



Dokumentation von Team Hahner® – Engineers of (Word) Solutions
Version zur Veröffentlichung auf dem Hahner-Blog und YouTube-Kanal von
Markus Hahner

Office-Sicherheit: Makro-, VBA- und andere Sicherheitseinstel- lungen verstehen und konfigu- rieren

Version 1.17

Autor Dipl.-Ing. (FH) Markus Hahner

Team Hahner® – Engineers of (Word) Solutions

www.hahner.de | www.schauen-statt-lesen.de |
www.office-sicherheit.de



Inhaltsverzeichnis

1.	Warum Office-Sicherheitseinstellungen unterschätzt werden – mit fatalen Konsequenzen	6
2.	Was ist ein Makro, was ist VBA?	9
2.1	VBA als Makro-Programmiersprache.....	10
2.2	VBA-Editor als Programmierumgebung	11
2.3	Speicherort des VBA-Codes	13
2.4	Aufbau des VBA-Codes	15
2.5	Leistungsumfang von VBA	16
2.5.1	Befehlszeilenoptionen zur Vermeidung von Automatisierungsfunktionen	16
2.5.2	Auszug aus den Word-Events zur Automatisierung	17
2.5.3	Auszug aus den Excel-Events zur Automatisierung.....	19
2.5.4	Via VBA auf VBA zugreifen	20
3.	Makroschutz: Ungewollte Ausführung verhindern	23
3.1	Makro-Viren – Schadprogramme im Office-Datei-Format	23
3.2	Makro-Schutzeinstellungen – was wie einstellen?.....	24
3.3	Achtung Viren-Einfallstor: Entwicklereinstellung	26
3.4	Makroschutz: Anpassungen via Gruppenrichtlinien.....	28
3.4.1	Microsoft Excel.....	28
3.4.2	Microsoft PowerPoint	30
3.4.3	Microsoft Word.....	31
3.4.4	Microsoft Access	33
4.	Digital signierten VBA-Code nutzen	35
4.1	Wie funktioniert ein digitales Zertifikat?	35
4.2	Digitale Signatur am Beispiel von QuoVadis.....	36
4.3	Zertifikatsdateien für die Verteilung via Gruppenrichtlinien vorbereiten	40
4.4	Zertifikate via Gruppenrichtlinien verteilen.....	45
4.5	VBA-Code digital signieren.....	47
4.6	Digitale Signatur in Add-Ins verifizieren.....	48
5.	Vertrauenswürdige Speicherorte: Auswirkungen auf die Sicherheitseinstellungen	50
5.1	So wirken sich vertrauenswürdige Speicherorte aus.....	50
5.2	Standardeinstellungen der vertrauenswürdigen Speicherorte.....	53
5.2.1	Vertrauenswürdige Speicherorte in Excel	53
5.2.2	Vertrauenswürdige Speicherorte in PowerPoint.....	61
5.2.3	Vertrauenswürdige Speicherorte in Word	66
5.2.4	Vertrauenswürdige Speicherorte in Access	70

5.3	Vertrauenswürdige Speicherorte: Benutzeranpassungen	71
5.4	Vertrauenswürdige Speicherorte: Anpassungen via Gruppenrichtlinien	74
5.4.1	Microsoft Office	74
5.4.2	Microsoft Excel	78
5.4.3	Microsoft PowerPoint	83
5.4.4	Microsoft Word	88
5.4.5	Microsoft Access	93
5.5	Praxiserfahrungen bei vertrauenswürdigen Speicherorten	97
5.5.1	Ausführung eigener Makros in Word & Excel wird „versehentlich“ gesperrt	98
5.5.2	Word-Startup-Add-Ins trotz Sperrung des Startup-Ordners ausführen	100
5.5.3	Vagabundierende Word-Registerkarten bei fehlender Vertrauenswürdigkeit	101
5.5.4	OneDrive-Ordner vertrauenswürdig machen	102
5.5.5	Zugriff auf Access-Assistenten erlauben	105
6.	Vertrauenswürdige Dokumente: Auswirkungen auf die Sicherheitseinstellungen	108
6.1	So wirken sich vertrauenswürdige Dokumente auf die anderen Sicherheitseinstellungen aus	109
6.2	Wie sich die Office-Programme vertrauenswürdige Dokumente merken	111
6.3	Einstellungsmöglichkeiten der vertrauenswürdigen Dokumente	112
6.4	Vertrauenswürdige Dokumente: Anpassungen via Gruppenrichtlinien	114
6.4.1	Microsoft Excel	114
6.4.2	Microsoft PowerPoint	117
6.4.3	Microsoft Word	119
6.4.4	Microsoft Access	122
7.	Übersicht: Wann wird VBA-Code ausgeführt?	125
7.1	Ergebnismatrix bei nicht zertifiziertem VBA-Code	126
7.2	Ergebnismatrix bei zertifiziertem VBA-Code	127
8.	Dateiformate erlauben bzw. sperren (Zugriffsschutzeinstellungen)	128
8.1	Microsoft Excel	128
8.1.1	Standardverhalten für den Zugriffsschutz festlegen	128
8.1.2	Einstellungen für den Zugriffsschutz	130
8.1.3	Standarddateiformat	135
8.2	Microsoft PowerPoint	136
8.2.1	Standardverhalten für den Zugriffsschutz festlegen	136
8.2.2	Einstellungen für den Zugriffsschutz	138
8.2.3	Standarddateiformat	141
8.3	Microsoft Word	142
8.3.1	Standardverhalten für den Zugriffsschutz festlegen	142

8.3.2	Einstellungen für den Zugriffsschutz	144
8.3.3	Standarddateiformat	148
9.	Makro-Virus: Funktionsweise entschlüsselt	150
9.1	Word: Makro-Virus „info_01_28.doc“	150
9.1.1	Schritt 1: Die Word-Datei im E-Mail-Anhang	150
9.1.2	Schritt 2: So schlägt der VBA-Code der Word-Datei zu	155
9.1.3	Schritt 3: Wie der VBA-Code weiteren Schadcode aus dem Internet nachlädt	159
9.1.4	Aktivitätsdauer des Makro-Virus	161
9.2	Excel: Makro-Virus „Buchung_16.xlsm“	162
9.2.1	Schritt 1: Die Excel-Datei im E-Mail-Anhang	163
9.2.2	Schritt 2: So schlägt der Excel-4-Makro-Code zu	168
9.2.3	Schritt 3: Wie der Excel-4-Makro-Code weiteren Schadcode aus dem Internet nachlädt	170
9.2.4	Aktivitätsdauer des Makro-Virus	172
10.	Weitere Gruppenrichtlinien zur Office-Sicherheit	173
10.1	Microsoft Office allgemein	173
10.2	Microsoft Excel	177
10.3	Microsoft PowerPoint	184
10.4	Microsoft Word	191
10.5	Microsoft Access	197
11.	Linksammlung	199
12.	Übersicht der Gebietsschema-ID (LCID)	202
13.	Beispieldateien	209
13.1	Ordner „01_VBA-Code-Beispiele“	209
13.2	Ordner „02_Unschaedliche-Schadprogramme“	212
13.3	Ordner „03_Word-Versionsdateien“	213
13.4	Ordner „04_Testdateien-fuer-Code-Signing-Zertifikate“	215
13.5	Ordner „05_Zertifikate“	216
14.	Erweiterungen/Versionen	218
14.1	Geplante Erweiterungen dieser Dokumentation	218
14.2	Versionsverlauf der Dokumentation	218
	Abbildungsverzeichnis	222
	Über den Autor Kontakt	228
	Impressum	229

Sämtliche Screenshots, Kommandos und Einstellungen dieser Dokumentation basieren im Wesentlichen auf Windows 10 und Office 365, gelten aber auch für Office LTSC 2021, Office 2021 (nachfolgend nur als Office 2021 bezeichnet), Office 2019 und Office 2016 und 2013.

Die Office-Versionen 2010 und 2007 sind nicht mehr Bestandteil dieser Dokumentation, da deren Support am 13. Oktober 2020 (Office 2010) bzw. am 10. Oktober 2017 (Office 2007) geendet hat.

Alle Pfadangaben basieren auf der zum Zeitpunkt der Veröffentlichung dieses Dokumentes (01.11.2021) aktuellen Windows-10/11-Pro-Version.

1. Warum Office-Sicherheitseinstellungen unterschätzt werden – mit fatalen Konsequenzen

„Die Haustür ist mit Mehrfachverriegelung abgesichert und perfekt geschützt. Schade, dass die Fenster nur angelehnt sind und die Kellertüre hinten am Haus offensteht.“

Was sich wie eine Plattitüde liest, ist in aktuellen Office-Sicherheitseinstellungen leider die Regel. Da wird die Makro-Sicherheit aktiviert oder sogar nur digital signierter VBA-Code zugelassen. Um im gleichen Atemzug ganze Serverstrukturen als vertrauenswürdig einzustufen oder leicht überlistbare vertrauenswürdige Dokumente zuzulassen. (Makro-)Viren kompromittieren so im Handumdrehen das ganze System.

Die häufig anzutreffende Meinung, dass Schadprogramme ins eigene Netz nicht eindringen können, da schließlich hochsensible Sicherheitspakete wie Firewalls, umfangreiche Filterpakete in sämtlichen Internetschnittstellen wie E-Mail-Servern, FTP-Zugängen etc. oder geschützte USB-Ports den Zugang unterbinden, darf durchaus in Frage gestellt werden. Schließlich waren die IT-Abteilungen der in der vergangenen Zeit befallenen Unternehmen, Verlage und (Bundes-)Behörden alles andere als blauäugig oder untätig und hatten umfangreiche Schutzmaßnahmen im Einsatz.



Abbildung 1: Vor dem Befall durch Schadprogramme wie Emotet ist niemand sicher – das Thema ernst zu nehmen, sollte für alle IT-Abteilungen Pflicht sein (Quelle: Süddeutsche Zeitung vom 18.12.2019).

Generell haben alle Schutzmaßnahmen einen Pferdefuß: Sie schützen nur das, was ihnen bekannt ist. Schafft es ein neues Schadprogramm, sich in den ersten Stunden bis zur Entdeckung durch die Sicherheitshersteller zu verbreiten – beispielsweise wie bei einigen Emotet-Varianten als Makro-Virus in Form von Office-Datei-E-Mail-Anhängen –, stehen Tür und Tor offen. Irgendjemanden wird es immer geben, der in gutem Glauben den E-Mail-Anhang öffnet und dem Virus Zutritt gewährt. Nicht umsonst lautet das Resümee der Emotet-Geschädigten: **Wir werden einen Befall nicht verhindern können, wir können aber sehr wohl den Schaden minimieren.**

Das Thema „Office-Sicherheit“ spielt eine wichtige Rolle: Da Schadprogramme nach wie vor häufig als Office-Datei per E-Mail-Anhänge verteilt werden, sollten Sie sicherstellen, dass den Schadprogrammen alle bekannten Wege verwehrt werden und Sie über eine umfassende Absicherung verfügen. In den Office-Programmen stehen wie in den Kapiteln dieser Dokumentation beschrieben unzählige Schutzmechanismen zur Verfügung, die den Befall wirkungsvoll verhindern können.

Notwendig ist hierzu aber eine Betrachtung aller Sicherheitseinstellungen. Das Aktivieren der anfangs erwähnten Makro-Sicherheit ist deshalb auch nur ein Punkt von vielen, da sich die Makro-Sicherheit nur auf die nicht vertrauenswürdigen Speicherorte bezieht. Ein Blick in die Trustcenter-Einstellungen der Office-Installationen zeigt hier und an weiteren Punkten bei fast allen Office-Installationen eklatante Sicherheitslücken.

Resümee: Dass es sicher nicht zu den Lieblingsaufgaben der IT gehört, sich nun auch ausgerechnet noch mit Office bzw. der Office-Sicherheit herumzuschlagen, ist aufgrund der vielen anderen Aufgaben verständlich. In Zeiten von Emotet und Co. aber im Hinblick auf die IT-Sicherheit jedoch wenig zielführend. Erst recht sind das nicht die Empfehlungen des BSI (siehe Kapitel 11 „[Linksammlung](#)“), das allen Ernstes vorschlägt, die komplette VBA-Umgebung abzuschalten und somit alle VBA-basierende Workflow-Automatisierung im Unternehmen lahmlegt.

Erst wenn alle einzelnen Office-Sicherheitseinstellungen in Summe Beachtung finden und mit den unternehmenseigenen Arbeitsweisen in Einklang gebracht werden, führt dies zu einem Sicherheitskonzept, das seinen Namen auch verdient. Nur wer weiß, wie Autofunktionen in VBA funktionieren (siehe Kapitel 2 „[Was ist ein Makro, was ist VBA?](#)“) und sich ein Bild über die Funktionsweise eines Makro-Virus macht (siehe Kapitel 9 „[Makro-Virus: Funktionsweise entschlüsselt](#)“), kann erfolgreich weitere Erkenntnisse zu den passenden Sicherheitseinstellungen gewinnen.

Und zum Schluss: So geht es definitiv nicht, wie die ernst gemeinte Antwort auf eine Frage im Office-365-User-Group-Forum glauben machen will:

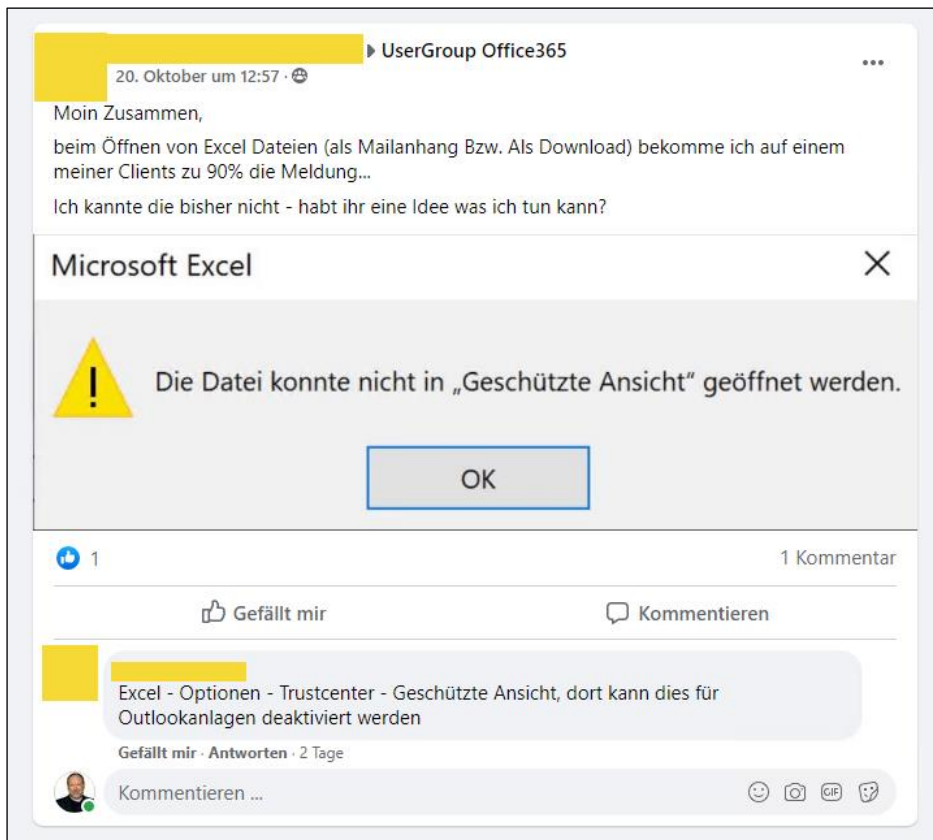


Abbildung 2: Sicherheit? Einfach mal ausschalten, dann hat man Ruhe. Fatale Lösung!

Die Sicherheit einfach auszuschalten, mag zwar komfortabel sein, rächt sich aber garantiert. Wie bereits beschrieben ist es letztlich eine Frage der Zeit, bis ein Schadprogramm zuschlägt und sich die ausgeschaltete Sicherheitsmeldung zu Nutze macht.

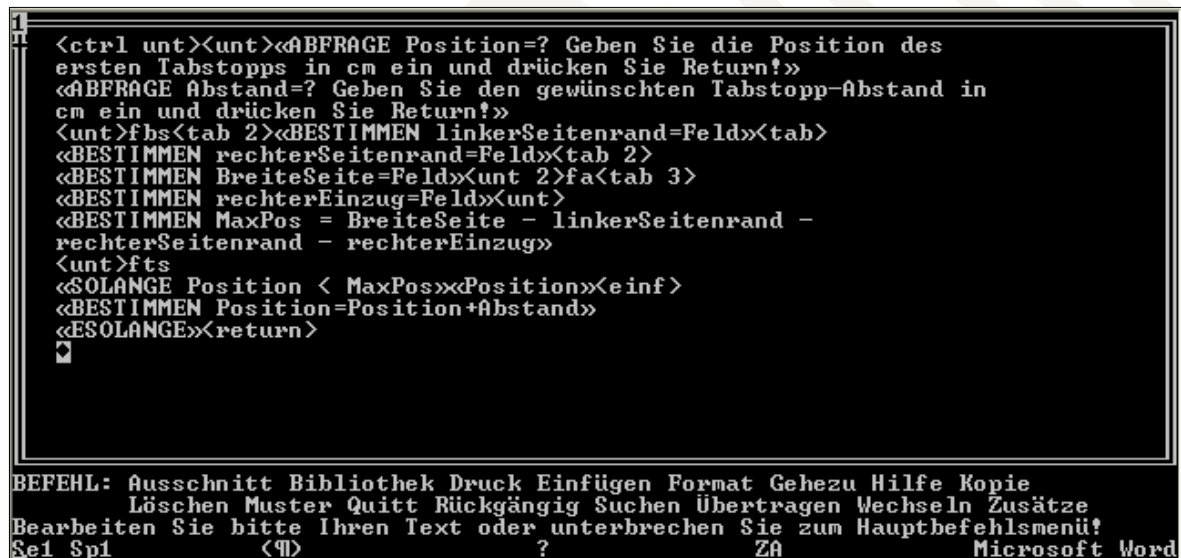
2. Was ist ein Makro, was ist VBA?

Was sich hinter dem Begriff **Makro** verbirgt, lässt sich ganz allgemein zusammenfassen:

Ein Makro ist eine Aufzeichnung von Menü-/Funktionsaufrufen und Tastatureingaben in einem Programm (heutzutage als App bezeichnet). Die Aufzeichnung kann anschließend beliebig oft wiedergegeben werden, um so immer wiederkehrende Prozesse zu automatisieren.

Makros gibt es schon seit den Anfangstagen der Microsoft-Anwendungsprogramme (sowie natürlich auch Nicht-Microsoft-Programme wie Lotus 1-2-3, WordPerfect, Harvard Graphics etc.), als die damaligen Programme immer leistungsfähiger wurden und der Wunsch von Seiten der Anwender entstand, zur Zeiteinsparung sich wiederholende Aufgaben zu automatisieren.

Bereits zu DOS-Zeiten verfügte die Textverarbeitung Word oder das damalige Microsoft-Kalkulationsprogramm Multiplan über Makro-Funktionen.



```

<ctrl><unt><unt><ABFRAGE Position=? Geben Sie die Position des
ersten Tabstopps in cm ein und drücken Sie Return!>
<ABFRAGE Abstand=? Geben Sie den gewünschten Tabstopp-Abstand in
cm ein und drücken Sie Return!>
<unt><fbs<tab 2><BESTIMMEN linkerSeitenrand=Feld><tab>
<BESTIMMEN rechterSeitenrand=Feld><tab 2>
<BESTIMMEN BreiteSeite=Feld><unt 2><fa<tab 3>
<BESTIMMEN rechterEinzug=Feld><unt>
<BESTIMMEN MaxPos = BreiteSeite - linkerSeitenrand -
rechterSeitenrand - rechterEinzug>
<unt><fts
<unt><fts
<SOLANGE Position < MaxPos><Position><einf>
<BESTIMMEN Position=Position+Abstand>
<ESOLANGE><return>

```

BEFEHL: Ausschnitt Bibliothek Druck Einfügen Format Gehezu Hilfe Kopie
Löschen Muster Quitt Rückgängig Suchen Übertragen Wechseln Zusätze
Bearbeiten Sie bitte Ihren Text oder unterbrechen Sie zum Hauptbefehlsmenü!
Sei Spl <9> ? ZA Microsoft Word

Abbildung 3: Die Makro-Sprache zu Zeiten von Word für DOS.

Die Windows-Anwendungen um Excel und später auch Word und PowerPoint wurden ebenfalls mit Makro-Funktionen erweitert. Schon bald konnten Makros auch nachbearbeitet und später dann programmiert werden. Hierzu verfügte anfangs jedes Programm über seine eigene, spezifische Skriptsprache. Diese war nicht oder nur in Teilbereichen mit anderen Programmen kompatibel, der Funktionsumfang im Vergleich zu aktuellen Programmiersprachen war eher zurückhaltend.

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

3. Makroschutz: Ungewollte Ausführung verhindern

Der in Kapitel 2. „[Was ist ein Makro, was ist VBA?](#)“ beschriebene Leistungsumfang lässt sich leider nicht nur für nützliche Funktionen wie die Vorlagenverwaltung, automatische Datenverarbeitung etc. nutzen. Insbesondere die in Kapitel 2.5 „[Leistungsumfang von VBA](#)“ beschriebenen Auto-Funktionen machen die VBA-Umgebung von Office für Programmierer von Schadprogrammen interessant.

3.1 Makro-Viren – Schadprogramme im Office-Datei-Format

Befindet sich beispielsweise in einer als **Rechnung.docm** bezeichneten, via E-Mail versendeten Word-Datei VBA-Code, ist die Wahrscheinlichkeit groß, dass die Mehrzahl der Empfänger die Datei ohne weitere Kontrolle sofort öffnet. Wird der VBA-Code via Auto-Funktion jetzt automatisch gestartet, kann er sofort – für den Anwender unsichtbar – beliebige Funktionen auf dem PC/Mac ausführen.

Solche Schadprogramme gab es in der Vergangenheit immer wieder, sie wurden unter dem Begriff „Makro-Virus“ bekannt (wenngleich es sich richtigerweise um VBA-Viren handelt). Der erste weltweit bekannte Fall war im März 1999 das Makro-Virus Melissa, das beim Öffnen einer Word-Datei sämtliche Empfänger aus Outlook ermittelt und diese angemailt hat. Der geschätzte Schaden lag weltweit bei rund 80 Millionen Dollar (Quelle: [https://de.wikipedia.org/wiki/Melissa_\(Computervirus\)](https://de.wikipedia.org/wiki/Melissa_(Computervirus))). Der bekannteste und gefährlichste aktuelle Vertreter (Stand Q1 2020) der sich der Makro-Funktion bedient, ist Emotet (siehe <https://de.malwarebytes.com/emotet>).

Während lange Zeit an der Makro-Viren-Front Ruhe herrschte, haben diese in den letzten drei Jahren wieder enorm zugelegt. Aktuelle Makro-Viren arbeiten jetzt deutlich filigraner und subtiler. Sie nutzen zum einen das alte Word-Binärformat **.doc**, bei dem am Dateinamen nicht zu erkennen ist, dass sich darin VBA-Code befindet.

Damit die Makro-Viren nicht von Schutzprogrammen erkannt werden, lädt der VBA-Code in der **.doc**-Datei den eigentlichen Schadcode erst zu einem späteren Zeitpunkt von unterschiedlichen Internetservern nach; oft durch Ausführung von Windows-Powershell-Kommandos, denen – bei nicht ausreichender Einschränkung der Benutzerrechte – die ganze Welt erschreckend offen steht. Außerdem kopiert sich der VBA-Code von einer Word-Datei zur nächsten, sodass sich der Schadcode in Windeseile im ganzen Netzwerk verteilt.

Da der ganze Vorgang schrittweise erfolgt und das eigentliche Makro-Virus – sprich der VBA-Code in der Word-Datei – immer wieder modifiziert wird, ist es für Schutzprogramme sehr schwer, die Oberhand zu behalten.

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

4. Digital signierten VBA-Code nutzen

Eine weitere Schutzstufe beim Einsatz von VBA-Code ist das digitale Signieren des Codes. Mit aktiver Code-Signing-Signatur lassen sich die Office-Programme – wie in Kapitel „[3.2 Makro-Schutzeinstellungen – was wie einstellen?](#)“ beschrieben – so einstellen, dass VBA-Code nur mit hinterlegter digitaler Signatur ausgeführt wird. Da der Erwerb einer digitalen Signatur strengen Regeln unterworfen ist, müssen Makro-Viren-Programmierer zumindest eine weitere Hürde nehmen, um eine digitale Signatur zu erwerben und den Schadcode damit zu signieren.

Hinweise:

- Beim Einsatz von digital signiertem VBA-Code unbedingt die in Kapitel 5 „[Vertrauenswürdige Speicherorte: Auswirkungen auf die Sicherheitseinstellungen](#)“ beschriebenen Zusammenhänge von signiertem VBA-Code und vertrauenswürdigen Speicherorten/Dokumenten beachten!
- Einen schnellen Überblick über die unterschiedlichen Sicherheitseinstellungen bietet die Ergebnismatrix in Kapitel 7 „[Übersicht: Wann wird VBA-Code ausgeführt?](#)“.
- Bei signiertem VBA-Code unbedingt sicherstellen, dass das Zertifikat auch als vertrauenswürdiger Herausgeber vorhanden ist, da sonst zwar der signierte Code ausgeführt wird, dies aber mangels Herausgeber-Zertifikat immer zu einer gelben Warnmeldung führt.
- Insbesondere beim Einsatz von Excel-Inplace-Lösungen mit VBA-Code im Rahmen der SAP-GUI ist der Einsatz von signiertem VBA-Code unerlässlich.

4.1 Wie funktioniert ein digitales Zertifikat?

Ein digitales Zertifikat besteht immer aus zwei Teilen: dem öffentlichen und dem privaten Schlüssel. Die Schlüssel liegen als Datei in einem fest definierten Format vor und enthalten letztlich u. a. eine vom Zertifikat abhängige Zeichenfolge.

Ein digitales Zertifikat wird von einer Zertifizierungsstelle (im Beispiel „QuoVadis“: <https://www.quovadisglobal.de/>) erzeugt. QuoVadis – seit Januar 2019 eine 100%ige Tochtergesellschaft von DigiCert – sorgt dabei als offiziell registrierter, weltweit im Einsatz befindlicher Zertifizierer für die Veröffentlichung des öffentlichen Schlüssels.

Ein öffentlicher Schlüssel kann bereits fester Bestandteil jeder Windows-Installation sein und wird dann im Rahmen der Windows-Updates automatisch aktuell gehalten. Die Mehrzahl der öffentlichen Schlüssel wird beim erstmaligen Gebrauch automatisch aus dem Internet auf den PC übertragen.

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

5. Vertrauenswürdige Speicherorte: Auswirkungen auf die Sicherheitseinstellungen

Bei den Einstellungen der in Kapitel 3 „[Makroschutz: Ungewollte Ausführung verhindern](#)“ und Kapitel 4 „[Digital signierten VBA-Code nutzen](#)“ beschriebenen Sicherheitsfunktionen sind immer auch die vertrauenswürdigen Speicherorte zu beachten, da diese bei nicht sachgemäßer Konfiguration die anderen Sicherheitsfunktionen vollständig torpedieren und alle Bemühungen zunichtemachen.

Eingestellt werden die vertrauenswürdigen Speicherorte in den Office-Programmen Access, Excel, PowerPoint und Word in den Optionen (**Datei | Optionen**) unter **Trust Center | Einstellungen für das Trust Center | Vertrauenswürdige Speicherorte**.

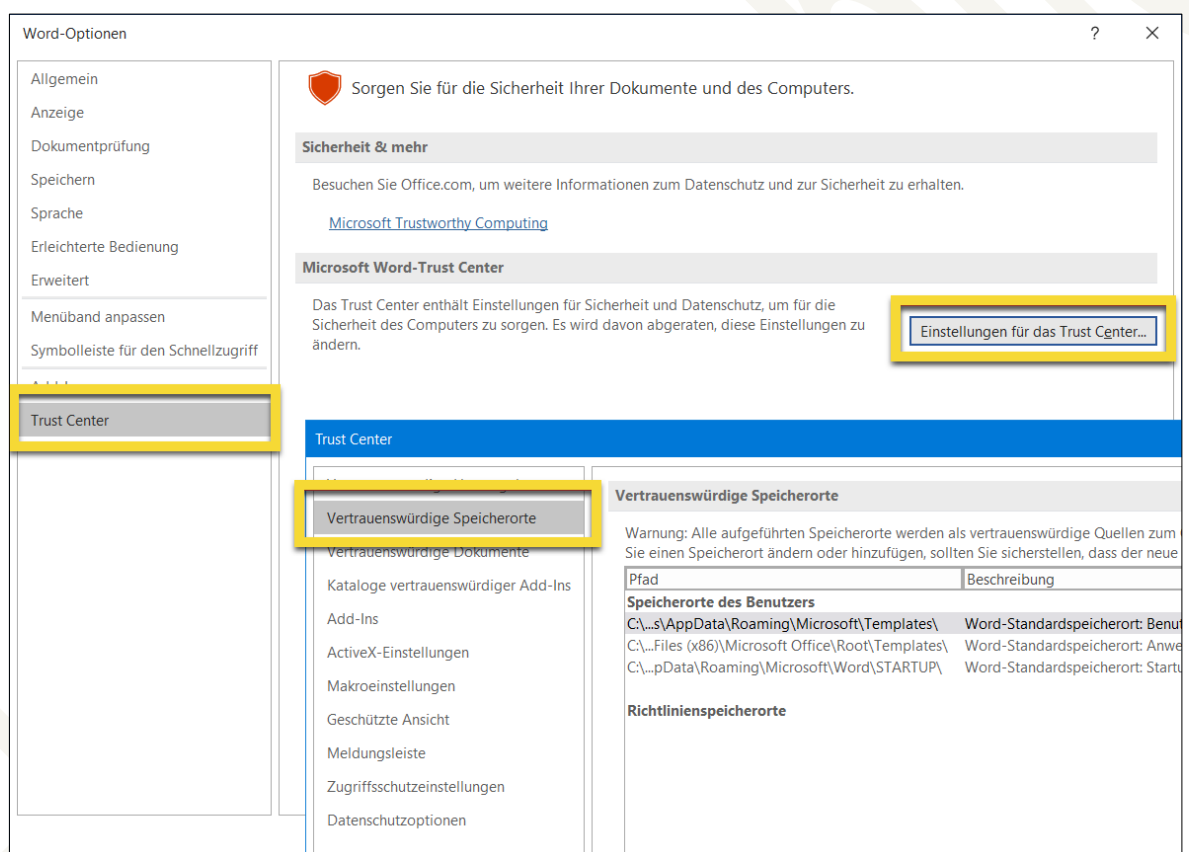


Abbildung 36: Vertrauenswürdige Speicherorte werden in den Optionen festgelegt.

5.1 So wirken sich vertrauenswürdige Speicherorte aus

Vertrauenswürdige Speicherorte haben direkten Einfluss darauf, wie die Office-Programme Access, Excel, PowerPoint und Word mit VBA-Code in den Office-Dateien umgeht. Zu beachten ist, dass in den Dateien enthaltener XML-Code für die Menüband-Anpassungen immer aktiv ist

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

6. Vertrauenswürdige Dokumente: Auswirkungen auf die Sicherheitseinstellungen

Neben den Makroschutzeinstellungen (siehe Kapitel 3 „[Makroschutz: Ungewollte Ausführung verhindern](#)“ und Kapitel 4 „[Digital signierten VBA-Code nutzen](#)“) und den vertrauenswürdigen Speicherorten (siehe Kapitel 5 „[Vertrauenswürdige Speicherorte: Auswirkungen auf die Sicherheitseinstellungen](#)“) stellt der dritte Sicherheitspfeiler in Office die vertrauenswürdigen Dokumente dar. Ist ein Dokument – respektive der VBA-Code in einem Dokument – einmal als vertrauenswürdig definiert, so merkt sich Word die Einstellung im Registry-Zweig des Benutzers und behält sie dauerhaft bei.

Sowohl unter Excel als auch unter PowerPoint und Word werden die Dateien in Bezug auf die Vertrauenswürdigkeit als **Dokumente** bezeichnet – auch wenn es sich bei Excel um Arbeitsmappen und bei PowerPoint um Präsentationen handelt.

Werden im Nachhinein beispielsweise die Makroschutzeinstellungen oder die vertrauenswürdigen Speicherorte geändert, gilt die Vertrauenswürdigkeit der Dokumente unverändert und überlagert so die beiden anderen Sicherheitseinstellungen.

Eingestellt werden die vertrauenswürdigen Dokumente in den Office-Programmen Excel, PowerPoint und Word in den Optionen (**Datei | Optionen**) unter **Trust Center | Einstellungen für das Trust Center | Vertrauenswürdige Dokumente**.

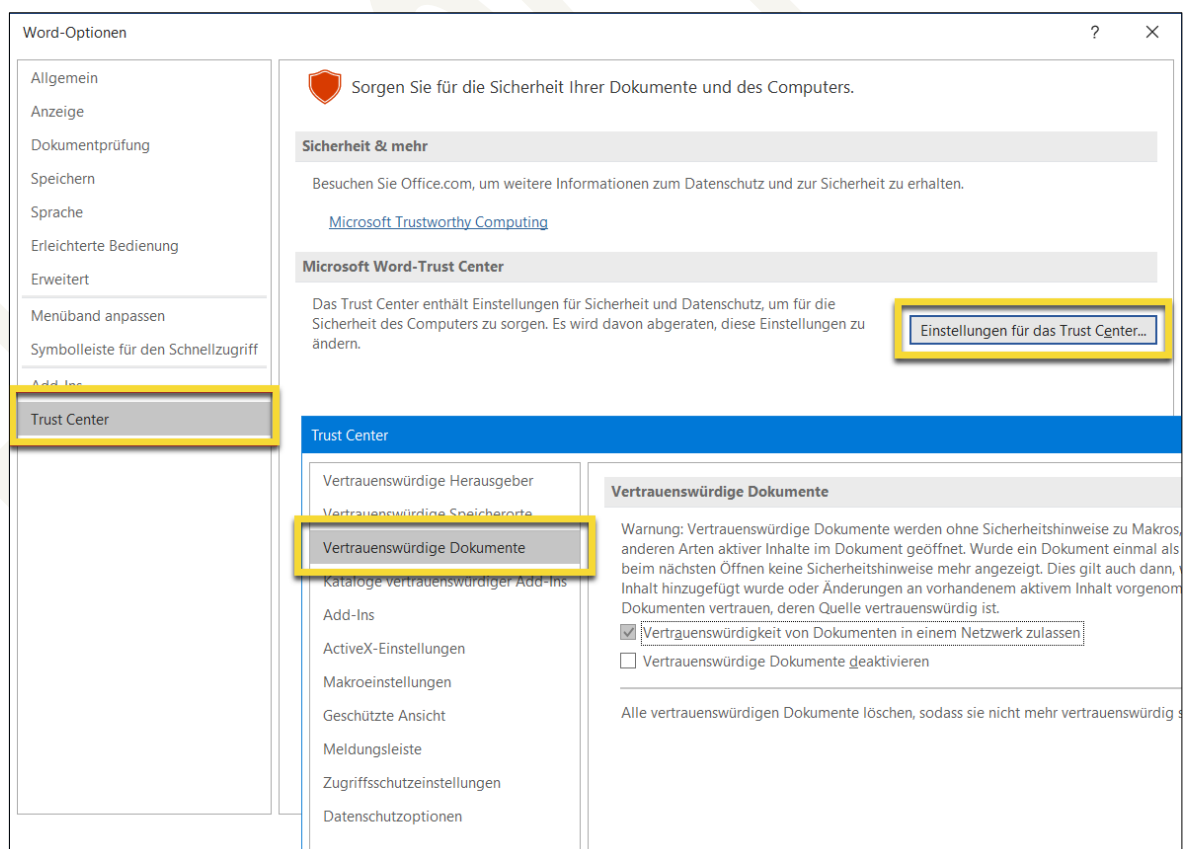


Abbildung 71: Vertrauenswürdige Speicherorte werden in den Optionen festgelegt.

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

7. Übersicht: Wann wird VBA-Code ausgeführt?

In den beiden folgenden Tabellen in Kapitel [7.1 „Ergebnismatrix bei nicht zertifiziertem VBA-Code“](#) und Kapitel [7.2 „Ergebnismatrix bei zertifiziertem VBA-Code“](#) ist zusammengefasst, wie sich VBA-Code unter Berücksichtigung der folgenden Sicherheitseinstellungen am Beispiel von Word und Excel verhält:

- Makroschutzeinstellungen (siehe Kapitel [3 „Makroschutz: Ungewollte Ausführung verhindern“](#))
- Digital signierter VBA-Code (siehe Kapitel [4 „Digital signierten VBA-Code nutzen“](#))
- Vertrauenswürdige Speicherorte (siehe Kapitel [5 „Vertrauenswürdige Speicherorte: Auswirkungen auf die Sicherheitseinstellungen“](#))
- Vertrauenswürdige Dokumente (siehe Kapitel [6 „Vertrauenswürdige Dokumente: Auswirkungen auf die Sicherheitseinstellungen“](#))

Beispieldateien: Zur Verifikation der Ergebnisse in den folgenden beiden Kapiteln stehen Beispieldateien zur Verfügung (siehe Kapitel [13.4 „Ordner „04 Testdateien-fuer-Code-Signing-Zertifikate““](#))

Hinweis: Eine Besonderheit bei der Betrachtung von Office-Dateien mit VBA-Code stellen Add-Ins dar (siehe Kapitel [2.3 „Speicherort des VBA-Codes“](#)). Die unautorisierte Ausführung von VBA-Code in Excel-, PowerPoint- und Word-Add-Ins lässt sich via „verwalteten Add-Ins“ schützen (siehe Kapitel [10.2 „Microsoft Excel“](#), Kapitel [10.3 „Microsoft PowerPoint“](#) und Kapitel [10.4 „Microsoft Word“](#)).

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

8. Dateiformate erlauben bzw. sperren (Zugriffsschutzeinstellungen)

Ältere Dateiformate sind in aller Regel anfälliger für den Befall von Schadprogrammen. Sei es, dass sie Sicherheitslücken enthalten oder dass sich so leicht Schadcode in Dateien dieses Dateityps einbetten lässt (Stichwort „Steganographie“). Oder dass sie von außen unerkannt Programmcode (VBA/Makro) enthalten können (.doc contra .docx/.docm, .xls contra .xlsx/.xlsm, .ppt contra .pptx/.pptm). Entsprechend ist es wichtig, nicht benötigte Dateiformate zu deaktivieren, um so Sicherheitslücken zu stopfen.

Muss Excel wirklich noch SYLK-Dateien öffnen können? Sind allen Ernstes nach wie vor Excel-Makro-4-Dateien im Einsatz, sprich wird noch immer auf sicherheitstechnisch problematische Technik aus dem Jahr 1992 gesetzt (siehe Kapitel 9.2 „Excel: Makro-Virus „Buchung 16.xlsm““)? Ist es tatsächlich notwendig, dass Word Dokumente der Version 2.0 für Windows aus dem Jahr 1991 öffnen kann?

Für jedes Office-Programm lässt sich sowohl im Trust Center des jeweiligen Programms (**Datei | Optionen | Trust Center | Einstellungen für das Trust Center | Zugriffsschutzeinstellungen**) sowie via Gruppenrichtlinie detailliert festlegen, welche Dateiformate blockiert, nur in der geschützten Ansicht angezeigt oder geöffnet werden können (siehe Kapitel 8.1.1 „Standardverhalten für den Zugriffsschutz festlegen“, (Excel) Kapitel 8.2.1 „Standardverhalten für den Zugriffsschutz festlegen“ (PowerPoint) und Kapitel 8.3.1 „Standardverhalten für den Zugriffsschutz festlegen“ (Word)).

Doch Vorsicht: Das Sperren mancher Dateitypen hat direkten Einfluss auf Programm-eigene Funktionen, auf Office-übergreifende Funktionen oder auf Funktionen von Drittanwendungen. Werden solche Dateitypen blockiert, funktioniert die entsprechende Funktion nicht mehr. Entsprechend gilt, dass nach dem Blockieren nichtbenötigter Dateitypen umfangreiche Inhouse-Tests notwendig sind und die Einstellungen bei Bedarf nachjustiert werden müssen.

8.1 Microsoft Excel

Nachfolgend die Gruppenrichtlinien zur Festlegung bzw. Einschränkung der erlaubten Dateiformate in Excel sowie zur Festlegung des Excel-Standard-Dateiformats.

8.1.1 Standardverhalten für den Zugriffsschutz festlegen

Kategorie-Pfad	Benutzerkonfiguration Administrative Vorlagen Microsoft Excel 2016 Excel-Optionen Sicherheit Trust Center Einstellungen für den Zugriffsschutz
----------------	--

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

9. Makro-Virus: Funktionsweise entschlüsselt

Welche Office-Sicherheitseinstellung wie konfigurieren? Für diese Frage gibt es – wie in den vorhergehenden Kapiteln in den unterschiedlichsten Formen beleuchtet – keine einheitliche Antwort. Letztlich müssen alle Faktoren der Unternehmens-Arbeitsumgebung berücksichtigt werden, um einerseits die Sicherheit maximal zu erhöhen, andererseits den „Störfaktor“ Sicherheit beim Office-Einsatz zu minimieren.

Zur Entscheidungsfindung sind in diesem Kapitel die Funktionsweisen von Makro-Viren beschrieben und mit Hintergrundinformationen kommentiert – beginnend von der Meldung, die der Anwender angezeigt bekommt und so den ganzen Vorfall startet, bis hin zum eigentlichen Befall des PCs. So können Sie sich selbst ein Bild machen und für Ihr Unternehmen die geeigneten Schlüsse ziehen.

Auch wenn die „Lebensdauer“ von Makro-Viren meist nur wenige Stunden beträgt, bevor diese von den Schutzprogramm-Anbietern erkannt und herausgefiltert werden: Die Zeitspanne reicht aus, um eine relevante Gefahr darzustellen.

9.1 Word: Makro-Virus „info_01_28.doc“

In den folgenden Abschnitten sind die Funktionsweise, der Befall sowie Hintergrundinformationen zum Makro-Virus **info_01_28.doc** zusammengefasst.

9.1.1 Schritt 1: Die Word-Datei im E-Mail-Anhang

Der Befall des PCs beginnt über einen E-Mail-Anhang. Im spärlichen E-Mail-Text ist zu lesen, dass noch eine Rechnung offen ist und bitte sofort zu begleichen sei – man wolle dann auch auf Mahngebühren verzichten.

Die Zeiten, in denen E-Mails mit Schadprogrammanhängen schon an schlechtem Deutsch, kryptischen Sonderzeichen oder fremden E-Mail-Adressen zu erkennen waren, sind schon lange vorbei. Heute stammen manipulierte E-Mails von vertrauten Absendern, deren E-Mail-Adressen ausgespäht und missbraucht werden. Der Text ist meist kurzgehalten, in perfektem Deutsch und mit echten Absenderdaten hinterlegt. Davon ahnt der aufgeführte Absender nichts, er hat damit überhaupt nichts zu tun. Die Daten wurden schlicht auf Internetseiten oder in E-Mail-Adressbüchern befallener PCs ausgelesen und missbraucht.

Der klassische Anwender wird entsprechend auf den E-Mail-Anhang **info_01_28.doc** sofort doppelklicken und die Word-Datei somit öffnen.

Word-Dokumente mit der Dateinamenerweiterung **.doc** kamen in den Zeiten von Word 2003 und früher zum Einsatz. Beim **.doc**-Format handelt es sich um ein binäres Dateiformat, in dem die Daten stark verschachtelt sind, um möglichst kompakte Dateien zu erhalten (was mit Ausnahme von eingebundenen Bildern auch perfekt funktionierte).

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

10. Weitere Gruppenrichtlinien zur Office-Sicherheit

Neben den in den vorherigen Kapiteln beschriebenen Gruppenrichtlinien sind weitere Sicherheits-Gruppenrichtlinien vorhanden, die nicht im Userinterface des jeweiligen Office-Programms zur Auswahl stehen. Nachfolgend deshalb in loser Reihenfolge die entsprechenden Gruppenrichtlinien mit den jeweiligen Handlungsempfehlungen.

10.1 Microsoft Office allgemein

Benutzeroberflächenerweiterung von Dokumenten und Vorlagen deaktivieren

Kategorie-Pfad	Benutzerkonfiguration Richtlinien Administrative Vorlagen Microsoft Office 2016 Globale Optionen Benutzerdefiniert
In den Office-Anwendungen zu finden unter	–
Bedeutung/Einstellung	In Vorlagen, Arbeitsmappen/Präsentationen/Dokumenten und Add-Ins von Excel, PowerPoint und Word, in Access-Datenbanken, sowie in den Dateien von Outlook, Publisher, Project, Visio und InfoPath kann XML-Code für Benutzeroberflächenerweiterungen, sprich zur Anpassung des Menübands, der Kontextmenüs, der Backstage oder zur Überlagerung von programmeigenen Funktionen zum Einsatz kommen.
Hinweise	<ul style="list-style-type: none"> ■ In der Gruppenrichtlinie legen Sie für jedes der zuvor aufgeführten Programme individuell fest, ob die Benutzeroberflächenerweiterungen aktiviert werden und somit sichtbar sind. ■ Die Gruppenrichtlinie wirkt sich nicht auf .NET-basierende Add-Ins mit integrierten Benutzeroberflächenerweiterungen aus. ■ Bei den Menübandanpassungen lassen sich sowohl zusätzliche Schaltflächen/Gruppen/Registerkarten hinzufügen als auch vorhandene Schaltflächen/Gruppen/Registerkarten ausblenden bzw. deaktivieren. Auch das Hinzufügen eigener, individueller Registerkarten ist möglich. ■ Der XML-Code für die Benutzeroberflächenerweiterung befindet sich bei Excel, PowerPoint und Word im Root-Ordner \customUI der gezippten XML-Struktur der Vorlage, der Datei oder des VBA-Add-Ins. In Access-Datenbanken wird der XML-Code in der Systemtabelle USSRibbons hinterlegt. ■ Im XML-Code wird bei benutzerdefinierten Funktionen via onAction-Attribut die via Wert übergebene (VBA-)Routine aktiviert. Beispiel:

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

11. Linksammlung

Befehlszeilenoptionen in Office: Zusammenfassung

Quelle: Microsoft

<https://support.office.com/de-de/article/befehlszeilenoptionen-für-microsoft-office-produkte-079164cd-4ef5-4178-b235-441737deb3a6>

BSI-Dokumentation zu „Erste Hilfe bei einem schweren IT-Sicherheitsvorfall“

Quelle: Bundesamt für Sicherheit in der Informationstechnik

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Ransomware/Erste-Hilfe-IT-Sicherheitsvorfall.pdf?__blob=publicationFile&v=3

BSI-Empfehlung zur sicheren Konfiguration von Access 2013/2016/2019

Quelle: Bundesamt für Sicherheit in der Informationstechnik

https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_140.html

BSI-Empfehlung zur sicheren Konfiguration von Excel 2013/2016/2019

Quelle: Bundesamt für Sicherheit in der Informationstechnik

https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_136.html

BSI-Empfehlung zur sicheren Konfiguration von Office 2013/2016/2019

Quelle: Bundesamt für Sicherheit in der Informationstechnik

https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_135.html

BSI-Empfehlung zur sicheren Konfiguration von Outlook 2013/2016/2019

Quelle: Bundesamt für Sicherheit in der Informationstechnik

https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_139.html

BSI-Empfehlung zur sicheren Konfiguration von PowerPoint 2013/2016/2019

Quelle: Bundesamt für Sicherheit in der Informationstechnik

https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_137.html

BSI-Empfehlung zur sicheren Konfiguration von Word 2013/2016/2019

Quelle: Bundesamt für Sicherheit in der Informationstechnik

https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_138.html

Datenbank mit URLs, die Schadprogramme verteilen

Quelle: URLhaus/abuse.ch

<https://urlhaus.abuse.ch/>

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

12. Übersicht der Gebietsschema-ID (LCID)

Nachfolgend eine Übersicht der Gebietsschema-IDs, kurz LCID (Language Code ID). Die Tabelle enthält die LCID sowohl als Dezimal-Wert (wie die LCID beispielsweise bei Vorlagenpfaden genutzt wird (siehe Kapitel 5) als auch als Hexadezimal-Wert (wie die LCID bei den Ausschlusswörterbüchern genutzt wird).

Sprache	LCID (dezimal)	LCID (hexadezimal)
Afrikaans	1078	0436
Albanisch	1052	041c
Elsässisch	1156	0484
Amharisch	1118	045e
Arabisch (Ägypten)	3073	0c01
Arabisch (Algerien)	5121	1401
Arabisch (Bahrain)	15361	3c01
Arabisch (Irak)	2049	0801
Arabisch (Jemen)	9217	2401
Arabisch (Jordanien)	11265	2c01
Arabisch (Katar)	12289	3001
Arabisch (Kuwait)	4097	1001
Arabisch (Libanon)	6145	1801
Arabisch (Libyen)	8193	2001
Arabisch (Marokko)	16385	4001
Arabisch (Oman)	1025	0401
Arabisch (Saudi-Arabien)	10241	2801
Arabisch (Syrien)	7169	1c01
Arabisch (Tunesien)	14337	3801
Arabisch (Vereinigte Arabische Emirate)	9217	2401
Armenisch	1067	042b
Assamisch	1101	044d
Aserbaidshanisch (Kyrillisch)	2092	082c
Aserbaidshanisch (Lateinisch)	1068	042c
Baschkirisch	1133	046d
Baskisch	1069	042d
Belarussisch	1059	0423

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

13. Beispieldateien

Zur Veranschaulichung und zum Test der in dieser Dokumentation gezeigten Funktionen stehen folgende Beispieldateien auf Anfrage zur Verfügung:

13.1 Ordner „01_VBA-Code-Beispiele“

Beispieldateien für Word und Excel zur Dokumentation der via VBA verfügbaren Funktionen, die beim Öffnen der Dateien automatisch ausgeführt werden.

Dateiname	Beschreibung
Hahner__Ereignis_Dokument.docm	<p>Datei im Word-docm-Format = Word-Dokument mit integriertem VBA-Code.</p> <p>Im VBA-Code sind zahlreiche Routinen enthalten, die bei der Ausführung entsprechender Events zum Einsatz kommen und dann eine Meldung anzeigen. So lässt sich anhand der Meldungen leicht nachvollziehen, wie Word innerhalb von Dokumenten beim Öffnen, Schließen etc. selbstständig Automatismen starten kann.</p> <p>Siehe Kapitel 2.5.2 „Auszug aus den Word-Events zur Automatisierung“.</p>
Hahner__Ereignis_Add-In_Dokumentvorlage.dotm	<p>Datei im Word-dotm-Format, die sowohl als Word-Dokumentvorlage als auch als Word-Add-In genutzt werden kann.</p> <p>Im VBA-Code sind zahlreiche Routinen enthalten, die bei der Ausführung entsprechender Events zum Einsatz kommen und dann eine Meldung anzeigen. So lässt sich anhand der Meldungen leicht nachvollziehen, wie Word innerhalb von Dokumentvorlagen und VBA-basierenden Add-Ins selbstständig Automatismen starten kann.</p> <p>Siehe Kapitel 2.5.2 „Auszug aus den Word-Events zur Automatisierung“.</p>
Hahner__Ereignis-Add-In_Arbeitsmappenvorlage.xltn	<p>Datei im Excel-xltn-Format = Excel-Arbeitsmappenvorlage mit integriertem VBA-Code.</p> <p>Im VBA-Code sind zahlreiche Routinen enthalten, die bei der Ausführung entsprechender Events zum Einsatz kommen und dann eine Meldung anzeigen. So lässt sich anhand der Meldungen leicht nachvollziehen, wie Excel beim Aufruf von Funktionen wie dem Öffnen oder Schließen von</p>

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

14. Erweiterungen/Versionen

Nachfolgend sind geplante Erweiterungen sowie durchgeführte Änderungen dieser Dokumentation aufgeführt.

14.1 Geplante Erweiterungen dieser Dokumentation

Folgende Erweiterungen für diese Dokumentation sind bereits in Planung/Arbeit:

- Kapitel zu ActiveX-Einstellungen
- Kapitel zu Office-Optionen, die für den Zugriff auf fremde Daten relevant sind.
- Weitere Erweiterung des Kapitels [10 „Weitere Gruppenrichtlinien zur Office-Sicherheit“](#)
- Kapitel zum COM-/VSTO-Add-Ins, deren Installation und Schutzmaßnahmen
- Erweiterung Kapitel [5.5](#) mit Praxiserfahrungen und -empfehlungen zum Anlegen vertrauenswürdiger Speicherorte

14.2 Versionsverlauf der Dokumentation

- | | | |
|-------|------------|---|
| v1.00 | 13.05.2019 | ■ Basisversion von Team Hahner® |
| v1.01 | 07.06.2019 | ■ Erweiterungen im Kapitel 4.5 „VBA-Code digital signieren“ gemäß Leserrückmeldung |
| v1.02 | 23.06.2019 | ■ Erweiterungen im Kapitel 3.1 „Makro-Viren – Schadprogramme im Office-Datei-Format“ gemäß Leserrückmeldung |
| v1.03 | 04.07.2019 | <ul style="list-style-type: none"> ■ Korrekturen zu Outlook im Kapitel 2.5 „Aufbau des VBA-Codes“ gemäß Leserrückmeldung ■ Hinzufügen der Pfadangaben für die jeweiligen Speicherorte der Dateien mit VBA-Code im Kapitel 2.3 „Speicherort des VBA-Codes“ ■ Hinzufügen der Kapitel 2.5.1, 0 und 2.5.4 gemäß Leserrückmeldung |
| v1.04 | 07.07.2019 | <ul style="list-style-type: none"> ■ Anpassung Kapitel 2 „Was ist ein Makro, was ist VBA?“ gemäß Leserrückmeldung ■ Austausch der Bildschirmfotos mit Versionen aus der aktuellen Office-365-Version (Version 1906, Monatlicher Kanal). Hinzufügen von Bildunterschriften und einem Abbildungsverzeichnis. ■ Hinzufügen des Kapitels 2.5.3 „Auszug aus den Excel-Events zur Automatisierung“ |

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

Abbildungsverzeichnis

Abbildung 1:	Vor dem Befall durch Schadprogramme wie Emotet ist niemand sicher – das Thema ernst zu nehmen, sollte für alle IT-Abteilungen Pflicht sein (Quelle: Süddeutsche Zeitung vom 18.12.2019).....	6
Abbildung 2:	Sicherheit? Einfach mal ausschalten, dann hat man Ruhe. Fatale Lösung!	8
Abbildung 3:	Die Makro-Sprache zu Zeiten von Word für DOS.....	9
Abbildung 4:	Makro-Programmierung und Dialogfeld-Erstellung bei Word für Windows 2.0 aus dem Jahr 1992.	10
Abbildung 5:	Sämtliche Funktionen zum Thema VBA stehen in den Office-Programmen auf der Registerkarte Entwicklertools zur Verfügung.	11
Abbildung 6:	Der VBA-Editor arbeitet als eigenständige Anwendung und verfügt noch über die klassischen Menüstrukturen.	12
Abbildung 7:	Das Dialogfeld dient zur Verwaltung der Makros (= VBA-Routinen) außerhalb des VBA-Editors.	12
Abbildung 8:	Ist keine Freigabe via Kontrollkästchen Zugriff auf das VBA-Projektmodell vertrauen vorhanden, erscheint diese Meldung beim Versuch, das VBA-Objektmodell via VBA anzusprechen.	21
Abbildung 9:	Für den VBA-Zugriff auf VBA muss der Veweis auf die Applications Extensibility 5.3 gesetzt sein.....	21
Abbildung 10:	Die Makro-Meldung erscheint je nach Einstellung beim Öffnen einer Datei mit VBA-Code.	24
Abbildung 11:	Autofunktionen sorgen dafür, dass beispielsweise beim Öffnen der Datei Makros ausgeführt werden.	25
Abbildung 12:	Makroeinstellung: Praktisch ist leider nicht gleich sicher.	25
Abbildung 13:	Das Kontrollkästchen öffnet Viren Tür und Tor - unbedingt prüfen!	27
Abbildung 14:	QuoVadis ist anerkannter Trust Service Provider (TSP).....	36
Abbildung 15:	Der eToken enthält das Code-Signing-Zertifikat (im Beispiel ein eToken des Herstellers SafeNet, der 2014 von Gemalto übernommen wurde. Gemalto wurde wiederum 2019 von Thales übernommen).	37
Abbildung 16:	Der SafeNet Authentication Client muss zur Zertifizierung des VBA-Codes auf dem Client installiert sein, auf dem der Code zertifiziert wird.	37
Abbildung 17:	Die Fehlermeldung erscheint beim Einsatz des SafeNet Authentication Client in der Version 10.5 oder größer.....	37
Abbildung 18:	Mit der Version 10.4 – noch herausgegeben unter dem Gemalto-Label – ist das Signieren fehlerfrei möglich.	38
Abbildung 19:	Die Installation des SafeNet Authentication Client sorgt automatisch für die Installation des Root-Zertifikats.....	38
Abbildung 20:	Der USB eToken ist nicht eingesteckt , das Symbol im Infobereich erscheint in grau. Die Schaltflächen auf der Startseite des Programmfensters sind inaktiv	39
Abbildung 21:	Der USB eToken ist eingesteckt , das Symbol im Infobereich erscheint in Farbe. Die Schaltflächen auf der Startseite des Programmfensters sind aktiv , links wird jetzt auch der Name des signierenden Unternehmens angezeigt.	39

Die vollständige Dokumentation einschließlich zugehöriger Beispieldateien ist Bestandteil unseres eintägigen Individual-(Online-)Seminars „Office Sicherheit“.

**Kontakt: Markus Hahner
+49 7720 810046
info@hahner.de
www.hahner.de
www.office-sicherheit.de**

Über den Autor | Kontakt



Markus Hahner

Diplom-Ingenieur (FH)

+49 7720 810046

info@hahner.de

www.hahner.de | www.schauen-statt-lesen.de | www.office-sicherheit.de

- Zertifizierter Office-Trainer mit den Schwerpunkten Word, VBA, XML, Office-Sicherheit und Office 365-Apps
- Projekterfahrener Ingenieur für Mittelständler/Großunternehmen beim Office-Rollout
- Entwickler von Vorlagen-Lösungen/Vorlagen mit vielen zehntausend Installationen
- Fachbuchautor mit weltweit über 50 Büchern u. a. bei Microsoft Press; LinkedIn-Learning/video2brain-Trainer zu Word
- Video-Blogger (schauen-statt-lesen.de) und Office-Blogger (hahner.de)



Impressum

© 2013 – 2021, Team Hahner® – Engineers of (Word) Solutions, Dipl.-Ing. (FH) Markus Hahner

Alle Daten urheberrechtlich geschützt. Jegliches Kopieren ist verboten. All copyrights for data reserved. Unauthorized downloading or other kinds of copying prohibited. Keine Weitergabe oder Veröffentlichung in gedruckter oder elektronischer Form ohne ausdrückliche schriftliche Genehmigung von Team Hahner® – Engineers of (Word) Solutions, Dipl.-Ing. (FH) Markus Hahner.

Team Hahner® – Engineers of (Word) Solutions
Dipl.-Ing. (FH) Markus Hahner & Dipl.-Ing. (FH) Christin Starke
Arndtstraße 28
78054 Villingen-Schwenningen
Germany

+49 7720 810046

info@hahner.de

www.hahner.de | www.schauen-statt-lesen.de | www.office-sicherheit.de

Sämtliche Screenshots in dieser Dokumentation wurden mit Snagit® erstellt
(Details siehe www.techsmith.de/snagit.html).



CAMTASIA, SNAGIT UND TECHSMITH SIND MARKEN VON TECHSMITH CORPORATION UND IN DEN USA SOWIE IN ANDEREN LÄNDERN ALS SOLCHE EINGETRAGEN.

Maßgeschneiderte Trainings, Workshops & Online-Seminare
zu Snagit, Camtasia und Audacity | www.screencast-training.de

SCREENCAST  TRAINING

Version: 1.17 vom 01.11.2021