

SSL-MitM Angriffe auf Online-Banking-Applikationen

Das iTAN-Verfahren, session-basierte Formularfeldnamen und teure Extended Validation SSL-Zertifikate, welche die Adressleiste des Webbrowsers grün färben, sollen die Sicherheit beim Online-Banking erhöhen. Doch ist eine für den Benutzer nicht mehr zu durchschauende Manipulation wirklich unmöglich?



Christian Eichelmann
Dipl.-Inform. Matthias Deeg
Dipl.-Inform. Sebastian Schreiber

22. März 2010

Einleitung

Das unsichere TAN-Verfahren wurde längst von den Banken durch das sicherere iTAN-Verfahren abgelöst. Benutzer achten inzwischen auf Zertifikatswarnungen und moderne Webbrowser zeigen bei Webseiten, die *Extended Validation SSL*-Zertifikate verwenden, eine grüne Adressleiste an, die dem Benutzer versichert, dass er wirklich mit dem Server der Bank kommuniziert. Die Betreiber der Online-Banking-Applikationen haben ebenfalls dazugelernt und nutzen als Bezeichner für ihre Formularfelder kryptische Zeichenketten, die sich bei jedem Aufruf des Formulars ändern und die zudem an die aktuelle Browser-Sitzung gebunden sind.

Wenn man nun als Kunde ebenfalls nicht auf gefälschte Webseiten hereinfällt, die den Besucher auffordern, mehrere iTAN-Nummern auf einmal einzugeben, sollte es doch eigentlich nicht mehr gelingen, Online-Überweisungen zu manipulieren. Oder doch?

Das Szenario

Der Benutzer verwendet das Betriebssystem MICROSOFT WINDOWS XP SP3 mit dem Webbrowser INTERNET EXPLORER 8. Er meldet sich ganz normal bei der Kreissparkasse Leipzig an und führt anschließend eine Überweisung in Höhe von 20 Euro mit einer iTAN durch. Auf seinem nächsten Kontoauszug findet er aber anstatt einer Überweisung über 20 Euro eine Überweisung in Höhe von 2000 Euro an ein völlig anderes Konto. Es folgt die Beschreibung eines möglichen Angriffs zur Realisierung dieses Szenarios.

Schritt 1: Ausnutzen einer bekannten Sicherheitslücke

Der bei Sicherheitstests häufig zu kurz kommende Aspekt der Client-Sicherheit gewinnt mehr und mehr an Bedeutung. Es gibt inzwischen zahlreiche frei verfügbare Exploits für alle erdenklichen Client-Anwendungen, wie beispielsweise Webbrowser, Dokumentenbetrachter oder Chat-Applikationen.

Im dargestellten Beispiel wird ein Exploit für den ADOBE ACROBAT READER verwendet, um ein manipuliertes PDF-Dokument zu erstellen, welches beim Öffnen dieses Dokuments eine ausführbare Datei von einem durch den Angreifer kontrollierten Webserver lädt und dieses Programm (Schadsoftware) anschließend auch automatisch startet.

Zu diesem Zweck eignen sich ebenfalls Browser-Exploits, wie beispielsweise "Aurora" (MS10-002¹).

¹<http://www.microsoft.com/technet/security/bulletin/MS10-002.mspx>

Schritt 2: Umleiten des Datenverkehrs

Das Ziel des Angreifers ist die Manipulation einer Online-Überweisung. Um eine solche Manipulation durchzuführen, muss der Angreifer in der Lage sein, den Datenverkehr zwischen dem Bankkunden und der Bank bei einer Online-Überweisung verändern zu können. Dies kann unter anderem durch das Umleiten des gesamten Datenverkehrs der Online-Banking-Applikation über ein vom Angreifer kontrolliertes System erfolgen. In diesem Zusammenhang spricht man daher auch von einem *Man-in-the-Middle*-Angriff (siehe Schritt 4).

Es gibt zwei Methoden, die den Datenverkehr umleiten können:

1. Manipulation der `hosts`-Datei

Die Schadsoftware des Angreifers fügt der `hosts`-Datei² des Betriebssystems einen manipulierten Eintrag für die Namensauflösung des Bank-Webservers hinzu.

Zum Beispiel:

```
10.10.10.10 banking.sparkasse-leipzig.de
```

Dies hat zur Folge, dass das Opfer beim Aufruf der URL

`https://banking.sparkasse-leipzig.de/` nicht mit dem eigentlichen Webserver der Sparkasse Leipzig kommuniziert, sondern mit einem vom Angreifer kontrollierten System.

2. Manipulation des verwendeten DNS-Servers

Die Schadsoftware des Angreifers manipuliert den vom Betriebssystem verwendeten DNS-Server, indem der primäre DNS-Server auf ein vom Angreifer kontrolliertes System gesetzt wird. Anfragen zur Namensauflösung von Domainnamen werden somit durch den DNS-Server des Angreifers beantwortet, der dann für den Webserver der Sparkasse Leipzig eine andere IP-Adresse an das Opfer senden kann.

Die beiden hier vorgestellten Methoden setzen voraus, dass die Schadsoftware des Angreifers über administrative Rechte verfügt.

Des Weiteren hat die Manipulation der `hosts`-Datei den Nachteil, dass gängige Antiviren-Produkte Veränderungen an der `hosts`-Datei erkennen beziehungsweise verhindern. Ein Nachteil der Manipulation des DNS-Servers ist, dass für einen erfolgreichen Angriff die Kommunikation mit externen DNS-Servern in der IT-Umgebung des Opfers prinzipiell möglich sein muss und nicht beispielsweise durch eine Firewall unterbunden wird.

Im Beispiel der SySS GmbH für einen Angriff auf Online-Banking-Applikationen wurde die erste Methode, nämlich die einer Manipulation der `hosts`-Datei gewählt.

²auf WINDOWS-Systemen im Verzeichnis `%WINDIR%\system32\drivers\etc` zu finden

Schritt 3: Manipulation des Windows-Zertifikatsspeichers

Ob es sich bei einem SSL-Zertifikat, das beispielsweise von einem Online-Banking-Webserver verwendet wird, um ein gültiges EV-SSL-Zertifikat handelt, wird durch den Webbrowser ermittelt. Ist eine solche Überprüfung erfolgreich, wird die Adressleiste des Webbrowsers grün gefärbt, wie Abbildung 1 anhand des INTERNET EXPLORER 8 beispielhaft zeigt:

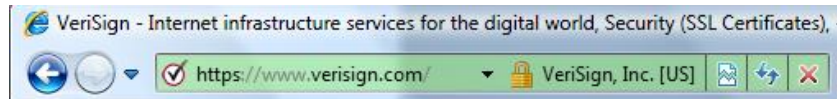


Abbildung 1: Grüne Adressleiste des INTERNET EXPLORER 8

Der INTERNET EXPLORER 8 greift für eine solche Überprüfung auf den Zertifikatsspeicher des WINDOWS-Betriebssystems zu. Darin befindet sich unter anderem auch eine Liste vertrauenswürdiger *Certificate Authorities*, die EV-SSL-Zertifikate ausstellen können.

Martin Christinat von der Firma Keyon hat in seiner Veröffentlichung mit dem Titel *Faking Extended Validation Certificates in Internet Explorer 7* [4] bereits im Jahr 2007 beschrieben, wie sich EV-SSL-Zertifikate für den INTERNET EXPLORER 7 fälschen lassen. Dieselbe Vorgehensweise führt auch für den INTERNET EXPLORER 8, der im Beispiel der SySS GmbH verwendet wurde, zum Erfolg.

Die Schadsoftware des Angreifers importiert ein entsprechend erzeugtes Stammzertifikat des Angreifers als vertrauenswürdige *Certificate Authority* in den WINDOWS-Zertifikatsspeicher. Beim Importieren werden für dieses Zertifikat die beiden Eigenschaften

- CERT_FRIENDLY_NAME_PROP_ID und
- CERT_ROOT_PROGRAM_CERT_POLICIES_PROP_ID

gesetzt.

Schritt 4: Web-MitM mit einem transparenten Reverse-HTTP-Proxy

Wird nun vom Opfer die Webseite <https://banking.sparkasse-leipzig.de/> angefordert, wird in Wirklichkeit eine SSL-Verbindung zu der in der manipulierten `hosts`-Datei angegebenen IP-Adresse aufgebaut. Auf dem vom Angreifer kontrollierten System mit dieser IP-Adresse läuft auf TCP-Port 443 ein HTTPS-Dienst, der ein EV-SSL-Zertifikat verwendet, das von der in den Zertifikatsspeicher importierten *Certificate Authority* des Angreifers signiert wurde.

Die Adressleiste im Webbrowser des Opfers färbt sich nun wie gewohnt grün und es gibt bis auf den Vergleich des Zertifikats-Fingerprint keine Möglichkeit, um zu erkennen, dass es sich hierbei um ein gefälschtes Zertifikat handelt.

Sämtliche Daten werden nun von dem Server des Angreifers aus an die Bank weitergeleitet und die Antworten des Bankservers werden ebenso wieder an den Kunden gesendet. Somit nimmt der Server des Angreifers für die Bank die Rolle des Kunden und für den Kunden die Rolle der Bank ein. Auch für die Bank ist es nicht möglich zu erkennen, dass es sich hierbei nicht um den echten Kunden handelt. Der Angreifer hat somit eine *Man-in-the-Middle*-Position eingenommen.

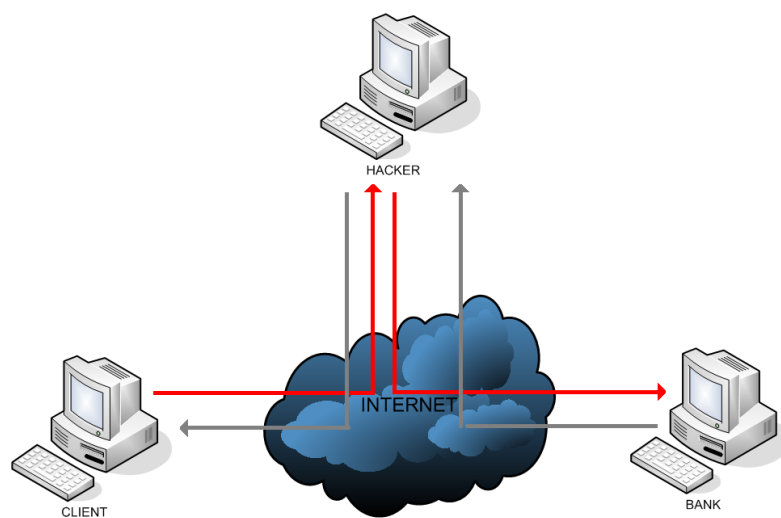


Abbildung 2: Klassisches *Man-in-the-Middle*-Szenario

In dieser Position ist es ohne Probleme möglich, bestimmte Anfragen an den Server wie beispielsweise das Absenden einer Überweisung zu erkennen und die gesendeten Daten zu manipulieren. Wird also vom Opfer eines solchen Angriffs eine Überweisung getätigt, ersetzt der Angreifer einfach die dort angegebene Kontonummer, die Bankleitzahl und den Betrag durch beliebige andere Werte und sendet diese dann an den Server der Bank. Sendet dieser anschließend eine Übersichtsseite zur Bestätigung und fordert zur Eingabe einer bestimmten iTAN auf, so werden die Daten ebenfalls manipuliert, so dass das Opfer wieder die von ihm eingegebenen Daten sieht. Wird die Überweisung nun durch eine gültige iTAN bestätigt, wird die Überweisung mit den Daten des Angreifers durchgeführt. Dies wird womöglich erst Tage später auf dem Kontoauszug bemerkt.

Um dieses Szenario praktisch umzusetzen, wurde von der SySS GmbH ein Programm entwickelt, das diese Aufgaben übernimmt. Dieses Programm ist nicht öffentlich verfügbar.

Fazit

Es ist für einen Angreifer durchaus möglich, erfolgreiche Angriffe auf Online-Banking-Applikationen durchzuführen, die von den Opfern zum Zeitpunkt des Angriffs mit hoher Wahrscheinlichkeit nicht erkannt werden.

Voraussetzung hierfür ist allerdings, dass ein Programm (Schadsoftware) des Angreifers auf dem Rechner des potenziellen Opfers mit administrativen Rechten ausgeführt wird. Dies kann getarnt durch Schwachstellen in gängigen Dateiformaten (Bilder, Dokumenten, etc.) oder auch in alltäglich genutzten Softwareanwendungen (E-Mail-Client, Webbrowser, etc.) geschehen, oder auch leichter zu erkennen durch simples Versenden eines ausführbaren Programms via E-Mail.

Die Wahrscheinlichkeit, dass bei dem Versand eines PDF-Dokuments an 100.000 E-Mailadressen einige Anwender den Anhang öffnen, einen verwundbaren Dokumentenbetrachter verwenden und zudem auf diesen Rechnern auch Online-Banking betreiben, ist als relativ hoch einzuschätzen. Somit ist das beschriebene Angriffsszenario durchaus als realistisch anzusehen.

Quellen

- [1] Schwachstelle im Internet Explorer (Aurora Exploit), <http://www.microsoft.com/technet/security/bulletin/MS10-002.msp>
- [2] Schwachstelle im Adobe Acrobat Reader, <http://cve.mitre.org/cgi-bin/cvename.cgi?name=2009-3953>
- [3] Extended Validation Certificates, <http://de.wikipedia.org/wiki/Extended-Validation-Zertifikat>
- [4] Martin Christinat, *Faking Extended Validation Certificates in Internet Explorer 7*, http://www.keyon.ch/wKeyon/ueber-keyon/Fachartikel/Faking_ExtendedValidation_SSL_Certificates_in_Internet_Explorer_7_V1.1b.pdf 4