

Verfahren und zugehörige Computersysteme zur Sicherung der Integrität von Daten

Technisches Gebiet

Die vorliegende Erfindung bezieht sich im Allgemeinen auf ein Verfahren und entsprechende
5 computerbasierte Vorrichtungen zur Sicherung der Integrität von Daten zum Betrieb eines
Service. Die Erfindung bezieht sich dabei insbesondere auf ein Verfahren zum Erzeugen
eines intendierten Zustandes von Daten, die in einem Medium oder System gespeichert oder
gesichert werden können, zum Übertragen des intendierten Zustands der Daten an ein
weiteres System, zum Sichern der Integrität der Daten auf dem weiteren System, und zum
10 Ermöglichen der späteren Analyse nach einem unerwünschten Manipulieren der Daten. Die
Erfindung bezieht sich weiterhin auf computerbasierte Vorrichtung, Computersysteme, oder
ein Computernetzwerk, wobei diese das Verfahren bzw. die einzelnen Verfahrensschritte
zumindest teilweise oder vollständig ausführen, sowie ein computerlesbares Medium mit
computerausführbaren Anweisungen.

15

Stand der Technik

Die vorliegende Erfindung bezieht sich auf das technische Gebiet der Datenverarbeitung,
insbesondere handelt es sich um eine Erfindung aus dem Feld der IT-Sicherheit. Auf diesem
technischen Gebiet der Datenverarbeitung sind verschiedene Techniken aus dem Stand der
20 Technik bekannt.

Sicherheit in der IT muss dabei insbesondere weniger als ein Zustand und viel mehr als ein
Prozess verstanden werden, da es sowohl auf Seiten der Angreifer als auch auf Seiten der
Anwender, die ihre Systeme schützen wollen, immer wieder zu neuen Entwicklungen kommt,
die sich durch das Internet schnell verbreiten.

25

Zur Absicherung der Kommunikation von Computern über Netzwerke haben sich
verschiedene Methoden etabliert.

Ein Computer ist eine physische Repräsentation einer universellen Turingmaschine. Auf ihm
wird ein Computerprogramm betrieben, das Grundfunktionen wie Dateizugriff, (visuelle)
30 Benutzeroberflächen ((graphic) user interface, (G)UI) oder auch Netzwerk-Funktionalitäten
bereitstellt (Betriebssystem/ Operating System, OS). Das Betriebssystem wiederum erlaubt
den Betrieb weiterer Software, die auf einer anderen Abstraktionsebene geschrieben werden
kann.

Ein Computer besteht in der Regel aus mindestens einer Recheneinheit (central processing unit, CPU), einem Hauptspeicher (Arbeitsspeicher/RAM) und einem Permanentenspeicher. LU500837

Auch sogenannte Smartphones oder IoT (Internet of Things) Devices, wie bspw. Webcams, smarte Leuchtmittel, Thermostate, aber auch Waschmaschinen, Geschirrspüler usw. können

5 Computer nach dieser Definition sein.

Ein Netzwerk bezeichnet einen Verbund aus logisch voneinander getrennten Computersystemen, die miteinander kommunizieren können. Werden mehrere Computersysteme auf einem physischen Computer betrieben und können diese Computersysteme untereinander kommunizieren, bezeichnet man dies auch als ein

10 Netzwerk.

Möchte man zwischen einem Netzwerk von Computern und einem Netzwerk von Computersystemen innerhalb eines Computers unterscheiden, spricht man bei ersterem von einem physischen Netzwerk und bei letzterem von einem virtuellen Netzwerk. Ein virtuelles Netzwerk (virtual local area network, VLAN) kann auch über ein physisches Netzwerk

15 zwischen Computersystemen, die über mehrere Computer verteilt sind, hergestellt werden.

Netzwerke werden vor allem zum Austausch von Informationen zwischen Computern verwendet. Dieser Informationsaustausch findet durch verschiedene (Web-)Services statt.

Ein (Web-)Service besteht aus mindestens einem Computerprogramm, das mindestens eine Funktionalität über ein Netzwerk bereitstellt. Dabei kann es sich auch um ein virtuelles

20 Netzwerk innerhalb eines Hostcomputers handeln. Üblicherweise offenbaren Webservices Informationen. Die Worte Dienst, Webdienst oder Internetdienst werden synonym zu Webservices verwendet. Webdienste zeichnen sich in der Regel dadurch aus, dass sie über bestimmte, vordefinierte Protokolle wie bspw. Hypertext transfer protocol (HTTP) ansprechbar sind.

25 Prinzipiell gibt es verschiedene Möglichkeiten zum Betrieb und zur Skalierung eines Webservice. Der offensichtliche Weg wäre vermutlich, einen Computer in einem Netzwerk mit einem Betriebssystem zu betreiben und auf diesem Computer einen Webservice zu betreiben.

30 Zur Skalierung könnte man dann zunächst den Computer aufrüsten, bis er auf dem Maximum der Leistungsfähigkeit der jeweilig aktuellen Hardwaregeneration wäre.

Aus verschiedensten Gründen haben sich unterschiedliche Mittel der Virtualisierung, insbesondere virtuelle Maschinen und sogenannte Container (auch Jails genannt) durchgesetzt.

Ein Computersystem bezeichnet daher eine beliebige Konfiguration (Computer mit Betriebssystem, virtuelle Maschine, Container o.Ä.), welche die notwendigen Voraussetzungen zum Betrieb eines Webservices (i.d.R. Bereitstellung eines Dateisystems, einer Netzwerkfunktionalität und Betrieb von Software) erfüllt.

- 5 Ein Computersystem, das einen Webservice in einem Netzwerk bereitstellt, nennt man Server oder Webserver.

Ein Netzwerk, bei dem man davon ausgehen muss, dass ein potenzieller Angreifer in diesem Netzwerk Zugriff auf einen anderen Computer hat, sodass der Angreifer beliebige Daten über das Netzwerk senden und so ggf. das eigene Computersystem angreifen kann, ist
10 unsicher. Insbesondere ist das Internet ein unsicheres Netzwerk.

Zur Absicherung der Kommunikation über ein potenziell unsicheres Netzwerk stehen verschiedenste Techniken zur Verfügung.

- Aus dem Stand der Technik ist beispielsweise eine Technik bekannt, bei der aus Daten ein
15 sogenannter Streuwert (auch als Hash oder Hashwert oder auch Prüfsumme bezeichnet) berechnet wird. Regelmäßig kommt dabei eine sogenannte Streuwertfunktion (Hashfunktion) zum Einsatz. Eine solche Hashfunktion wird beispielsweise durch den sicheren Hash-Algorithmus (SHA, secure hash algorithm) bereitgestellt. Mit einer Hashfunktion kann häufig ein in seiner Größe nicht notwendigerweise beschränkter Datenblock auf einen Datenblock
20 fester Größe, den Hash oder Hashwert, abgebildet werden. Eine typische Länge für einen Hash ist beispielsweise 256 bit. Eine wünschenswerte Eigenschaft einer guten kryptologischen Hashfunktion ist eine näherungsweise Injektivität und eine dadurch erzielte näherungsweise Kollisionsresistenz. Die „ideale Hashfunktion“ ist daher vollständig linkseindeutig und kollisionsfrei und bildet unterschiedliche Eingangsdaten immer auf
25 unterschiedliche Hashes ab.

- Das Transmission Control Protocol (TCP) ist ein typischer Anwendungsfall für eine Hashfunktion, wobei ein Hashwert für eine größere Datenmenge berechnet wird. Beispielsweise soll eine größere Datenmenge über ein möglicherweise nicht sicheres Übertragungsmedium, wie das Internet, übertragen werden. Zum Beispiel kann durch einen
30 technischen Fehler ein Teil der Daten falsch übertragen werden oder ein Dritter (Angreifer) manipuliert diese Daten.

Wenn der Sender der Daten ihren Hashwert berechnet und diesen Hashwert dem Empfänger zur Verfügung stellt, kann der Empfänger die Echtheit und Unverfälschtheit der Daten verifizieren, vorausgesetzt der Hashwert selbst ist echt und verlässlich.

Abseits von der Nutzung von Prüfsummen hat sich die Nutzung von Verschlüsselung, insbesondere asymmetrischer Verschlüsselung etabliert. Diese bietet den Vorteil, dass der Schlüsselaustausch über ein unsicheres Netzwerk stattfinden kann. Dabei werden ein privater und ein öffentlicher Schlüssel (public key) erzeugt. Daten, die mit dem öffentlichen Schlüssel verschlüsselt wurden, können ausschließlich mittels des privaten Schlüssels (private key) entschlüsselt werden.

Neben der Überprüfung der Integrität nach einer Übertragung mittels Prüfsummen gibt es noch die Absicherung der Integrität durch Nutzung von Zertifikaten. Diese basiert in der Regel auf der Nutzung asymmetrischer Verschlüsselung, bei der ein Webservice ein Zertifikat bereitstellt, in dem eine vertrauenswürdige Instanz (Certificate Authority) für die Identität des Webservices garantiert (durch Signatur mittels eines Zertifikates). Ein Zertifikat enthält dann meist einen öffentlichen Schlüssel, der für Verschlüsselung der initialen Kommunikation mit dem Webservice genutzt werden kann.

Das Hash-Verfahren bringt jedoch den Nachteil mit sich, dass das Verfahren völlig außer Kraft gesetzt wird, wenn der Hashwert selbst während einer Man-In-The-Middle-Attacke manipuliert wird. In einem Beispiel überträgt ein Benutzer A (z. B. Alice) an einen Benutzer B (z. B. Bob) eine Datei und außerdem den Hashwert der Datei. Ein Angreifer E (z. B. Eve) fängt diese Kommunikation ab und ersetzt die Datei durch eine gefälschte Datei. Die gefälschte Datei enthält gefälschte Daten. Außerdem ersetzt E den Hashwert der Datei durch den Hashwert der gefälschten Datei. Wenn B nun die Daten empfängt und überprüft, indem er die Hashfunktion der Datei selbst berechnet und mit dem Hashwert vergleicht, kommt er zu dem Schluss, dass die Daten nicht manipuliert wurden. So hat E erfolgreich die Daten manipuliert, ohne dass dies von B festgestellt wurde. Dies kann auch durch gängige Verschlüsselungsmethoden wie SSL nicht notwendigerweise festgestellt werden, da auch die zur Verschlüsselung genutzten Zertifikate durch E ersetzt werden können, sodass beide Parteien verschlüsselt mit E kommunizieren und vermuten, die Verbindung sei sicher.

Man-In-The-Middle Attacks lassen sich weitestgehend vermeiden, indem bei der Kommunikation zwischen A und B beide zuvor austauschen, welche Zertifikate sie voneinander erwarten und dadurch ausschließen können, dass ein Angreifer E die Kommunikation manipuliert. Dieser Vorgang wird als Certificate Pinning bezeichnet.

Sobald die Kommunikation zwischen zwei Computern ausreichend geschützt ist, wird es für einen möglichen Angreifer einfacher und somit auch kostengünstiger, eine der Kommunikationsparteien direkt anzugreifen, um die Kommunikation abzufangen bzw. zu manipulieren. Somit ist es unerlässlich auch einzelne Computersysteme hinreichend abzusichern.

Angriffe auf Webservices sind insofern besonders attraktiv für mögliche Angreifer, da sie mitunter durch einen geglückten Angriff Zugriff auf die Kommunikation mit dutzenden, hunderten oder tausenden Kunden des Opfers haben, die die geladenen Daten wiederum für berechtigt halten und ihnen vertrauen.

- 5 Daraus ergibt sich für Betreiber von Webservices ein besonderes Schutzbedürfnis, zumal auch die Konkurrenz des Betreibers bzw. weitere Interessensgruppen wie Hacker mit einer Gewinnabsicht, aber auch Gruppen wie Anonymous, die rein auf Basis eines ideologischen Antriebs Ziele attackieren.

- 10 Zudem ist spätestens mit der Einführung von Datenschutzregulierungen wie der Datenschutzgrundverordnung (DSG-VO), aber auch zuvor in der Handhabung von Zahlungsdaten wie Kreditkartendaten, ein erhebliches wirtschaftliches Schadensrisiko mit dem Betrieb eines Webservices verbunden.

- 15 Allgemein lässt sich sagen, dass bei der Absicherung von Computersystemen ein Ansatz mit möglichst vielen Schutzschichten vonnöten ist, da nahezu jedes Schutzsystem überwunden werden kann, aber insbesondere wirtschaftlich motivierte Angreifer in der Regel zunächst die einfachsten Ziele attackieren, da auch sie einer Gewinnmaximierungsabsicht unterliegen. Viele einfach durchzuführende und erfolgreiche Angriffe sind in der Regel profitreicher als der Angriff auf ein komplex abgesichertes System. Daher überwiegt der Nutzen weiterer Sicherheitsmaßnahmen im Normalfall überproportional zu den für die
- 20 Sicherheitsmaßnahmen aufgewendeten Kosten.

Die erste Schutzschicht, die einem Angreifer entgegensteht, ist meist eine Firewall, die sowohl ein- als auch ausgehende Kommunikation an einem Computer überwacht und eine Zuordnung von Prozessen auf dem Computer zu bestimmten Protokollen, Ports, URLs sowie IP-Adressen erlaubt.

- 25 Dabei kann ein Blacklisting, also eine negative Bestimmung von Zugriffsrechten oder eine Whitelisting, eine positive Bestimmung von Zugriffsrechten erfolgen.

- 30 Antivirenprogramme überprüfen regelmäßig die permanenten Speicher eines Computers auf bekannte Schadprogramme. Dafür sammelt der Anbieter eines Antivirenprogramms Signaturen bekannter Schadsoftware und häufig auch bestimmte Verhaltensmuster. Nachteilig dabei ist, dass das Überprüfen des Speichermediums auf diese Weise zum einen relativ rechen- und somit zeitaufwändig ist. Zudem ist erforderlich, dass die Schadsoftware bereits bekannt ist.

Als weitere Schutzmaßnahme zur Sicherung von IT-Systemen dient das Berechtigungsmanagement, das die sinnvolle Verwaltung von Zugangsberechtigungen regelt. Nach dem „Principle of least privilege“ sollten Anwendungen und Prozesse nur mit den minimal nötigen Berechtigungen gestartet werden. Ferner sollten Strukturen im

- 5 Dateisystem nur insofern veränderbar sein, wie dies für den jeweiligen technischen Nutzer (bspw. Nutzer zum Ausführen eines Webservers) notwendig ist. Dieser könnte bspw. Lese-Berechtigung für die entsprechende Konfiguration haben, ohne die Konfiguration auch schreiben zu können.

In der Praxis ist diese Maßnahme sehr aufwändig und es kommt immer wieder zu

- 10 Fehlkonfigurationen, da die korrekte Konfiguration des Berechtigungsmanagements sehr komplex und unübersichtlich ist.

Das Sandboxing stellt ein Verfahren da, bei dem in der Regel durch das Betriebssystem sichergestellt wird, dass eine Anwendung nur einen ihr zugewiesenen Bereich des Dateisystems erreichen kann, sodass diese Anwendung keine Daten anderer Anwendungen

15 und insbesondere keine anderen Anwendungen manipulieren kann. Es handelt sich somit im Grunde um eine automatisierte Spezialanwendung des Berechtigungsmanagements.

Eine weitere Möglichkeit stellt die Isolation verschiedener Dienste in getrennte Computersysteme dar. Anstatt mehrere Services wie Webmail, Datenbank und Website auf

20 einem Computer innerhalb eines Betriebssystems bereitzustellen, werden diese Services bspw. auf unterschiedliche virtuelle Maschinen oder Container verteilt, wodurch beispielsweise ein kompromittierter Mail-Server nicht automatisch zu einer Kompromittierung der Datenbank führt.

- 25 Neben der Lösung mittels Antivirenprogrammen, bei der eine negative Berechtigungsliste (Programme, die nicht laufen dürfen) geführt wird, besteht auch die Möglichkeit nur bestimmte Anwendungen zuzulassen, deren Integrität wiederum häufig durch Prüfsummen sichergestellt wird. Diese Möglichkeit wird als Whitelisting bezeichnet.

Aus **US 2020/366696 A1** sowie **US 2001/044820 A1** sind Verfahren bekannt, bei denen die Integrität von Webservices im Besonderen überwacht wird, um der anspruchsvollen

30 Gefahrensituation im World Wide Web gerecht zu werden. In diesen Verfahren wird als Reaktion einer identifizierten Manipulation des Webservice allerdings nur eine vorher spezifizierte Warnung an ein weiteres System ausgegeben oder der Webservice in Teilen eingestellt bzw. der manipulierte Code entfernt. Somit wird nicht der vollständige Webservice wiederhergestellt. Insbesondere wenn unklar war, welcher Teil des Webservices in welcher

Art beeinträchtigt wurde, ist eine nachträgliche Analyse des Angriffsziels und der Angriffsart unter gleichzeitiger Wiederherstellung des Webservices nicht möglich. LU500837

Ein Computersystem, das in besonderem Maße durch Anwendung anderer, insbesondere mehrerer komplementärer Verfahren aus dem Stand der Technik geschützt und

- 5 insbesondere nur beschränkt aus einem unsicheren Netzwerk wie dem Internet zugänglich ist, ist ein geschütztes Computersystem im Sinne dieses Dokuments.

Neben der Absicherung eines Webservices steht zudem die Effizienz im Vordergrund.

- 10 Insbesondere lange Ladezeiten führen schnell zum „Abspringen“ des Nutzers, der aus Ungeduld die Internetseite eines alternativen Anbieters aufruft, aber auch die kostengünstige Skalierung spielt eine Rolle beim wirtschaftlichen Betrieb von Webservices.

- Oft werden die Daten, die ein Webservice zur Verfügung stellt, in Teilen oder vollständig zum Zeitpunkt der Anfrage neu berechnet. Als Caching bezeichnet man den Vorgang, bei dem die dabei erzeugten Daten, bspw. HTML-Dokumente, für eine Zeit gespeichert und bei einer
15 in diesem Zeitraum erfolgenden, identischen Anfrage wieder ausgeliefert werden, ohne dass die gesamte Berechnung erneut stattfinden muss. So wird Rechenleistung gespart und die Ladezeit der Daten entsprechend minimiert. Caching für Webservices ist bekannt aus
US 2005/193096 A1 sowie **US 2008/098301 A1**.

- Wie bereits beschrieben, besteht beim Betrieb eines Webservice mitunter der Bedarf, die
20 Last effizienter zu verarbeiten, um bspw. mehr Nutzer zufrieden stellen zu können. Ein wesentliches Merkmal für den Erfolg eines Webservice ist mitunter die Ladezeit, bevor der Server dem Nutzer eine Antwort bereitstellt (response time).

- Um diese Ladezeit möglichst gering zu halten, werden in der Regel weitere
Computersysteme als Server bereitgestellt, die parallel Anfragen bearbeiten und so die
25 Wartezeit bei einer Vielzahl an Anfragen im Schnitt deutlich senken können.

- Ferner können Betreiber eines internationalen Webservice auch entscheiden,
Computersysteme in anderen Rechenzentren zu betreiben, um die Distanz zu ihren Kunden zu reduzieren, sodass die Ladezeiten für den Kunden nicht durch Unwägbarkeiten wie
physikalische Begrenzungen der Übertragungsleitungen, Peeringabkommen zwischen
30 verschiedenen Internet-Service-Providern (ISPs) oder Ähnliches beeinträchtigt werden.

Ferner sind diese Verfahren in der Regel auch dazu geeignet, um die Ausfallsicherheit eines Webservice zu erhöhen.

Verfahren der hier beschriebenen Art nennt man „Load Balancing“ oder Lastverteilung.

Ungeduld die Internetseite eines alternativen Anbieters aufruft, aber auch die kostengünstige Skalierung spielt eine Rolle beim wirtschaftlichen Betrieb von Webservices. LU500837

Oft werden die Daten, die ein Webservice zur Verfügung stellt, in Teilen oder vollständig zum Zeitpunkt der Anfrage neu berechnet. Als Caching bezeichnet man den Vorgang, bei dem die dabei erzeugten Daten, bspw. HTML-Dokumente, für eine Zeit gespeichert und bei einer in diesem Zeitraum erfolgenden, identischen Anfrage wieder ausgeliefert werden, ohne dass die gesamte Berechnung erneut stattfinden muss. So wird Rechenleistung gespart und die Ladezeit der Daten entsprechend minimiert.

Wie bereits beschrieben, besteht beim Betrieb eines Webservice mitunter der Bedarf, die Last effizienter zu verarbeiten, um bspw. mehr Nutzer zufrieden stellen zu können. Ein wesentliches Merkmal für den Erfolg eines Webservice ist mitunter die Ladezeit, bevor der Server dem Nutzer eine Antwort bereitstellt (response time).

Um diese Ladezeit möglichst gering zu halten, werden in der Regel weitere Computersysteme als Server bereitgestellt, die parallel Anfragen bearbeiten und so die Wartezeit bei einer Vielzahl an Anfragen im Schnitt deutlich senken können.

Ferner können Betreiber eines internationalen Webservice auch entscheiden, Computersysteme in anderen Rechenzentren zu betreiben, um die Distanz zu ihren Kunden zu reduzieren, sodass die Ladezeiten für den Kunden nicht durch Unwägbarkeiten wie physikalische Begrenzungen der Übertragungsleitungen, Peeringabkommen zwischen verschiedenen Internet-Service-Providern (ISPs) oder Ähnliches beeinträchtigt werden.

Ferner sind diese Verfahren in der Regel auch dazu geeignet, um die Ausfallsicherheit eines Webservice zu erhöhen.

Verfahren der hier beschriebenen Art nennt man „Load Balancing“ oder Lastverteilung.

Neben den bereits beschriebenen Verfahren sollen der Vollständigkeit halber noch weitere Technologien genannt werden, die beim Betrieb eines Webservices eine Rolle spielen können:

Bittorrent ist ein Protokoll zum Austausch von Dateien im Internet. Dabei werden Dateien in direkt zwischen Computern über ein Netzwerk ausgetauscht, ohne dass eine zentrale Instanz zur Koordinierung nötig ist. In dem Protokoll werden ebenfalls Prüfsummen genutzt, um Teile von Dateien auf die korrekte Übertragung zu überprüfen. Auf Basis dieses Protokolls bieten einige Anbieter die Synchronisierung von Daten an. Dies ist bspw. auch geeignet, um die Daten zum Betrieb eines Webservices zwischen verschiedenen Computersystemen, die gemeinsam das Load Balancing umsetzen, zu synchronisieren.

- 5 a) Erzeugen (S01) eines intendierten Zustands von Daten, insbesondere Dateien und Daten in Datenbanken, auf einem ersten, sicheren Computersystem (primäres System),
- b) Übertragen (S02) der Daten durch eine gesicherte Verbindung auf einen oder mehrere Server (sekundäres Computersystem) in einem potentiell unsicheren Netzwerk zum Bereitstellen eines Web-Service,
- 10 c) Prüfen (S03) der Integrität der Daten in der Weise, dass ein Manipulieren der Daten durch einen Angreifer unmittelbar, also bevor der Angreifer, der die Datei manipuliert hat, auffällt und keinen weiteren Schaden anrichten kann,
- d) sofortiges Wiederherstellen (S04) des intendierten Status der manipulierten Daten vor dem unbeabsichtigten Manipulieren.

wobei zwischen Schritt S03 und S04 die Daten nach einer Manipulation (17) in Quarantäne verbracht werden, sodass diese im Nachhinein analysiert, aber nicht mehr ausgeführt werden und somit keinen Schaden verursachen können.

- 15 Weitere vorteilhafte Ausgestaltungen sind den Unteransprüchen und der Beschreibung zu entnehmen.

Allgemeine Vorteile

- 20 Das erfindungsgemäße Verfahren erhöht die Sicherheit beim Betreiben eines Webservice erheblich, während sie den technischen Aufwand zur Absicherung und somit die Kosten des Betriebs erheblich senkt. Durch die Software-Architekturprinzipien werden die allgemein bekannten technischen Risiken beim Betrieb eines Webservice ohne eine tiefe spezifische Kenntnis des eingesetzten Programms verringert oder eliminiert. Ferner erhöht das erfindungsgemäße Verfahren den technischen Aufwand und damit die Kosten für einen
- 25 potentiellen Angreifer enorm, da dieser

- a) wissen muss, dass der Webservice durch dieses Verfahren geschützt ist,
- b) den Agenten (11) nicht angreifen kann, da der Agent selbst durch das Verfahren geschützt ist,
- 30 c) er den Webservice (8) nicht nutzen kann, um Angriffe auf weitere Nutzer vorzunehmen, ohne dabei entdeckt zu werden,
- d) auch nicht das primäre System (1) angreifen kann, da dieses durch eine sichere Verbindung ausschließlich mit dem Agenten kommuniziert und sonst nicht mit dem unsicheren Netzwerk verbunden ist.

Aufgrund der deutlich erhöhten Kosten für den Angreifer ist nicht davon auszugehen, dass ein derartig geschütztes sekundäres System (7) weiter attackiert werden würde, da auch Angreifer wirtschaftlichen Zwängen unterliegen und so den Gewinn pro Aufwand maximieren müssen.

5

Ausführliche Beschreibung der Erfindung

Im Sinne der vorliegenden Erfindung werden zwischen Schritt S03 und S04 die Daten nach einer festgestellten unbeabsichtigten Manipulation in Quarantäne verbracht, sodass die Daten im Nachhinein analysiert, aber nicht mehr ausgeführt werden können, wodurch

10 vorteilhaft ein weiterer Schaden nach der Prüfung verhindert wird.

Das Verbringen der Daten in Quarantäne dient dem Schutz des Computersystems. Dafür können auffällige (manipulierte) Daten bspw. eine korrumpierte Datei in ein Verzeichnis vorzugsweise auf einem Server verbracht werden, aus dem ein in dieser Datei befindliches Programm nicht gestartet werden kann (Isolation).

15

In einer bevorzugten Ausgestaltung wird die Integrität der Dateien auf Basis kryptographischer Verfahren sichergestellt.

Als kryptographisches Verfahren eignen sich bspw. Signaturen durch Zertifikate. Dieses Verfahren ist standardisiert und wenig fehleranfällig, sodass es nicht zu sogenannten false positives („falsch Positiven“) kommen sollte.

20

Alternativ bevorzugt kann auf Basis von Hashwerten wie bspw. SHA256, SHA512, MD5 mit möglichst geringer Kollisionswahrscheinlichkeit sichergestellt werden, dass die Datei keinerlei Änderung erfährt, was ggf. leicht fehleranfalliger, aber dafür auch sicherer sein könnte.

25

In einer weiterhin bevorzugten Ausgestaltung der Erfindung findet das Überprüfen der Dateien unmittelbar nach dem Manipulieren der Daten statt.

Dies kann beispielsweise durch eine regelmäßige Prüfung der Integrität, insbesondere häufiger als einmal pro Stunde, weiterhin insbesondere häufiger als einmal pro 30 Minuten, weiterhin insbesondere häufiger als einmal pro 10 Minuten, weiterhin insbesondere häufiger als einmal pro 5 Minuten, weiterhin insbesondere häufiger als einmal pro Minute, weiterhin insbesondere häufiger als einmal pro Sekunde, weiterhin insbesondere häufiger als einmal pro 500 Millisekunden, weiterhin insbesondere häufiger als einmal pro 100 Millisekunden,

30

- d) sofortiges Wiederherstellen (S04) des intendierten Status der manipulierten Daten LU500837 vor dem unbeabsichtigten Manipulieren.

Weitere vorteilhafte Ausgestaltungen sind den Unteransprüchen und der Beschreibung zu entnehmen.

5

Allgemeine Vorteile

Das erfindungsgemäße Verfahren erhöht die Sicherheit beim Betreiben eines Webservice erheblich, während sie den technischen Aufwand zur Absicherung und somit die Kosten des Betriebs erheblich senkt. Durch die Software-Architekturprinzipien werden die allgemein
10 bekannten technischen Risiken beim Betrieb eines Webservice ohne eine tiefe spezifische Kenntnis des eingesetzten Programms verringert oder eliminiert. Ferner erhöht das erfindungsgemäße Verfahren den technischen Aufwand und damit die Kosten für einen potentiellen Angreifer enorm, da dieser

- a) wissen muss, dass der Webservice durch dieses Verfahren geschützt ist,
- 15 b) den Agenten (11) nicht angreifen kann, da der Agent selbst durch das Verfahren geschützt ist,
- c) er den Webservice (8) nicht nutzen kann, um Angriffe auf weitere Nutzer vorzunehmen, ohne dabei entdeckt zu werden,
- d) auch nicht das primäre System (1) angreifen kann, da dieses durch eine sichere
20 Verbindung ausschließlich mit dem Agenten kommuniziert und sonst nicht mit dem unsicheren Netzwerk verbunden ist.

Aufgrund der deutlich erhöhten Kosten für den Angreifer ist nicht davon auszugehen, dass ein derartig geschütztes sekundäres System (7) weiter attackiert werden würde, da auch Angreifer wirtschaftlichen Zwängen unterliegen und so den Gewinn pro Aufwand maximieren
25 müssen.

Ausführliche Beschreibung der Erfindung

Nach einer bevorzugten Ausgestaltung der vorliegenden Erfindung, werden zwischen Schritt S03 und S04 die Daten nach einer festgestellten unbeabsichtigten Manipulation in
30 Quarantäne verbracht, sodass die Daten im Nachhinein analysiert, aber nicht mehr ausgeführt werden können, wodurch vorteilhaft ein weiterer Schaden nach der Prüfung verhindert wird.

Das Verbringen der Daten in Quarantäne dient dem Schutz des Computersystems. Dafür können auffällige (manipulierte) Daten bspw. eine korruptierte Datei in ein Verzeichnis vorzugsweise auf einem Server verbracht werden, aus dem ein in dieser Datei befindliches Programm nicht gestartet werden kann (Isolation).

5

In einer bevorzugten Ausgestaltung wird die Integrität der Dateien auf Basis kryptographischer Verfahren sichergestellt.

Als kryptographisches Verfahren eignen sich bspw. Signaturen durch Zertifikate. Dieses Verfahren ist standardisiert und wenig fehleranfällig, sodass es nicht zu sogenannten false positives („falsch Positiven“) kommen sollte.

10

Alternativ bevorzugt kann auf Basis von Hashwerten wie bspw. SHA256, SHA512, MD5 mit möglichst geringer Kollisionswahrscheinlichkeit sichergestellt werden, dass die Datei keinerlei Änderung erfährt, was ggf. leicht fehleranfälliger, aber dafür auch sicherer sein könnte.

15

In einer weiterhin bevorzugten Ausgestaltung der Erfindung findet das Überprüfen der Dateien unmittelbar nach dem Manipulieren der Daten statt.

Dies kann beispielsweise durch eine regelmäßige Prüfung der Integrität, insbesondere häufiger als einmal pro Stunde, weiterhin insbesondere häufiger als einmal pro 30 Minuten, weiterhin insbesondere häufiger als einmal pro 10 Minuten, weiterhin insbesondere häufiger als einmal pro 5 Minuten, weiterhin insbesondere häufiger als einmal pro Minute, weiterhin insbesondere häufiger als einmal pro Sekunde, weiterhin insbesondere häufiger als einmal pro 500 Millisekunden, weiterhin insbesondere häufiger als einmal pro 100 Millisekunden, weiterhin insbesondere häufiger als einmal pro 10 Millisekunden, weiterhin insbesondere häufiger als einmal pro Millisekunde, weiterhin insbesondere häufiger als einmal pro 500 Mikrosekunden, weiterhin insbesondere häufiger als einmal pro 100 Mikrosekunden, weiterhin insbesondere häufiger als einmal 100 Mikrosekunden, weiterhin insbesondere häufiger als einmal pro 10 Mikrosekunden, weiterhin insbesondere häufiger als einmal pro Mikrosekunde, weiterhin insbesondere häufiger als einmal pro 500 Nanosekunden, weiterhin insbesondere häufiger als einmal pro 100 Nanosekunden, weiterhin insbesondere häufiger als einmal pro 10 Nanosekunden, weiterhin insbesondere häufiger als einmal pro Nanosekunde, weiterhin insbesondere häufiger als einmal pro 500 Picosekunden, weiterhin insbesondere häufiger als einmal pro 100 Picosekunden, weiterhin insbesondere häufiger als einmal pro 10 Picosekunden, weiterhin insbesondere häufiger als einmal pro Picosekunde, weiterhin insbesondere häufiger als einmal pro 500 Femtosekunden, weiterhin insbesondere

35

häufiger als einmal pro 100 Femtosekunden, weiterhin insbesondere häufiger als einmal pro 10 Femtosekunden, weiterhin insbesondere häufiger als einmal pro Femtosekunde, erreicht werden. LU500837

5 Bevorzugt sollte das Prüfungsintervall im Fall des regelmäßigen Überprüfens der Dateien in der Größenordnung der Ausführungszeit der notwendigen Taktzyklen der CPU je nach verwendeten Prüfalgorithmus liegen.

10 Durch die regelmäßige Prüfung binnen kürzester Zeitintervalle wird die Zeit, die ein Angreifer zum Erreichen seiner Ziele hat, vorteilhaft verkürzt, sodass dieser im Optimalfall keinerlei Schaden anrichten kann. Da viele Angriffe nach einer erfolgten (teil)automatischen Infektion des angegriffenen Systems einer händischen Analyse durch den Angreifer bedürfen, kann bereits durch Prüfungen mit einem Intervall in der Größenordnung von wenigen Minuten ein erheblich höheres Sicherheitsniveau erreicht werden. Durch eine weitere Verkürzung des Prüfintervalls können auch hochautomatisierte Angriffe erkannt und blockiert werden, bevor ein Schaden entstehen kann.

15 Alternativ kann die Unmittelbarkeit der Prüfung durch die Verwendung von Watch Dogs sichergestellt werden. Moderne Betriebssysteme erlauben die Registrierung eines sogenannten Watch Dog. Dabei wird dem Betriebssystem mitgeteilt, dass bei einer Manipulation einer Datei oder der Inhalte eines Verzeichnisses ein bestimmter Prozess gestartet werden soll. Dieser Prozess wird als Watch Dog bezeichnet.

20 In einer Ausgestaltung des erfindungsgemäßen Verfahrens kann auf Basis der Verwendung eines Watchdog seine unmittelbare Reaktion auf eine Manipulation der Daten erfolgen, ohne dass viele Ressourcen durch das ständige Prüfen blockiert würden. Somit handelt es sich um eine ressourcensparende Umsetzung zur Sicherung der Unmittelbarkeit des Prüfens der manipulierten Daten.

25

In einer weiteren bevorzugten Ausgestaltung werden in Schritt S01 des Verfahrens vermeintlich dynamische Dateien, wie bspw. durch PHP gerenderte HTML-Seiten, soweit es der betriebene Dienst erlaubt, als statische Dateien, wie bspw. HTML-, JSON- oder XML-Dateien, an das sekundäre System übertragen. Dies ist hilfreich, um mögliche
30 Sicherheitslücken im ausführbaren Programmcode, soweit möglich, ausschließlich auf dem primären, sicheren System auszuführen und vorteilhaft die Ladezeiten sowie die Ausfallsicherheit des Dienstes zu verbessern.

In einer möglichen Ausbildung des erfindungsgemäßen Verfahrens findet das Übertragen der Daten von einem primären an ein sekundäres System manuell statt. Dies bietet dem Betreiber der Website die volle Kontrolle darüber, wann welche Daten übertragen werden.

In einer bevorzugten Ausgestaltung erfolgt die Übertragung der Daten automatisch
5 zeitgesteuert, was dafür sorgt, dass der Webservice in seiner öffentlich zugänglichen Form regelmäßig auf den aktuellen intendierten Zustand der Daten gebracht wird. Die Zeitsteuerung wiederum kann einem zufälligen Muster entsprechen, sodass ein möglicher Angreifer nicht vorhersehen kann, wann es zu einer erneuten Übertragung kommt.

10 In einer besonders bevorzugten Ausgestaltung des Verfahrens findet die Übertragung der Daten automatisch nach Veränderung des intendierten Zustandes der Daten statt. Dies hat den Vorteil, dass die öffentlich zugängliche Version der Daten immer dem zuletzt aktualisierten Zustand entspricht.

Die Übertragung der Daten kann wahlweise a) den gesamten intendierten Zustand, bevorzugt aber b) nur die Differenz zu dem vorausgehenden intendierten Zustand der Daten
15 vom primären zum sekundären System umfassen. Nach a) ist die Möglichkeit für Fehler durch eine fehlerhafte Delta-Berechnung stärker eingeschränkt, dafür bietet b) eine Bandbreitenersparnis, was insbesondere bei umfangreichen Daten von großem Vorteil ist. Zudem bietet die Option des inkrementellen Updates den Vorteil, dass selbst eine kompromittierte Kommunikation des primären und des sekundären Systems keinen
20 Rückschluss auf sämtliche Daten zum Betrieb des Webservices zulässt.

In einer weiteren Ausbildung des Verfahrens wird die Änderung der intendierten Zustände und deren Übertragung revisionssicher gespeichert, wodurch eine spätere Auditierung ermöglicht wird.

25 Das Auditieren der Änderung stellt eine Praxis dar, bei der im Nachhinein nachvollzogen werden kann, wann welche Änderungen von wem vorgenommen wurden. Dafür müssen die Daten revisionssicher, als unveränderlich, gespeichert werden können.

Vorzugweise erfolgt dies mittels einer Blockchain, da in dieser dezentral Informationen revisionssicher gespeichert und übertragen werden, wodurch auch S02 zumindest teilweise
30 realisiert werden kann.

Unter Blockchain versteht man eine Technologie, die ähnlich einer Datenbank Informationen speichert. Dabei sind die Informationen nicht zentralisiert in einem Computersystem gespeichert, sondern befinden sich auf vielen Systemen gleichzeitig. Alle Instanzen der Datenbank enthalten dabei alle Informationen und die gesamte Historie ist zu jedem

Zeitpunkt abrufbar. Solange sich kein Konsens im Netzwerk aller Instanzen bildet, können historische Einträge nicht gelöscht werden.

LU500837

Durch das erfindungsgemäße Verfahren werden auch Daten in einer Datenbank geschützt.

5 Die Integrität der Daten in der Datenbank wird gesichert indem

- 10 a) entweder nach jedem erwünschten Manipulieren der Daten durch den Webservice die entsprechende Abfrage (Query) zunächst auf schadhafte Eingaben (bspw. SQL Injections, oder Änderungen an verbotenen Datensätzen, wie Datenbanknutzern) überprüft, dann gesichert an das primäre System übertragen und in der Folge die erfolgten Änderungen der Daten an die sekundären Systeme gespiegelt werden, oder
- 15 b) nach jedem erwünschten Manipulieren der Daten durch den Webservice die Query zunächst auf schadhafte Eingaben (bspw. SQL Injections, oder Änderungen an verbotenen Datensätzen, wie Datenbanknutzern) überprüft und dann gesichert an das primäre System übertragen wird und auch Lesezugriffe auf die Datenbank durch den Agenten geprüft und an das primäre System übertragen werden, sodass keine Datenbank im sekundären System notwendig ist, oder
- c) die Datenbank auf dem Sekundärsystem regelmäßig auf ein unintendiertes Ändern von verbotenen Datensätzen wie Datenbanknutzern oder anderen Daten, die nicht durch den Service bearbeitet werden müssen, überprüft wird.

20 Der Schutz der Daten in Datenbanken ist insbesondere deshalb von größter Relevanz, da hierdurch wohlbekannte Problematiken, wie bspw. sogenannte SQL injections verhindert werden können. SQL injections sind ein großes Sicherheitsproblem, bei dem ein Angreifer Zugang zu beliebigen Daten in einer Datenbank erhalten kann, wenn die auf die Datenbank zugreifende Software nicht entsprechend abgesichert wurde. Kein Verfahren aus dem Stand
25 der Technik schützt jede Dritthersteller-Software vor solchen Angriffen.

Dies ist insbesondere deshalb relevant, weil besonders kleine und mittelständische Unternehmen oft Software, bspw. zum Betrieb eines Webshops nutzen, die sie nicht selbst entwickeln, wodurch sie keinerlei Informationen über mögliche Sicherheitslücken haben.

30 In einer bevorzugten Ausgestaltung werden mehrere sekundäre Systeme parallel betrieben, um mittels Load Balancing die Last zu verteilen und so Ladezeiten des betriebenen Dienstes zu minimieren und weiterhin vorteilhaft die Ausfallsicherheit zu erhöhen.

In einer weiterhin bevorzugten Implementierung des Verfahrens mit mehreren sekundären Systemen wird nach einer Verletzung der Integrität eines Datums der gesamte sekundäre Server, auf dem die Integrität beschädigt wurde, isoliert, um eine spätere Analyse des gesamten Angriffs zu ermöglichen. In dem Fall gehört zur Isolation auch der Stop des gesamten Systems, um zu vermeiden, dass die ggf. laufende Schadsoftware ihre Spuren beseitigt. Dadurch wird die Attribution des Angriffs vorteilhaft erleichtert.

Weiterhin bevorzugt wird im Falle des Isolierens eines sekundären Systems ein Ersatzsystem gestartet, um die Funktion des Load Balancing zu erhalten.

In einer weiteren Ausführung des Verfahrens wird außerdem ein Reverse Proxy eingeführt, um sogenannte (D)DoS-Angriffe zu unterbinden. Dies kann in einer Ausführung mit einem oder mehreren sekundären Systemen geschehen.

Denial of Service (DoS) ist eine Angriffsstrategie auf Webservices, bei der Angreifer versucht, möglichst viele Anfragen gleichzeitig an den Server zu senden, die dieser nicht alle gleichzeitig beantworten kann.

Auf diese Weise wird der Service überlastet und Kunden des Service erhalten mitunter gar keine Antwort, einen Fehler, oder die Antwort braucht sehr lange. Dies kann beim Betreiber, insbesondere wenn der Angriff länger andauert, erheblichen Schaden verursachen, da in dieser Zeit in der Regel Umsätze ausfallen, da die potentiellen Kunden die Leistung des Webservice nicht abrufen können. Erfolgt der Angriff von vielen verteilten Computersystemen, spricht man von einem Distributed Denial of Service (DDoS).

Gerade in den vergangenen Jahren wurden Internet of Things (IoT) Geräte, wie bspw. smarte Glühbirnen oder Webcams für solche DDoS Angriffe genutzt. Dies ist insbesondere möglich, weil deren Software nur sehr selten mit Sicherheitsupdates versorgt wird. Durch die Verwendung des erfindungsgemäßen Verfahrens in solchen IoT-Devices wären diese ebenfalls geschützt und die Verwendung solcher Geräte für entsprechende Angriffe wäre deutlich erschwert.

Ferner wird im erfindungsgemäßen Verfahren ein Reverse Proxy umgesetzt, um der unter Anderem dem Schutz vor (D)DoS-Angriffen dient, um vorteilhaft zu geringeren Ladezeiten beizutragen und einen Ausfall der Dienstes mit einhergehenden Umsatzverlusten zu vermeiden.

Insbesondere können auch mehrere sekundäre Systeme zum Zwecke des Load Balancing und der Reduzierung von Ladezeiten durch geschickte räumliche Verteilung verwendet werden.

Das erfindungsgemäße Verfahren wird auf Grund seiner Anwendung auf verschiedene Computersysteme, eines primären und mindestens eines sekundären Systems, durch ein oder mehrere Computerprogrammprodukte realisiert.

- 5 Ein **erstes Computerprogrammprodukt**, das Schritt S01 implementiert, wobei bevorzugt auch vermeintlich dynamische Daten in statische Daten umgewandelt werden. Zudem steuert es die Übertragung des intendierten Zustands an den Server im unsicheren Netzwerk entweder manuell, zeitgesteuert oder automatisch nach der Änderung des intendierten Zustandes der Daten. Außerdem ist es dazu geeignet, den intendierten Zustand entweder vollständig oder als Differenz-Update zu übertragen.
- 10 Ein **computerlesbares Medium** (Datenträger), umfassend das erste Computerprogrammprodukt zur Definition des intendierten Zustandes in S01.
- Das Computersystem umfassend das erste Computerprogrammprodukt und das computerlesbare Medium mit dem ersten Computerprogrammprodukt ist das primäre System.
- 15 Das **sekundäre Computerprogrammprodukt** implementiert die Schritte S03 und S04. Bevorzugt wird das Verbringen der manipulierten Daten in Quarantäne implementiert, wobei das Prüfen der Daten mittels kryptographischer Verfahren unmittelbar nach deren Manipulation erfolgt und die Datenbank durch eine der oben benannten Möglichkeiten sichert. Sollte das sekundäre Computerprogrammprodukt beendet werden, wird es
- 20 automatisch durch das sekundäre Computersystem neu gestartet.
- Das sekundäre Computerprogrammprodukt ist auf einem sekundären computerlesbaren Medium (Datenträger) enthalten.
- Das sekundäre Computerprogrammprodukt schützt bevorzugt ein tertiäres computerlesbares Medium (Datenträger).
- 25 Ein sekundäres Computersystem umfasst bevorzugt ein sekundäres und ein tertiäres Medium. Ein Medium kann die Eigenschaften des sekundären und des tertiären Mediums miteinander vereinen. Bevorzugt sind auch die Computerprogrammprodukte dieser Anmeldung selbst durch ein sekundäres Computerprogrammprodukt geschützt, sodass sie selbst vorteilhaft vor Manipulation geschützt sind.
- 30 Ein tertiäres Computerprogrammprodukt implementiert die Übertragung des intendierten Zustandes vom primären an das sekundäre System. Es ist Bestandteil eines computerlesbaren Mediums und wird entweder auf dem primären oder einem sekundären oder einem weiteren tertiären System ausgeführt.

Die Computerprogrammprodukte können zudem durch Plugins um weitere Funktionalitäten zum Betrieb und zur Verbesserung eines Webservices erweitert werden. Zum Beispiel könnte ein Plugin zur Aufbereitung und Analyse des generierten Datenverkehrs hinzugefügt werden, wodurch diese Dienstleistung vorteilhaft nicht durch einen Drittanbieter

5 vorgenommen werden muss und andere Aspekte der Datensparsamkeit und des Datenschutzes gewahrt werden können, indem die Informationen beim Betreiber des Webdienstes verbleiben.

Um sicherzustellen, dass das Computerprogrammprodukt selbst nicht manipuliert wird, wodurch der Schutz erlöschen würde, bieten sich mehrere Methoden an. Diese erschweren

10 positiv die Veränderung der Funktionalität des Computerprogrammprodukts, sodass der Schutz weiter erhöht wird.

Einerseits kann das Computerprogrammprodukt zur Implementierung des Verfahrens die Ordnerstruktur bzw. das Medium, in der es selbst abgelegt ist, überwachen. Während es im Arbeitsspeicher des Computers liegt, ist es vor Manipulation stark geschützt. Wird nun das

15 Programm selbst auf dem Medium modifiziert, kann es die Änderung erkennen und sich selbst wiederherstellen.

Ferner kann das Programm auf einem Medium, das nicht neu geschrieben werden kann, bspw. ein EPROM oder eine SD-Karte mit Schreibschutz gespeichert werden.

Des Weiteren schützen bereits heute viele Betriebssysteme Teile Ihrer Verzeichnisstruktur in der Form, dass diese nur im abgesicherten Modus beschrieben werden können und

20 ansonsten nicht einmal durch das System selbst modifiziert werden können. Das Computerprogrammprodukt könnte in einem solchen Teil der Verzeichnisstruktur während der Nutzung des abgesicherten Modus hinterlegt werden, um es weiterhin vorteilhaft zu schützen.

Auch die Manipulation des Computerprogrammprodukts im Arbeitsspeicher ist möglich. Durch einen Fachmann könnte die Struktur des Programms im Arbeitsspeicher erkannt werden, wodurch dieser das Programm im Speicher suchen und durch Ausnutzen anderer

25 Schwachstellen, wie zum Beispiel buffer overflows, manipulieren könnte. Durch gezielte Maßnahmen kann das Computerprogrammprodukt solche Angriffe positiv erschweren. Dafür

30 füllt es zufällige Speicherbereiche mit zufälligen Inhalten, um eine Analyse und Manipulation der für die im Programm enthaltenen Anweisungen zu erschweren, indem deren Position bei jeder Ausführung und über die Zeit wechselt.

Weitere vorteilhafte Ausgestaltungen und Weiterbildungen ergeben sich aus den Unteransprüchen sowie aus der Beschreibung unter Bezugnahme auf die Figuren.

Ausführungsbeispiele

LU500837

Anhand folgender Figuren und Ausführungsbeispiele wird die vorliegende Erfindung näher erläutert, ohne die Erfindung auf diese zu beschränken.

Dabei zeigt

5 **Fig. 1:** eine schematische Darstellung des primären Computersystems CS1 (1). Abgebildet ist ein auf CS1 laufender Webservice (2) mit dynamischen Inhalten („index.php“ und „shop.php“), sowie eine statische Bilddatei (image.png). Beide dynamische Dateien verwenden die Datenbank (4), allerdings liest (5) die Datei „index.php“ Daten aus der Datenbank (4), die sich in diesem Beispiel nicht ändern. Daher kann die Datei
10 „index.php“ durch ein Computerprogrammprodukt nach den Ansprüchen 12 und 5 in eine statische Datei (index.html) umgewandelt werden (S01). Die Datei „shop.php“ liest und schreibt (6) Daten aus der bzw. in die Datenbank (4). Daher kann diese nicht in eine statische Datei umgewandelt werden. Auch an der bereits statischen Datei ändert sich nichts. Somit ergibt sich der intendierte Zustand der Dateien des
15 Webservice (3). Ferner werden in Schritt S01 die Hashes, angedeutet durch alphanumerische Zeichenketten neben den Dateinamen, berechnet und gespeichert. Der Schild soll die entsprechende Sicherung der Dateien verdeutlichen.

Fig. 2: Eine schematische Darstellung des CS1 (1) aus Fig. 1 samt eines zweiten, sekundären Systems CS2 (7) in einem unsicheren Netzwerk. Durch S02 wird der
20 intendierte Zustand (3) der Datenbank (4) sowie der Dateien des Webservice (2) in das sekundäre System (7) überführt. Ein Agent (11) wird bei Veränderungen der Daten auf Dateisystemebene (gemeldet durch das Betriebssystem (10)) oder zeitlich regelmäßig tätig, um die Integrität der Daten des Webservice (2) zu überprüfen. Änderungen an der Datenbank (4) (in dem Beispiel durch die Datei
25 „shop.php“) werden durch den Agenten (11) geprüft (13), bevor dieser diese Änderungen an die Datenbank (4) im primären System (1) überträgt (14) und diese erneut durch S02 in die sekundäre Datenbank (9) des sekundären Systems (7) eingepflegt werden. Die dynamischen Dateien (im Beispiel die Datei „shop.php“) lesen (12) die notwendigen Daten aus der Datenbank im sekundären System (9).
30 Der Schritt S03, das Prüfen der Dateien des sekundären Webservice (8) durch den Agenten (11) nach der Manipulation, wird entweder zeitgesteuert oder durch einen Watchdog ausgelöst.

Fig. 3: Eine schematische Darstellung des primären Computersystems (1) und des sekundären Computersystems (7). In diesem Fall gibt es keine Datenbank im
35 sekundären System (7). Durch S02 wird der intendierte Zustand (3) der Dateien des

Webservice (2) in das sekundäre System CS2 (7) übertragen. Erneut löst das Betriebssystem (10) entweder bei Veränderung der Daten (über einen Watchdog) oder zeitlich gesteuert Schritt S03 aus, woraufhin der Agent (11) die Dateien des Webservice (8) im sekundären System (7) mit den intendierten Daten (3) des primären Systems (1) abgleicht. Da die sekundäre Datenbank entfällt, findet der Lesezugriff (15) von Dateien wie der „shop.php“ auf die Datenbank (4) des primären Systems (1) durch den Agenten (11) statt (16). Der Schreibzugriff (13) wird analog zu Fig. 2 durch den Agenten (11) geprüft und danach an die primäre Datenbank (4) übertragen (14).

10 **Fig. 4:** Schematische Darstellung des Agierens des Agenten (11) im Falle einer unerwünschten Manipulation (17 der Dateien (S04)).

15 A) Das Verfahren mit Hilfe eines Watch Dog: In diesem Fall reagiert (23) das Betriebssystem (10) auf die Änderung der Dateien des Webservices (8) und informiert (22) den Agenten (11). Dieser prüft (20) alle Dateien des Webservice im ggf. manipulierten Zustand (18). Nachdem er mittels eines kryptographischen Verfahrens, in diesem Fall mittels eines Hashwertes, eine Änderung erkannt hat, stellt er die erwünschten Datei-Zustände (19) wieder her (21) und verbringt die manipulierten Dateien ggf. zwecks späterer Analyse in Quarantäne.

20 B) In diesem Fall meldet (22) das Betriebssystem (10) regelmäßig die Systemzeit oder das Überschreiten eines bestimmten Zeitintervalls an den Agenten (11), der den gegenwärtigen Zustand der Dateien des Webservices (18) prüft (20) und nach einer unerwünschten Manipulation (17) zu dem Schluss kommt, dass diese manipuliert wurden, woraufhin er den erwünschten Zustand der Dateien wiederherstellt (21) und ggf. die manipulierten Dateien (18) zwecks späterer Analyse in Quarantäne verbringt.

30 Allen Ausführungsformen ist der Umstand gemein, dass das primäre Computersystem CS1 (1) nur durch eine sichere Verbindung durch den Agenten (11) erreicht werden kann. Der Webservice (2) des primären Systems (1) ist ausschließlich notwendig, um den intendierten Zustand (3) herzustellen und vermeintlich dynamische Dateien in statische Dateien umwandeln zu können (bspw. durch Rendering von HTML durch eine PHP-Datei). Das Feststellen des intendierten Zustands in Schritt S01 kann durch verschiedene kryptographische Methoden erfolgen. Figur 2 und 3 unterscheiden sich lediglich in der Verwendung der Datenbank, insbesondere darin, wie die Integrität der Daten in der Datenbank sichergestellt wird..

5

Prinzipiell kann es eine beliebige Zahl der sekundären Systeme (7) geben, auch wenn dies nicht grafisch dargestellt wird. Der Agent (11) dient in der Konfiguration mit einem sekundären System (7) ferner auch als Reverse Proxy zum Schutz vor DDoS-Angriffen, sodass auch bei den sekundären Systemen (7) kein direkter Zugriff auf den Webservices (8) des sekundären Systems (7) erfolgt.

Bezugszeichenliste

LU500837

- | | |
|----|---|
| 1 | Computersystem 1 (primäres System) |
| 2 | Webservice auf dem primären System |
| 3 | Intendierter Status der Webservice-Dateien nach dem Reduzieren auf statische |
| 5 | Dateien (S01) durch das primäre Computerprogrammprodukt |
| 4 | Datenbank des Webservice auf dem primären System |
| 5 | Lesezugriff der Datei „index.php“ auf die Datenbank |
| 6 | Lese-/Schreibzugriff der shop.php auf die Datenbank |
| 7 | Computersystem 2 (sekundäres System) |
| 10 | 8 Webservice mit dem intendierten Zustand der Dateien nach Schritt S02 |
| 9 | Datenbank auf dem sekundären System, mit durch S02 synchronisierten
erwünschtem Zustand |
| 10 | Betriebssystem-Mechanismus für den Start der Prüfung (S03) (Watchdog oder Timer) |
| 11 | Agent laufend auf Computersystem 2 (Sekundäres Computerprogrammprodukt) |
| 15 | 12 Lesezugriff der Datei „shop.php“ von der Datenbank des sekundären Systems |
| 13 | Schreibzugriff der Datei „shop.php“ gesendet an den Agenten (14) |
| 14 | Schreibzugriff der Datei „shop.php“ nach der Überprüfung an den Agenten zum
Update des erwünschten Zustands der Datenbank des primären Systems |
| 15 | Lesezugriff der Datei „shop.php“, bei der durch den Agenten direkt aus der |
| 20 | Datenbank des primären Systems gelesen wird (16), ohne diese zuvor in eine
sekundäre Datenbank zu spiegeln |
| 16 | Lesezugriff des Agenten ausgelöst durch den Lesezugriff der Datei „shop.php“ (15)
geprüft durch den Agenten (11) |
| 17 | Manipulation der Dateien des Webservice |
| 25 | 18 Manipulierter Webservice |
| 19 | Webservice im wiederhergestellten intendierten Zustand |
| 20 | Feststellen des manipulierten Zustands der Dateien durch den Agenten |

- | | | |
|----|---|----------|
| 21 | Wiederherstellen des intendierten Zustands durch den Agenten mit Löschung der manipulierten Dateien oder vorzugsweise Verbringen der Dateien im manipulierten Zustand in „Quarantäne“ | LU500837 |
| 22 | Signal des Systems an den Agenten, dass eine Datei geändert wurde (Watchdog) | |
| 5 | oder ein Zeitintervall verstrichen ist (Timer) | |
| 23 | Registrierung der Dateiänderung beim System | |

1. **Verfahren zum Schutz der Integrität von Daten**, umfassend die folgenden Schritte:

- 5 a) Erzeugen (S01) eines intendierten Zustands (3) von Daten eines Webservice (2) auf einem ersten, sicheren Computersystem (primäres System) (1),
- b) Übertragen (S02) der Daten durch eine gesicherte Verbindung auf einen oder mehrere Server (sekundäres Computersystem) (7) in einem potentiell unsicheren Netzwerk zum Bereitstellen eines Web-Service,
- 10 c) Prüfen (S03) der Integrität der Daten in der Weise, dass ein Manipulieren der Dateien und Daten durch einen Angreifer unmittelbar auffällt,
- d) sofortiges Wiederherstellen (21) (S04) des intendierten Zustands der Daten (19) vor dem unbeabsichtigten Manipulieren.

wobei zwischen Schritt S03 und S04 die Daten nach einer Manipulation (17) in Quarantäne verbracht werden, sodass diese im Nachhinein analysiert, aber nicht mehr ausgeführt werden und somit keinen Schaden verursachen können.

15

2. Verfahren nach Anspruch 1, wobei die Integrität der Daten auf Basis kryptographischer Verfahren sichergestellt wird.

20 3. Verfahren nach einem der Ansprüche 1 oder 2, wobei das Überprüfen der Dateien (20) unmittelbar nach dem Manipulieren (17) der Dateien erfolgt.

4. Verfahren nach einem der Ansprüche 1 bis 3, wobei in Schritt S01 vermeintlich dynamische Dateien, wie bspw. durch PHP gerenderte HTML-Seiten, soweit es der betriebene Dienst erlaubt, als statische Dateien, wie bspw. HTML-, JSON- oder XML-Dateien, an das sekundäre System übertragen werden.

25

5. Verfahren nach einem der Ansprüche 1 bis 4, wobei das Auslösen des Übertragens der Daten von einem primären System (1) an ein sekundäres System (7)

- 30 a) manuell ausgelöst wird, oder
- b) automatisch zeitgesteuert ausgelöst wird, oder
- c) automatisch nach Veränderung des intendierten Zustandes der Daten ausgelöst wird.

6. Verfahren nach einem der Ansprüche 1 bis 5, bei dem nur die Differenz des intendierten Zustands zu dem vorausgehenden intendierten Zustand der Daten vom primären System (1) an das sekundäre System (7) übertragen wird.
- 5
7. Verfahren nach einem der Ansprüche 1 bis 6, wobei die Änderung der intendierten Zustände (3) und deren Übertragung an ein sekundäres System im Schritt (S02) revisionssicher gespeichert wird, wodurch eine spätere Auditierung ermöglicht wird.
- 10
8. Verfahren nach einem der Ansprüche 1 bis 7, wobei die Daten aus Daten einer Datenbank ausgewählt sind, und wobei die Integrität der Daten in der Datenbank (4) gesichert wird, indem
- 15
- a) entweder jede Anfrage an die Datenbank (Query) zunächst auf schadhafte Eingaben überprüft, dann gesichert an die primäre Datenbank (4) des primären Systems (1) übertragen und in der Folge die erfolgten Änderungen an die sekundäre Datenbank (9) im sekundären System (7) gespiegelt werden, oder
- 20
- b) nach jedem erwünschten Manipulieren der Daten (13) durch den Webservice die Query zunächst auf schadhafte Eingaben überprüft und dann gesichert an das primäre System (1) übertragen werden (14) und auch Lesezugriffe auf die Datenbank (15) durch den Agenten (11) geprüft und an das primäre System (16) übertragen werden, sodass keine Datenbank im sekundären System (7) notwendig ist, oder
- 25
- c) die Datenbank auf dem sekundären System (9) regelmäßig auf ein unintendiertes Ändern von verbotenen Datensätzen wie Datenbanknutzern oder anderen Daten, die nicht durch den Service (8) bearbeitet werden müssen, überprüft wird.
- 30
9. Verfahren nach einem der Ansprüche 1 bis 8, wobei mehrere sekundäre Systeme (7) parallel betrieben werden, um ein Load Balancing zu ermöglichen.
- 35
10. Verfahren nach Anspruch 9, wobei nach Verletzung der Integrität eines Datums das gesamte sekundäre System (7), auf dem die Integrität beschädigt wurde (kompromittiertes System), isoliert wird.

11. Verfahren nach Anspruch 10, wobei nach dem Isolieren des kompromittierten Systems LU500837 ein weiteres sekundäres System als Ersatzsystem gestartet wird.
- 5 12. Verfahren nach einem der Ansprüche 1 bis 8, wobei auf dem sekundären System (7) ein Reverse Proxy zum Einsatz kommt, um den Webservice (8) vorteilhaft vor (D)DoS-Angriffen zu schützen.
- 10 13. Verfahren nach einem der Ansprüche 9 bis 12, wobei neben dem Load Balancing noch ein Schutz vor (D)DoS implementiert ist.
14. **Computerprogrammprodukt**, das Schritt S01 aus Anspruch 1 ausführbar verwirklicht, wobei vermeintlich dynamische Daten in statische Daten umgewandelt werden.
- 15 15. **Computerlesbares Medium**, umfassend ein Computerprogrammprodukt nach Anspruch 14.
- 20 16. **Ein primäres Computersystem (1)**, umfassend ein computerlesbares Medium nach Anspruch 15, das zur Definition des definierten Status der Daten nach Anspruch 1 geeignet ist.
- 25 17. **Computerprogrammprodukt**, das die Schritte S03 und S04 des Anspruch 1 implementiert, wobei bevorzugt eine beliebige Kombination der Merkmale aus den Ansprüchen 2, 3 und 8 implementiert wird, wobei besonders bevorzugt alle Merkmale aus den genannten Ansprüchen implementiert werden.
- 30 18. **Computerlesbares Medium**, umfassend ein Computerprogrammprodukt nach Anspruch 17.
19. Computerlesbares Medium, mindestens teilweise geschützt durch ein Computerprogrammprodukt nach Anspruch 17.
- 35 20. **Computersystem**, umfassend ein computerlesbares Medium nach Anspruch 18 und ein computerlesbares Medium nach Anspruch 19, wobei auch ein computerlesbares Medium Anspruch 18 und Anspruch 19 erfüllen kann.
21. **Computerprogrammprodukt**, das Schritt S02 des Anspruch 1 implementiert, wobei bevorzugt eine beliebige Kombination der Merkmale aus den Ansprüchen 5, 8, 9, 10,

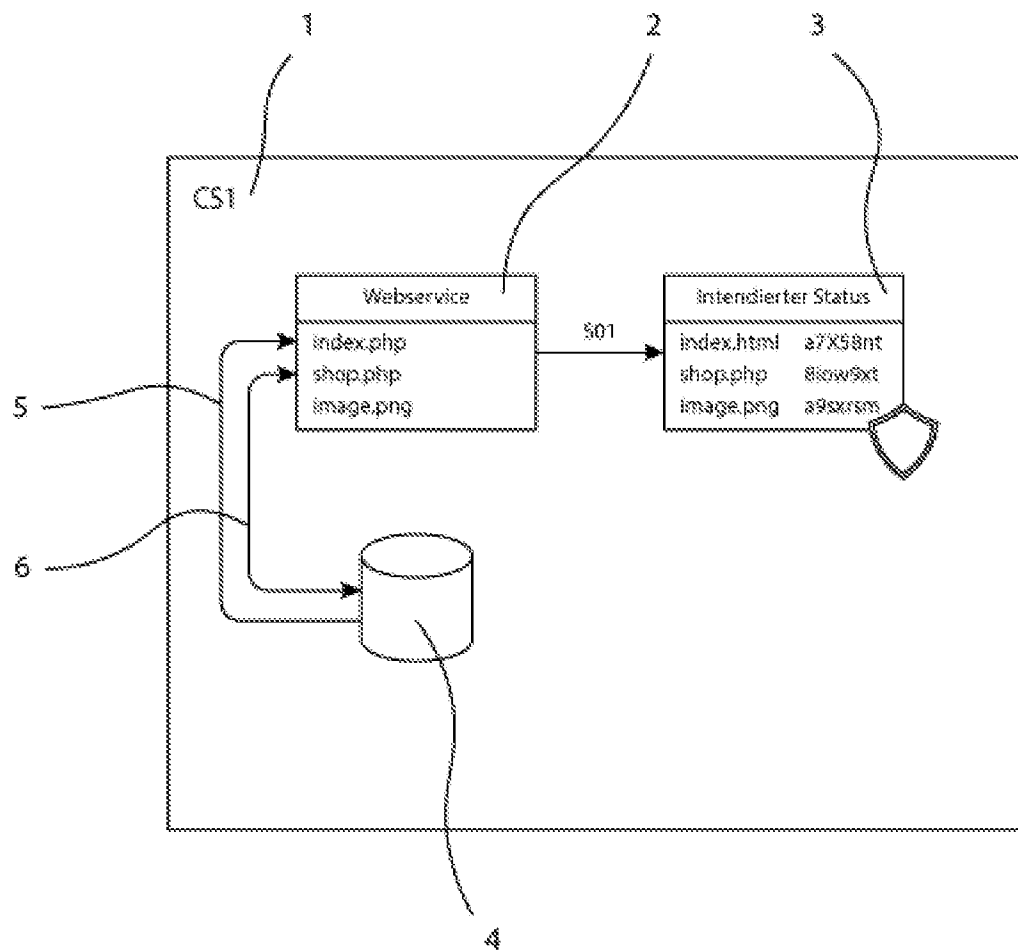
11, 12 und 13 implementiert wird, wobei besonders bevorzugt alle Merkmale aus den genannten Ansprüchen implementiert werden. LU500837

22. **Computerlesbares Medium**, umfassend ein Computerprogrammprodukt nach
5 Anspruch 21.

23. **Computersystem**, umfassend ein computerlesbares Medium nach Anspruch 22

24. **Netzwerk**, umfassend Computersysteme nach den Ansprüchen 16, 20 und 23.

10

**Fig. 1**

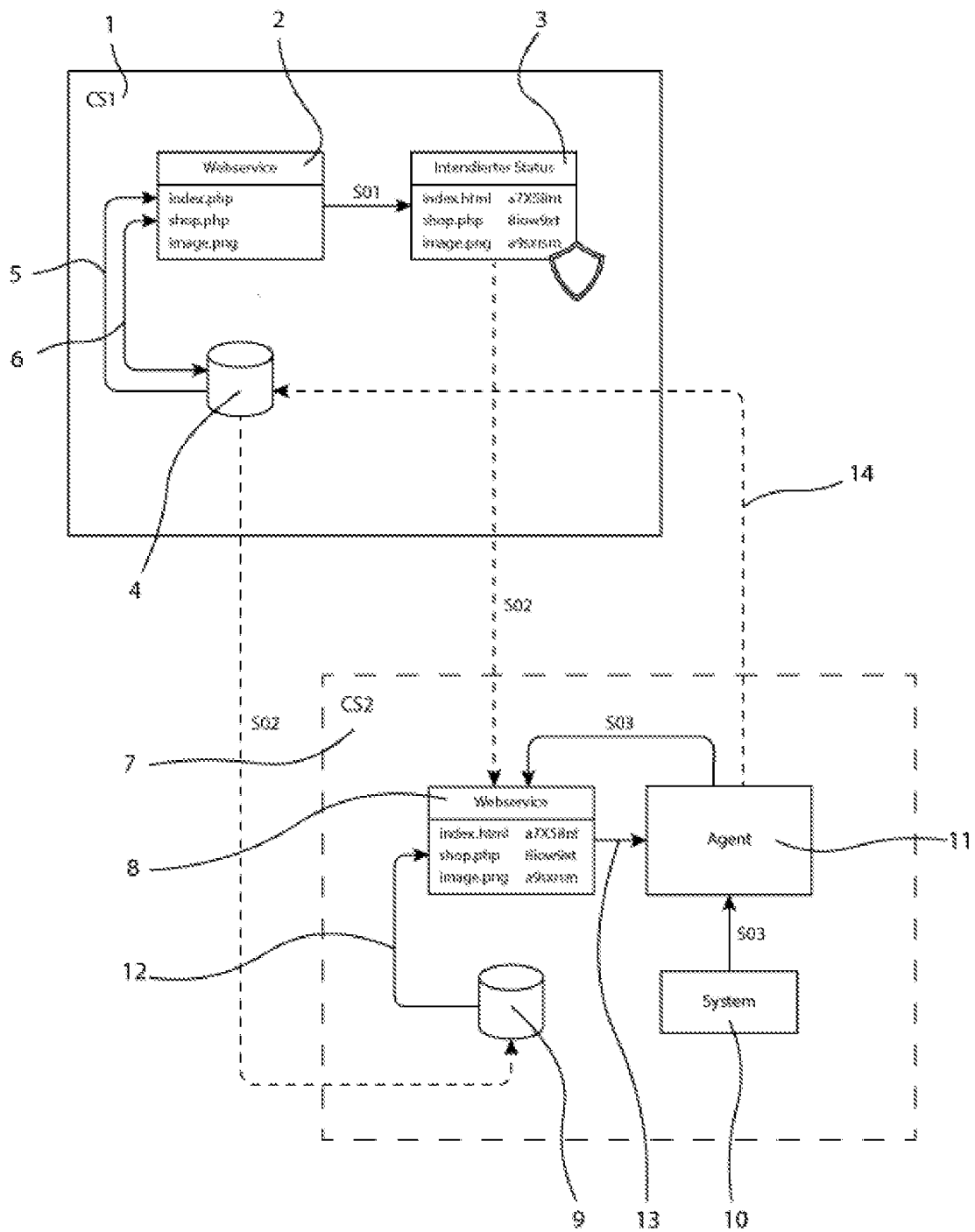


Fig. 2

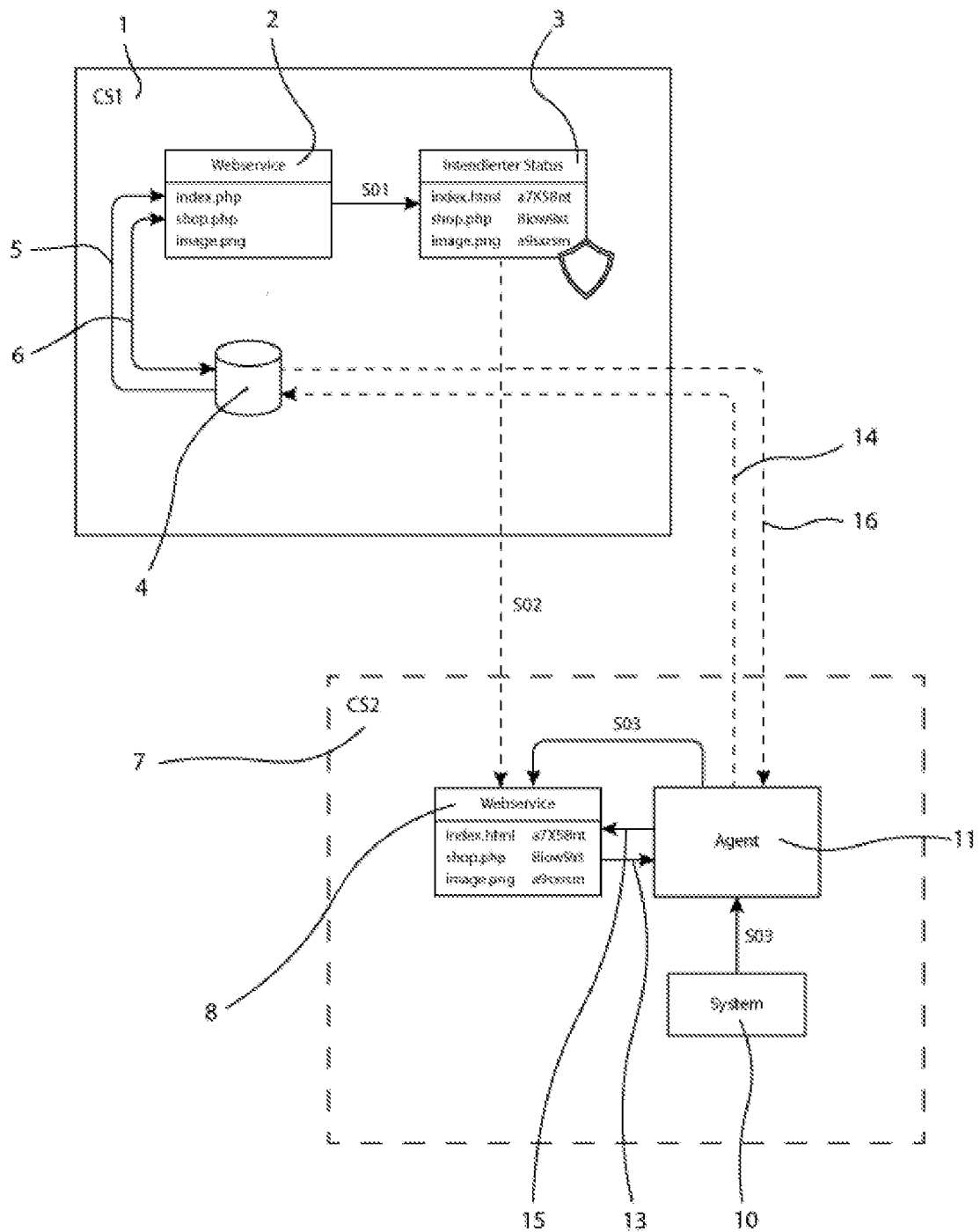


Fig. 3

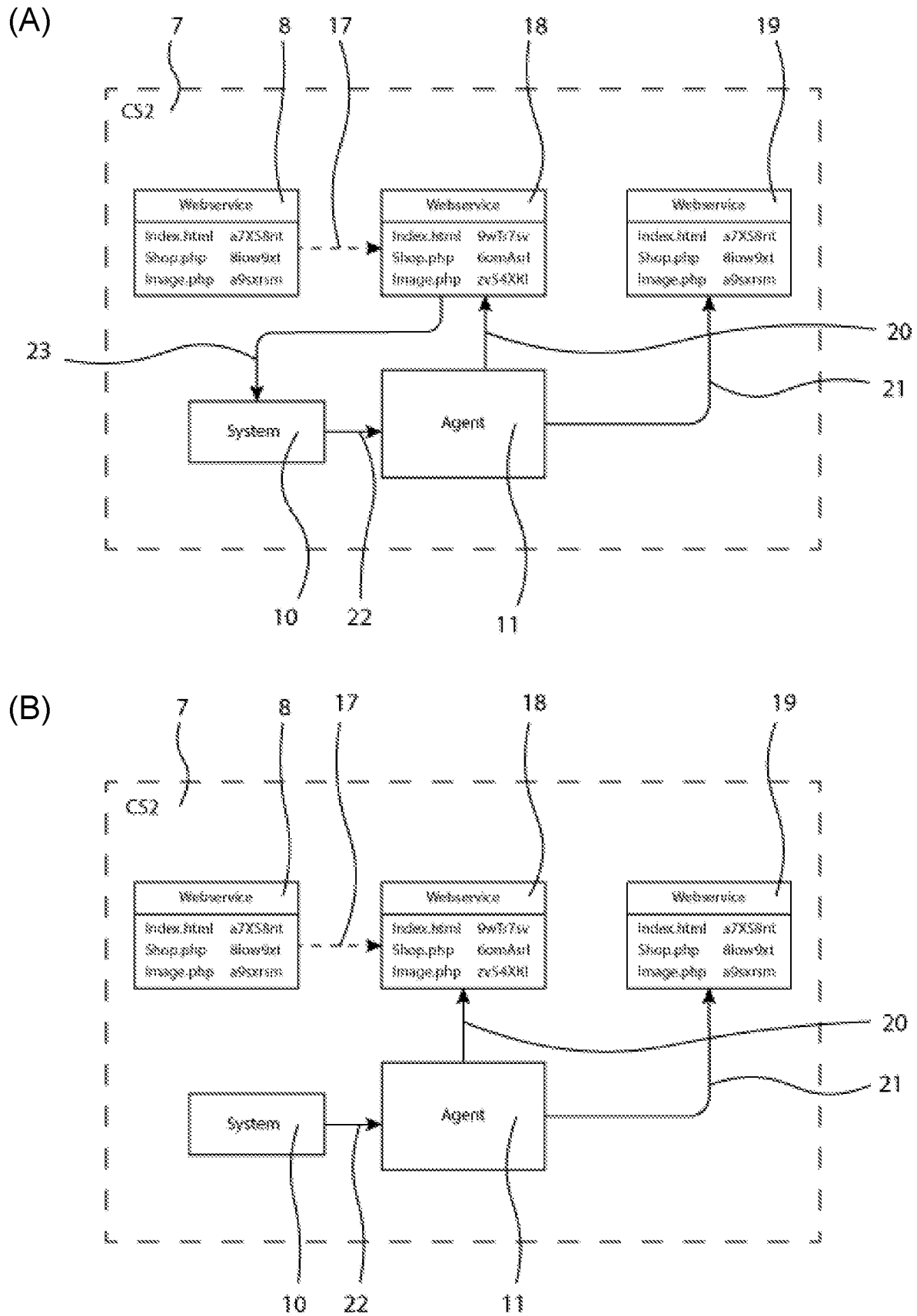


Fig. 4