellungen des open-source Wiki Dokuwiki, einer, nach Erfahrung des

Praxis – Sicherheitsmaßnahmen

Autors, nützlichen Software, orientiert[89]. Zusätzlich sind MS-Office-Dateien mit den Endungen .xlsx, .docx und .pptx erlaubt, da sie keine Makros enthalten dürfen[90].

Administratoren können die Whitelist in den Einstellungen bearbeiten. Dabei können sie auch potenziell gefährliche Dateitypen erlauben, eine, mach Meinung des Autors, prominent platzierte Warnung informiert sie jedoch über die Risiken. Die Eingabe erfolgt als tab-separiertes CSV in einem Textfeld, das beim Speichern validiert wird.

File Types

Some attachments in emails may contain malicious code. To prevent malware from being spread through Confluence only relatively safe file types are allowed by default. Windows uses file extensions to categorize files, the web uses [mime types](#). To prevent bypassing of filters it is important that file extensions and mime types match. You can edit the list of allowed file extensions/mime types to allow additional file types or to enforce more restrictions.

Allowing the following file types poses a large security risk:

- Executable files like .exe, .bat, .sh, .jar, ...
- Documents that can contain macros like .doc, .xls, .ppt, ...
- Archives, because they can carry dangerous files, like .zip, .tar, .7z, ...

File Extension	Mime Type
jpg	image/jpeg
jpeg	image/jpeg
gif	image/gif
png	image/png
ico	image/vnd.microsoft.icon
mp3	audio/mpeg
ogg	audio/ogg
wav	audio/wav
webm	video/webm
ogv	video/ogg
mp4	video/mp4
pdf	application/pdf
docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document
xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation
txt	text/plain

Abbildung 3.20: Einstellungen des Dateityp-Filters

Praxis – Sicherheitsmaßnahmen

Listing 3.6: Malware – Virus-Test-Skript getarnt als image/png – E-Mail

```
1 From: attacker@example.org
2 To: victim@example.org
3 Subject: EICAR
4 Content-Type: multipart/mixed; boundary=BOUNDARY
5 Date: Wed, 2 Aug 2017 10:12:18 +0200
6 MIME-Version: 1.0
7
8 --BOUNDARY
9 Content-Transfer-Encoding: 7bit
10 Content-Type: text/plain; charset="us-ascii"
11
12 Bitte öffnen Sie den Anhang.
13 --BOUNDARY
14 Content-Transfer-Encoding: 7bit
15 Content-Disposition: inline; filename="EICAR.COM"
16 Content-Type: image/png; x-unix-mode=0644; name="EICAR.COM"
17 Content-ID: <c588365f-181c-4735-b1ec-97e4a7e3dfe9>
18
19 X5O!P%@AP[4\PZX54(P^)7CC)7}{$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
20 --BOUNDARY--
```

Listing 3.7: Malware – Virus-Test-Skript getarnt als image/png – HTTP

```
1 GET /confluence/download/attachments/819201/EICAR.COM.png?api=v2 HTTP/1.1
2 Host: localhost:1990
3 Accept: */*
4
5 HTTP/1.1 200 OK
6 Content-Disposition: inline; filename="EICAR.COM.png"
7 Content-Type: image/png; charset=UTF-8
8 Content-Length: 69
9 Date: Fri, 11 Aug 2017 21:26:09 GMT
10
11 X5O!P%@AP[4\PZX54(P^)7CC)7}{$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

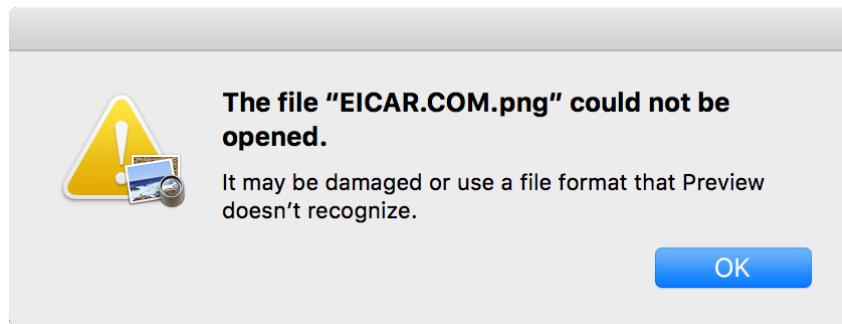


Abbildung 3.21: Malware – Virus-Test-Skript getarnt als image/png – OS X

›  EICAR.COM.png

Abbildung 3.22: Malware – Virus-Test-Skript getarnt als image/png – Confluence

3.5.3.1 Angriffsszenario

Ein Beispiel für eine Malware aus dem letzten Jahr stellt „GoldenEye“ da. Sie wurde als Bewerbungen getarnt und per E-Mail an Personalabteilungen gesendet. Häufig waren diese Bewerbungen in fehlerfreiem Deutsch formuliert und explizit an ausgeschriebene Stellen angepasst (siehe Abbildung 3.23). Die Angreifer haben sich, nach Analyse des Autors, gezielt Unternehmen mit unzureichender IT-Sicherheit ausgesucht, häufig kleine Unternehmen, die sich keinen hochwertigen Support leisten konnten und deswegen unzureichend geschützt waren. Die Malware verschlüsselte alle Dateien auf dem infizierten Rechner und auch auf allen Netzwerkspeichern, die sie erreichen konnte. Die Opfer wurden anschließend aufgefordert ein Lösegeld für die Entschlüsselung der Daten, zu zahlen. Im E-Mail Anhang befand sich eine Office-Datei, mit einem Makro, das die Malware installiert und ausgeführt hat. Für den Fall, dass das automatische Ausführen von Makros nicht erlaubt war, war in der Datei eine Erklärung hinterlegt, welche die Opfer dazu bewegen sollte, das Makro auszuführen (siehe Abbildung 3.25). [91]

Praxis – Sicherheitsmaßnahmen



Abbildung 3.23: Malware – Goldeneye – E-Mail



Abbildung 3.24: Malware – Goldeneye – Excel-Datei mit Schadcode

Praxis – Sicherheitsmaßnahmen

The screenshot shows a Confluence blog entry. The title is "Bewerbung als Rechtsanwaltsfachangestellter". The content starts with "Sehr geehrte Damen und Herren," followed by a message about applying for a position as a legal assistant. It ends with "Mit freundlichen Grüßen" and "Rolf Drescher". On the left, there's a sidebar with icons for search, create, and user management. Below the sidebar, there's a section for attachments with one PDF file listed. At the bottom, there are like and label options.

Abbildung 3.25: Malware – Goldeneye – Blog-Eintrag ohne Excel-Datei

3.5.4 Absender-Filter

Beim Absenderfilter handelt es sich um eine Anti-SPAM-Maßnahme. In den Einstellungen können Administratoren eine Confluence-Nutzergruppe einstellen. Mail2Blog akzeptiert dann nur E-Mails, deren Absenderadresse mit einem Confluence-Nutzer übereinstimmt, der in der entsprechenden Gruppe ist. Die Gruppe „confluence-users“ kann verwendet werden, um die Absenderadressen auf alle Confluence-Nutzer zu beschränken[92].

The plugin tries to use the senders mail address to identify a confluence user. To fight spamming you may choose a confluence group to restrict the accepted senders addresses.

! Warning: The sender address can be easily forged, do not rely on this for security reasons.

Group ▼

Choose **confluence-users** to allow all registered users

Abbildung 3.26: Einstellungen des Absender-Filters

3.6 Tests

3.6.1 Unit-Tests

Für die einzelnen Komponenten wurden Unit-Tests geschrieben. Die Confluence-API und andere Komponenten werden mit Mockito⁴ und Powermock⁵ nachgeahmt. Um E-Mail-Postfächer zu simulieren, wird das Projekt javamail-mock⁶ verwendet. Folgende Unit-Tests wurden angelegt:

ConfigurationActionTest Testet den Kontroller der Konfigurationsseite (siehe Kapitel 3.4).

FileTypeBucketTest Testet den Dateityp-Filter (siehe Kapitel 3.5.3) und überprüft die Serilisation/Deserialisation des FileTypeBucket (siehe Kapitel 3.3.2).

GlobalStateTest Überprüft, dass GlobalState (siehe Kapitel 3.3.2) die Konfiguration initial vom ConfigurationManager lädt und danach im RAM speichert.

Mail2BlogJobTest Überprüft den Ablauf des Mail2Blog-Job (siehe Kapitel 3.3.1). Stellt sicher, dass für jede E-Mail in einem Postfach eine passende Transaktion erstellt wird.

MailboxTest Testet die Mailbox (siehe Kapitel 3.3.3). Überprüft, dass E-Mails erfolgreich abgeholt werden und dass das Markieren von Nachrichten, unter IMAP und POP3 funktioniert.

MailConfigurationManagerTest Stellt sicher, dass der ConfigurationManager (siehe Kapitel 3.3.2) die richtigen Confluence-Api-Funktionen zum Laden/Sichern der Konfiguration aufruft.

MessageParserTest Stellt sicher, dass der MessageParser (siehe Kapitel 3.3.5.1) Informationen aus Beispiel-E-Mails richtig extrahiert.

⁴<http://site.mockito.org/>

⁵<https://github.com/powermock/powermock>

⁶<https://github.com/salyh/javamail-mock2>

MessageToBlogPostProcessorTest Stellt sicher, dass der MessageToBlogPostProcessor (siehe Kapitel 3.3.5.3) Blog-Posts mit passenden Inhalten erzeugt.

MessageTransactionTest Überprüft den Ablauf der Transaktionen des Mail2Blog-Jobs (siehe Kapitel 3.3.1). Stellt sicher, dass die passenden Funktionen aufgerufen werden.

SpaceKeyExtractionTest Prüft, dass die verschiedenen Methoden zur Bestimmung des Space-Keys (siehe Kapitel 3.3.4) richtig funktionieren.

3.6.2 Integrations-Test

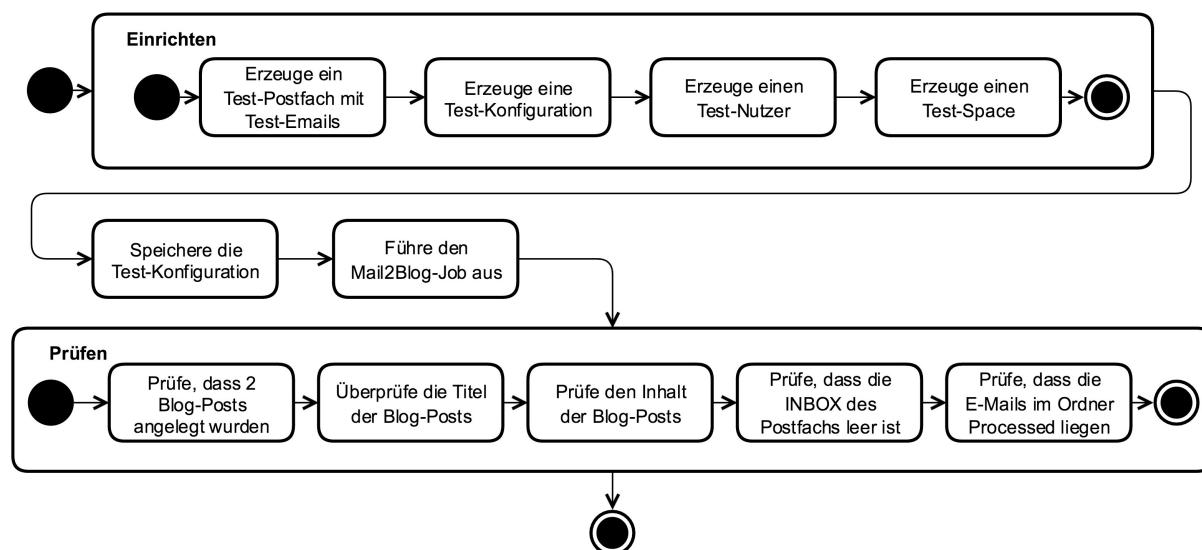


Abbildung 3.27: Aktivitätsdiagramm des Integrationstest

Neben den Unit-Tests gibt es einen Integrations-Test, der den kompletten Ablauf in einer echten Confluence-Instanz testet. Der Integrationstest nutzt den von Confluence bereitgestellte Runner AtlassianPluginsTestRunner um die Tests auszuführen. Mit dem Confluence SDK (siehe Kapitel 2.2.4) besteht die Möglichkeit eine Testinstanz automatisch erstellen zu lassen und darin den Test auszuführen. Alternativ kann der Test auch in einer bestehenden Instanz ausgeführt werden.

3.7 Veröffentlichung

Der Quellcode des Add-ons wurde auf Github veröffentlicht⁷, der Binärkode wird über den Atlassian-Marketplace verbreitet⁸. Die Dokumentation ist im Github-Repo in Markdown hinterlegt. Der Maven-Build-Vorgang wird automatisch nach jedem Commit von Travis-CI⁹ durchgeführt und die entstandenen Binärpakete auf Github veröffentlicht. Ebenfalls wird nach dem Build-Vorgang die Test-Abdeckung mithilfe von codecov.io¹⁰ berechnet.

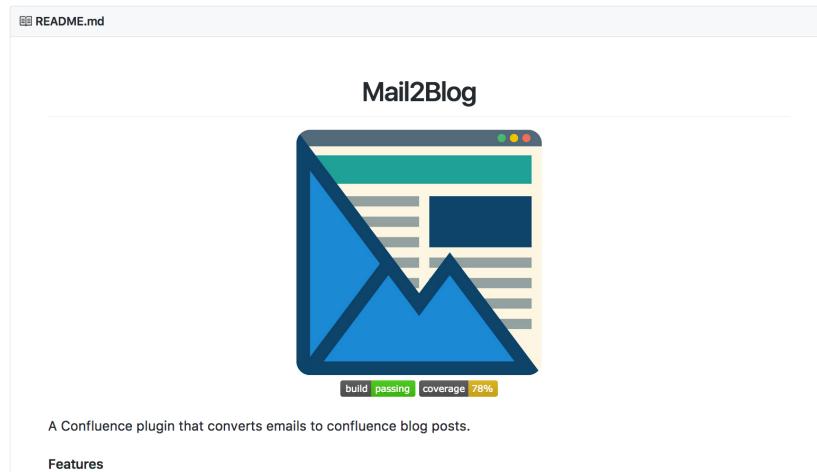


Abbildung 3.28: Github



Abbildung 3.29: Travis-CI

⁷<https://github.com/dm-drogeriemarkt/Mail2Blog>

⁸<https://marketplace.atlassian.com/plugins/de.dm.mail2blog.mail2blog/server/overview>

⁹<https://travis-ci.org/dm-drogeriemarkt/Mail2Blog>

¹⁰<https://codecov.io/gh/dm-drogeriemarkt/Mail2Blog>

4 Fazit und Ausblick

Der Workflow zur Erstellung eines Atlassian Add-ons wurde am Beispiel Mail2Blog bis zur Veröffentlichung (siehe Kapitel 3.7) in dieser Arbeit dokumentiert. Die wichtigsten Komponenten eines Add-ons für Atlassian-Produkte wurden im Kapitel 2.2.4 beschrieben. Eine mögliche Strukturierung des Java-Quellcodes eines Add-ons wurde in Kapitel 3.3 beispielhaft aufgezeigt.

Das fertige Add-On erstellt erfolgreich Blog-Beiträge für Confluence (siehe Kapitel 2.2) aus E-Mails (siehe Kapitel 2.3), dabei wird der MIME-Standard (erläutert in Kapitel 2.3.3), HTML-E-Mails und Text-Emails unterstützt. Das Sicherheitsumfeld wurde in Kapitel 2.4 untersucht und im Kapitel 3.5 wurden die getroffenen Sicherheitsmaßnahmen erläutert. Zur Konfiguration wurde eine Konfigurationsseite erstellt (siehe Kapitel 3.4), auf der alle Einstellungen des Add-ons durchgeführt werden können. Um die Qualität sicherzustellen, wurden Unit- und Integrations-Tests (siehe 3.6.1) für das Add-On geschrieben und dem Projekt auf GitHub eine Dokumentation in Markdown beigelegt (siehe Kapitel 3.7). Damit erfüllt das fertige Add-on den in der Motivation (siehe Kapitel 1.3) beschriebenen Zweck und die im Kapitel 3.1 festgelegten Anforderungen.

Für die Weiterentwicklung bleibt die Rückmeldung der OpenSource-Veröffentlichung abzuwarten. In der Zukunft wäre z. B. die Unterstützung mehrerer Postfächer denkbar. Um die Sicherheit gegen Phishing (siehe Kapitel 2.4.5) weiter zu erhöhen, könnte ein URL-Filter in das Add-on eingebaut werden.

Literatur

- [1] dm-drogerie markt GMBH + Co. KG. *Unternehmenszahlen*. <https://www.dm.de/unternehmen/ueber-uns/zahlen-und-fakten/> [siehe S. 1].
- [2] Ward CUNNINGHAM. *Correspondence on the Etymology of Wiki*. <http://c2.com/doc/etymology.html>. 2003 [siehe S. 3].
- [3] L. GRAY und H. KILLCOYNE. *What Is a Wiki and How Do I Use It?* Practical Technology. Rosen Publishing Group, Incorporated, 2014, S. 5. ISBN: 9781622750719 [siehe S. 3].
- [4] WIKIPEDIA. *Wikipedia:Size in volumes — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Wikipedia:Size_in_volumes. Online; Stand 24. August 2017. 2017 [siehe S. 3].
- [5] WIKIPEDIA. *Wikipedia:Size in volumes — Wikipedia, Die freie Enzyklopädie*. <https://de.wikipedia.org/wiki/Wikipedia:Statistik/B%C3%BCcherregal>. Online; Stand 24. August 2017. 2017 [siehe S. 3].
- [6] ALEXA. *The top 500 sites on the web*. <https://www.alexa.com/topsites/global>. [Online; Stand 24. August 2017]. 2017 [siehe S. 3].
- [7] M. SEIBERT, S. PREUSS und M. RAUER. *Enterprise Wikis: Die erfolgreiche Einführung und Nutzung von Wikis in Unternehmen*. Gabler Verlag, 2011, S. 67–73. ISBN: 9783834967862 [siehe S. 3].
- [8] ATLASSIAN. *Confluence*. <https://www.atlassian.com/software/confluence> [siehe S. 5].
- [9] S. MADDOX und R. MADDOX. *Confluence, Tech Comm, Chocolate: A Wiki as Platform Extraordinaire for Technical Communication*. Chandos social media series. XML Press, 2012, S. 22 –25. ISBN: 9781937434007 [siehe S. 5].
- [10] ATLASSIAN. *Atlassian Support – Confluence Server – Documentation – Confluence administrator's guide – Linking to Another Application*. <https://confluence.atlassian.com/doc/linking-to-another-application-360677690.html> [siehe S. 5].

Literatur

- [11] ATlassian. *Atlassian Support – Confluence 6.3 – Documentation – Confluence Release Notes*. <https://confluence.atlassian.com/doc/confluence-6-3-release-notes-909642701.html> [siehe S. 5].
- [12] ATlassian. *Atlassian Support – Confluence 6.3 – Documentation – Pages and blogs*. <https://confluence.atlassian.com/doc/pages-and-blogs-320602215.html> [siehe S. 5].
- [13] ATlassian. *Atlassian Support – Confluence 6.3 – Documentation – Spaces*. <https://confluence.atlassian.com/doc/spaces-139459.html> [siehe S. 6].
- [14] ATlassian. *Atlassian Support – Confluence 6.3 – Documentation – Pages and blogs – Confluence Markup – Confluence Storage Format*. <https://confluence.atlassian.com/doc/confluence-storage-format-790796544.html> [siehe S. 7].
- [15] ATlassian. *Confluence 4 Editor FAQ*. <https://confluence.atlassian.com/display/CONF40/Confluence+4+Editor+FAQ> [siehe S. 8].
- [16] ATlassian. *Atlassian Developers – Enabling TinyMCE Plugins*. <https://developer.atlassian.com/confdev/confluence-plugin-guide/writing-confluence-plugins/enabling-tinymce-plugins> [siehe S. 8].
- [17] ATlassian. *Atlassian Marketplace*. <https://marketplace.atlassian.com/search?product=confluence> [siehe S. 9].
- [18] ATlassian. *Atlassian Developers – Writing Confluence Plugins*. <https://developer.atlassian.com/confdev/confluence-plugin-guide/writing-confluence-plugins> [siehe S. 9].
- [19] J. KURUVILLA. *JIRA Development Cookbook*. Quick answers to common problems. Packt Publishing, 2016, S. 70 –76. ISBN: 9781785886331 [siehe S. 9].
- [20] J. KURUVILLA. *JIRA Development Cookbook*. Quick answers to common problems. Packt Publishing, 2016, S. 77 –81. ISBN: 9781785886331 [siehe S. 9].
- [21] J. KURUVILLA. *JIRA Development Cookbook*. Quick answers to common problems. Packt Publishing, 2016, S. 140 –144. ISBN: 9781785886331 [siehe S. 9].
- [22] ATlassian. *Atlassian Developers – Job Config Module*. <https://developer.atlassian.com/confdev/confluence-plugin-guide/confluence-plugin-module-types/job-config-module> [siehe S. 10].
- [23] M. FOWLER. *Patterns für Enterprise-Application-Architekturen*. Software-Engineering. mitp-Verlag, 2003, 365 ff. ISBN: 9783826613784 [siehe S. 11].
- [24] ATlassian. *Atlassian Developers – XWork-WebWork Module*. <https://developer.atlassian.com/confdev/confluence-plugin-guide/confluence-plugin-module-types/xwork-webwork-module> [siehe S. 11].
- [25] Apache Software FOUNDATION. *Velocity – Overview*. <http://velocity.apache.org/engine/2.0/overview.html> [siehe S. 12].

Literatur

- [26] B. KURNIAWAN. *Struts 2 Design and Programming: A Tutorial*. A Tutorial Series. Brainysoftware.com, 2007, S. 5–6. ISBN: 9780980331608 [siehe S. 12].
- [27] S. MADDOX und R. MADDOX. *Confluence, Tech Comm, Chocolate: A Wiki as Platform Extraordinaire for Technical Communication*. Chandos social media series. XML Press, 2012, S. 297–300. ISBN: 9781849689526 [siehe S. 12].
- [28] ATlassian. *Atlassian Developers – Enabling XSS Protection in Plugins*. <https://developer.atlassian.com/confdev/development-resources/confluence-architecture/anti-xss-documentation/enabling-xss-protection-in-plugins> [siehe S. 12].
- [29] J. KURUVILLA. *JIRA Development Cookbook*. Quick answers to common problems. Packt Publishing, 2016, S. 190 –208. ISBN: 9781785886331 [siehe S. 12].
- [30] ATlassian. *Atlassian Developers – XWork Plugin Complex Parameters and Security*. <https://confluence.atlassian.com/display/CONF30/XWork+Plugin+Complex+Parameters+and+Security>. 2008 [siehe S. 12].
- [31] Dave CROCKER. *Email History*. <http://www.livinginternet.com/e/ei.htm>. 2000 [siehe S. 17].
- [32] Martin WILHELM. *2016 Rekordjahr für E-Mail*. <https://newsroom.web.de/2017/02/13/2016-rekordjahr-fuer-e-mail/>. 2017 [siehe S. 17].
- [33] Peter W. RESNICK. *Internet Message Format*. RFC 5322. RFC Editor, 2008. URL: <http://www.rfc-editor.org/rfc/rfc5322.txt> [siehe S. 17, 18].
- [34] D. J. BERNSTEIN. *Variable Envelope Return Paths*. Techn. Ber. 1997. URL: <http://cr.yp.to/proto/verp.txt> [siehe S. 17].
- [35] *Plus signs ("+") in email addresses*. <https://www.cs.rutgers.edu/~watrous/plus-signs-in-email-addresses.html> [siehe S. 17].
- [36] Ned FREED und Nathaniel S. BORENSTEIN. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. RFC 2046. RFC Editor, 1996. URL: <http://www.rfc-editor.org/rfc/rfc2046.txt> [siehe S. 19].
- [37] Keith MOORE. *MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text*. RFC 2047. RFC Editor, 1996. URL: <http://www.rfc-editor.org/rfc/rfc2047.txt> [siehe S. 19].
- [38] N. FREED und J. KLENSIN. *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*. BCP 13. RFC Editor, 2005. URL: <http://www.rfc-editor.org/rfc/rfc4289.txt> [siehe S. 19].
- [39] Ned FREED und Nathaniel S. BORENSTEIN. *Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples*. RFC 2049. RFC Editor, 1996. URL: <http://www.rfc-editor.org/rfc/rfc2049.txt> [siehe S. 19].

Literatur

- [40] John G. MYERS und Marshall T. ROSE. *Post Office Protocol - Version 3*. STD 53. RFC Editor, 1996. URL: <http://www.rfc-editor.org/rfc/rfc1939.txt> [siehe S. 23].
- [41] M. CRISPIN. *Internet Message Access Protocol - Version 4*. RFC 1730. RFC Editor, 1994 [siehe S. 23].
- [42] Chris NEWMAN. *Using TLS with IMAP, POP3 and ACAP*. RFC 2595. RFC Editor, Juni 1999. URL: <http://www.rfc-editor.org/rfc/rfc2595.txt> [siehe S. 23].
- [43] Chris NEWMAN. *Using TLS with IMAP, POP3 and ACAP*. RFC 2595. RFC Editor, Juni 1999, S. 10. URL: <http://www.rfc-editor.org/rfc/rfc2595.txt> [siehe S. 23].
- [44] Jacob HOFFMAN-ANDREWS. *ISPs Removing Their Customers' Email Encryption*. <https://www.eff.org/deeplinks/2014/11/starttls-downgrade-attacks>. 2014 [siehe S. 23].
- [45] intel SECURITY – MCAFEE. *Net Losses: Estimating the Global Cost of Cybercrime*. <https://www.mcafee.com/de/resources/reports/rp-economic-impact-cybercrime2.pdf>. 2014 [siehe S. 24].
- [46] Z. TRABELSI u. a. *Network Attacks and Defenses: A Hands-on Approach*. CRC Press, 2012, 89 ff. ISBN: 9781466517974 [siehe S. 24].
- [47] V. BHARADWAJ. *WIRESHARK: The Packet Sniffer*. VINEET BHARADWAJ, S. 6 [siehe S. 24].
- [48] R. OPPLIGER. *Security Technologies for the World Wide Web*. Artech House computer security series. Artech House, 2003, 153 ff. ISBN: 9781580535854 [siehe S. 25].
- [49] J. JOSHI. *Network Security: Know It All*. Newnes Know It All. Elsevier Science, 2008, S. 13–15. ISBN: 9780080560151 [siehe S. 26].
- [50] S. JACOBS. *Engineering Information Security: The Application of Systems Engineering Concepts to Achieve Information Assurance*. IEEE Press Series on Information and Communication Networks Security. Wiley, 2015, S. 574. ISBN: 9781119104797 [siehe S. 27].
- [51] L. LOBO und U. LAKSHMAN. *CCIE Security v4.0 Quick Reference*. Quick Reference. Pearson Education, 2014, S. 69–70. ISBN: 9780133855111 [siehe S. 27].
- [52] J. STAPLETON und W.C. EPSTEIN. *Security without Obscurity: A Guide to PKI Operations*. CRC Press, 2016, S. 25. ISBN: 9781498707480 [siehe S. 28].
- [53] N. DHANJANI. *Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts*. O'Reilly Media, 2015, 150 ff. ISBN: 9781491902936 [siehe S. 28].
- [54] P.S. SAJJA und R. AKERKAR. *Intelligent Technologies for Web Applications*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press, 2016, S. 296. ISBN: 9781439871645 [siehe S. 28].

Literatur

- [55] B. SULLIVAN und V. LIU. *Web Application Security, A Beginner's Guide*. Beginner's Guide. McGraw-Hill Education, 2011, S. 13. ISBN: 9780071776127 [siehe S. 28].
- [56] B. BRINZAREA und A. HENDRIX. *AJAX and PHP: Building Modern Web Applications*. From technologies to solutions. Packt Publishing, 2009, S. 220. ISBN: 9781847197726 [siehe S. 29].
- [57] T.P. SCHÄFERS. *Hacking im Web: Denken Sie wie ein Hacker und schließen Sie die Lücken in Ihrer Webapplikation, bevor diese zum Einfallstor für Angreifer wird*. Hacking. Franzis Verlag, 2016, S. 109. ISBN: 9783645603768 [siehe S. 29].
- [58] WhiteHat SECURITYYY. *Application Security Statistics Report*. <https://www.whitehatsec.com/resources-category/premium-content/web-application-stats-report-2017>. 2017 [siehe S. 29].
- [59] ACUNETIX. *Acunetix Web Application Vulnerability Report*. <https://www.acunetix.com/acunetix-web-application-vulnerability-report-2016/>. 3016 [siehe S. 29].
- [60] C. TIMM und R. PEREZ. *Seven Deadliest Social Network Attacks*. Syngress seven deadliest attacks series. Elsevier Science, 2010, S. 27. ISBN: 9781597495462 [siehe S. 29].
- [61] Martin WILHELM. *Spam-Aufkommen in Deutschland*. <https://newsroom.web.de/2016/02/08/spam-aufkommen-in-deutschland/>. 2016 [siehe S. 29].
- [62] H.F. TIPTON und M. KRAUSE. *Information Security Management Handbook, Fifth Edition*. v. 3. CRC Press, 2006, S. 560. ISBN: 9781420003406 [siehe S. 30].
- [63] L.F. CRANOR und S. GARFINKEL. *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly Media, 2005, 281 ff. ISBN: 9780596553852 [siehe S. 30].
- [64] Vincent LYNCH. *PayPal Phishing Certificates Far More Prevalent Than Previously Thought*. <https://www.thesslstore.com/blog/lets-encrypt-phishing/>. 2017 [siehe S. 30].
- [65] Let's ENCRYPT. *The CA's Role in Fighting Phishing and Malware*. <https://letsencrypt.org/2015/10/29/phishing-and-malware.html>. 2015 [siehe S. 30].
- [66] R.L. KRUTZ und R.D. VINES. *The CEH Prep Guide: The Comprehensive Guide to Certified Ethical Hacking*. Wiley, 2007, S. 271. ISBN: 9780470135921 [siehe S. 31].
- [67] N. CLARKE und S. FURNELL. *Proceedings of the Eighth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2014)*. Centre for Security, Communications & Network Research, Plymouth University, 2014, S. 44. ISBN: 9781841023755 [siehe S. 31].
- [68] Federal Bureau of INVESTIGATION. *2016 Internet Crime Report*. https://pdf.ic3.gov/2016_IC3Report.pdf. 2016 [siehe S. 31].

Literatur

- [69] E. SKOUDIS und L. ZELTSER. *Malware: Fighting Malicious Code*. Prentice Hall Series in Comput. Prentice Hall PTR, 2004, S. 3. ISBN: 9780131014053 [siehe S. 32].
- [70] VERIZON. *2017 Data Breach Investigations Report*. http://www.verizonenterprise.com/resources/reports/rp_DBIR_2017_Report_en_xg.pdf. 2017 [siehe S. 32].
- [71] SYMANTEC. *Ransomware and Businesses 2016*. https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/ISTR2016_Ransomware_and_Businesses.pdf. 2016 [siehe S. 32].
- [72] BBC NEWS. *New ransomware strain coded entirely in Javascript*. <http://www.bbc.com/news/technology-36575687>. 2016 [siehe S. 32].
- [73] PHISHME. *Enterprise Phishing Susceptibility and Resiliency Report*. http://storage.pardot.com/46382/128954/PhishMe_Enterprise_Phishing_Susceptibility_and_Resiliency_Report_2016.pdf. 2016 [siehe S. 32].
- [74] Artemis SOFTWARE. *Send Email To Page Plugin*. <https://marketplace.atlassian.com/plugins/biz.artemissoftware.plugins.confluence.send-email-to-page/server/overview> [siehe S. 34].
- [75] Artemis SOFTWARE. *Send Email To Page Plugin*. <https://marketplace.atlassian.com/plugins/biz.artemissoftware.plugins.confluence.send-email-to-page/server/pricing> [siehe S. 34].
- [76] Stimmt AG. *Mail to News*. <https://marketplace.atlassian.com/plugins/com.midori.confluence.plugin.mail2news/server/overview> [siehe S. 34].
- [77] Stimmt AG. *LICENSE.txt*. <https://github.com/stimmt/Confluence-Mail-to-News-Plugin/blob/master/LICENSE.txt> [siehe S. 34].
- [78] L. PRECHELT, G. MALPOHL und M. PHILIPPSEN. *JPlag: Finding Plagiarisms Among a Set of Programs*. 2000. URL: <https://publikationen.bibliothek.kit.edu/542000/759910> [siehe S. 36].
- [79] B. PRIBYL und S. FEUERSTEIN. *Learning Oracle PL/SQL*. Learning Series. O'Reilly Media, 2002, S. 113. ISBN: 9780596001803 [siehe S. 49].
- [80] Joseph Clark (Atlassian EMPLOYEE). *RE: Which HTML tags are valid to PUT using Confluence API?* <https://community.atlassian.com/t5/Answers-Developer-Questions/Which-HTML-tags-are-valid-to-PUT-using-Confluence-API/qaq-p/464921> [siehe S. 52].
- [81] Open Web Application Security PROJECT. *OWASP HTML Sanitizer Project*. https://www.owasp.org/index.php/OWASP_Java_HTML_Sanitizer_Project [siehe S. 52].
- [82] Open Web Application Security PROJECT. *Class Sanitizers*. <https://rawgit.com/OWASP/java-html-sanitizer/master/distrib/javadoc/org/owasp/html/Sanitizers.html> [siehe S. 52].

Literatur

- [83] ATlassian. *Atlassian Documentation – Confluence 6.3 – HTML Macro*. <https://confluence.atlassian.com/doc/html-macro-38273085.html> [siehe S. 53].
- [84] Jodson SANTOS. *Persisted Cross-Site Scripting (XSS) in Confluence Jira Software*. <http://seclists.org/fulldisclosure/2017/Jan/3>. Jan. 2017 [siehe S. 55].
- [85] ATlassian. *Atlassian Support – Confluence 6.3 – Documentation – Confluence use-cases – Use Confluence as your Intranet*. <https://confluence.atlassian.com/doc/use-confluence-as-your-intranet-230819424.html> [siehe S. 57].
- [86] M. NIEMELÄ. *Anatomy of a cyberattack*. BookBaby, 2016. Kap. Intranet – world's most common name for intranet. ISBN: 9781483562100 [siehe S. 57].
- [87] B. GILMER. *File Interchange Handbook: For Professional Images, Audio and Metadata*. Taylor & Francis, 2012, S. 291. ISBN: 9781136037146 [siehe S. 57].
- [88] M. HEIDERICH. *Sichere Webanwendungen: das Praxishandbuch ; [Web 2.0-Sicherheit, sichere PHP-, JavaScript- und Flash-Anwendungen, XSS, CSRF, Remote Code Execution, SQL Injection u.v.m., Angriffstechniken verstehen und Sicherheitslücken vermeiden]*. Galileo computing. Galileo Press, 2009, S. 315. ISBN: 9783836211949 [siehe S. 57].
- [89] DOKUWIKI. *mime.conf*. <https://github.com/splitbrain/dokuwiki/blob/master/conf/mime.conf> [siehe S. 58].
- [90] H. Bos, F. MONROSE und G. BLANC. *Research in Attacks, Intrusions, and Defenses: 18th International Symposium, RAID 2015, Kyoto, Japan, November 2-4, 2015. Proceedings*. Lecture Notes in Computer Science. Springer International Publishing, 2015, S. 229. ISBN: 9783319263625 [siehe S. 58].
- [91] Fabian A. SCHERSCHEL. *Goldeneye Ransomware greift gezielt Personalabteilungen an*. <https://www.heise.de/security/meldung/Goldeneye-Ransomware-greift-gezielt-Personalabteilungen-an-3562281.html>. 2016 [siehe S. 60].
- [92] ATlassian. *Atlassian Support – Confluence 5.7 – Documentation – Confluence 101 – Add users and set permissions*. <https://confluence.atlassian.com/conf57/add-users-and-set-permissions-701434560.html#Addusersandsetpermissions-PermissionsandGroups> [siehe S. 62].

Versionshistorie

Version	Datum	Autor(en)	Änderungen
	18.09.17	SG	Die Arbeit wurde unter dem Titel „Konzeption und Einsatz eines Prozesses zur Erweiterung von Atlassian-Produkten durch Eigenentwicklungen“ an der DHBW-KA eingereicht.
1.0	18.09.17	SG	Neues Deckblatt für die Online-Publikation. Einfügung des Hinweis zur Revision. Löschung des Begriffs „dreimal“ aus dem Glossar. Kleinere redaktionelle Änderungen.