Der Fall Kaspersky: Liebesgrüße aus Moskau

Weitere Schwachstellen in Kaspersky-Virenschutz

Nach c't 18/2019 bekamen wir viel Post – darunter Informationen über weitere Kaspersky-Schwachstellen. Es drängt sich die Frage auf, ob Antivirensoftware mit der Manipulation des verschlüsselten Datenverkehrs nicht zu weit geht. Und dann meldete sich auch noch Herr Kaspersky persönlich zu Wort.

Von Ronald Eikenberg

Es dauerte nach dem Erscheinen der vorherigen c't-Ausgabe, in der wir eine Schwachstelle im Kaspersky-Virenschutz aufdeckten, nicht lange, bis sich Wladimir Palant mit uns in Verbindung setzte. Er entwickelt nicht nur den Werbeblocker Adblock Plus, er interessiert sich auch für IT-Sicherheit und hat unabhängig von uns die Schutzprogramme von Kaspersky untersucht. Er stieß dabei auf weitere Lücken, die insbesondere die Art und Weise betreffen, wie Kaspersky den Datenverkehr der Nutzer manipuliert.

Um nämlich den Inhalt verschlüsselter Verbindungen auswerten und verändern zu können, unterbrechen Virenschutz-Programme häufig den TLS/SSL-verschlüsselten Tunnel zwischen Server und Browser. Das ist keine ungefährliche Operation: Dadurch verlagert sich die Validierung der Serverzertifikate und die Aushandlung der Krypto-Parameter vom Browser zur Schutzsoftware. Das kann zu einem geringeren Schutzniveau führen – insbesondere dann, wenn der Virenschutz moderne Krypto-Funktionen noch nicht beherrscht oder bei der Validierung der Zertifikate patzt.

Kaspersky in the Middle

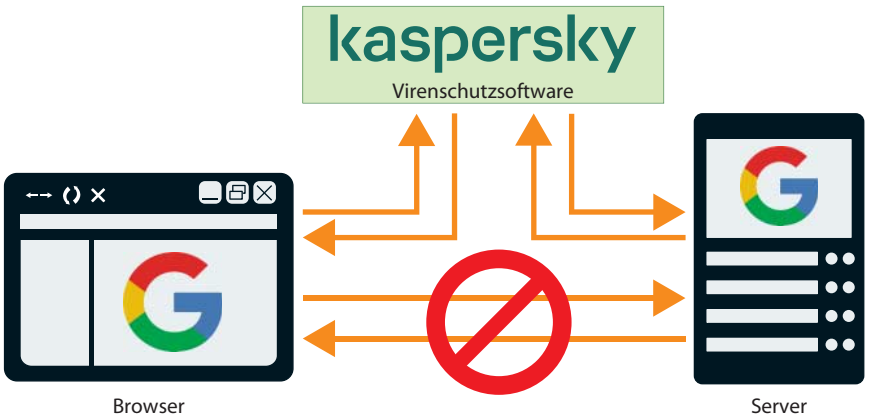
Wie das konkret aussehen kann, zeigt Wladimir Palants Blog-Eintrag „Kaspersky in the Middle – what could possibly go wrong?“ (siehe ct.de/y2q8). Palant berichtet darin etwa, dass Kaspersky die Schutzfunktion „HTTP Strict Transport Security“ (HSTS) unzureichend implementiert hatte. Ist HSTS aktiv und es kommt zu einem Zertifikatsfehler, unterbinden Chrome und Firefox aus Sicherheitsgründen den Zugriff auf die jeweilige Site.

Hatte sich Kaspersky dazwischengeschaltet, konnte Palant diesen Schutz jedoch umgehen und die Seite trotzdem anzeigen lassen. Ein größeres Problem entdeckte Palant im Zusammenspiel mit Microsoft Edge: Es gelang ihm dort, das eingeschleuste Kaspersky-Skript für Universal Cross-Site-Scripting (UXSS) zu missbrauchen. Das heißt im Klartext, dass eine bösartige Website darüber Code im Kontext einer anderen Website ausführen kann – die Webseite des Angreifers könnte durch die Lücke prinzipiell etwa das Webmail-Postfach des Kaspersky-Nutzers durchstöbern. Wladimir Palant begann im Dezember vergangenen Jahres damit, seine Funde an Kaspersky zu melden. Mitte August dieses Jahres gab der Hersteller bekannt, dass die Probleme offenbar mit dem Patch E beseitigt wurden, der bereits im April erschienen war.

Dass es möglicherweise nicht die allerbeste Idee ist, den sensiblen TLS/SSL-Traffic der Nutzer zu manipulieren, zeigt auch eine Studie aus dem Jahr 2017 (siehe ct.de/y2q8), an der neben drei Unis auch Cloudflare, Google und Mozilla beteiligt waren. Im Rahmen der Studie wurde untersucht, in welcher Weise verschiedene Schutzprogramme die Sicherheit von verschlüsselten Verbindungen beeinflussen. Das Ergebnis war desaströs: Die Forscher stellten bei 62 Prozent des Datenverkehrs eine reduzierte Sicherheit fest und stießen bei 58 Prozent sogar auf ernst zu nehmende Schwachstellen. Die Studie endet mit dem Satz: „Wir hoffen, dass wir die Hersteller ermuntern können, ihre Sicherheitsstandards zu verbessern, und die Security-Community dazu anregen, Alternativen zur HTTPS-Interception zu diskutieren.“ Wie der Fall Kaspersky zeigt, ist das Kernproblem nach wie vor aktuell.

Datenumleitung durch Virenschutz

Ist Kaspersky installiert, fließt der Datenverkehr des Browsers nicht mehr direkt zum Webserver, sondern durch die Schutzsoftware.



Moskau meldet sich zu Wort

Kaspersky hat die von uns entdeckte Sicherheitslücke zeitig geschlossen. Nach unserer Berichterstattung war es um das Unternehmen erstaunlich ruhig – es antwortete zwar zügig auf Anfragen von Journalisten, eine umfangreiche öffentliche Stellungnahme folgte jedoch erst nach rund fünf Tagen (siehe ct.de/y2q8). Der Hersteller schildert darin das Problem und mögliche Angriffsvektoren. Gleichzeitig kritisiert das Unternehmen, dass das Problem von den Medien aufgebauscht worden sei. Last, but not least meldete sich auch noch Eugene Kaspersky persönlich zu Wort: Auf Twitter bedankte er sich dafür, dass wir die „Responsible Disclosure“ eingehalten haben, also dem Unternehmen vor der Veröffentlichung ausreichend Zeit einräumten, das Datenleck zu schließen.

Der Fall in aller Kürze

Falls Sie unser „c’t deckt auf“ in Ausgabe 18/2019 verpasst haben, gibt es hier noch mal die Ultrakurzfassung. Durch einen Zufall haben wir entdeckt, dass die Kaspersky-Virenschutzsoftware eine eindeutig zuordenbare ID in sämtliche Websites einschleust, die der Nutzer aufruft (siehe ct.de/y2q8). Das Problem dabei ist, dass diese ID von den Websites ausgelesen werden kann und sich daher für umfassendes Tracking eignet. Anhand der ID können Websites einzelne Besucher wiedererkennen – im Prinzip wie bei einem Cookie, nur viel schlimmer: Denn anders als Cookies funktioniert die Kaspersky-ID auch über Browsergrenzen hinweg und sogar im Inkognitomodus des Browsers.

Kaspersky hatte also eine mächtige Tracking-Funktion geschaffen. Zum Tracking war nur etwas JavaScript-Code nötig, wie eine von c’t erstellte Demoseite beweist. Verteilt man den JavaScript-Code über mehrere Websites, etwa über ein Anzeigennetzwerk, ist gar browserübergreifendes Tracking möglich. Dass dieses Problem in einer Schutzsoftware steckte, macht den Fall besonders tragisch – denn Kaspersky wirbt nicht nur mit Sicherheit, sondern auch mit einer Verbesserung der Privatsphäre. Das Gegenteil war der Fall.

Wir meldeten unseren Fund vor der Veröffentlichung an Kaspersky und räumten dem Unternehmen genügend Zeit ein, um die Schwachstelle zu beseitigen. Die Lücke trägt die Schwachstellen-Identifikationsnummer CVE-2019-8286. Kaspersky hat die Anfälligkeit im Juni mit dem „Patch F“ entschärft, der seit dem 7. Juni automatisch verteilt wird. Seitdem schleust die Schutzsoftware nicht länger eine individuelle ID ein, sondern eine statische, die bei allen Nutzern gleich ist. *(rei@ct.de)* c't

***Weitere Hintergründe:* ct.de/y2q8**