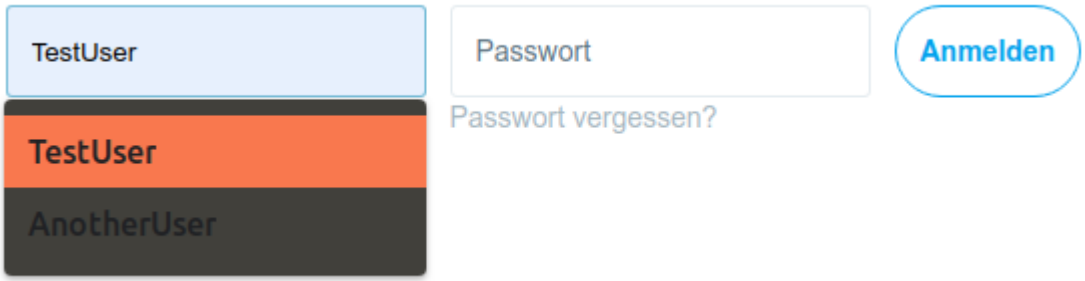
HTML Input-Field: Remote cache stealing

Inhalt:

Prinzip der Schwachstelle

Bei dieser Schwachstelle handelt es sich um eine, welche das Stehlen/Ermitteln von gespeicherten Nutzereingaben aus HTML-Input-Feldern ermöglicht. Hierzu wird der Nutzer angewiesen einige wenige Tastenschläge auf seiner Tastatur zu machen, welche beispielsweise als Teil eines Captcha oder Spiel getarnt werden können.

Beim Surfen im Internet mit einem Browser werden vom Nutzer immer wieder HTML-Formulare (normale Eingaben auf Webseiten) genutzt. Diese können Beispielsweise die Eingabe von Daten wie Telefonnummern, User-Namen oder Suchanfragen sein. Dementsprechend sind HTML-Input-Felder ein fundamentaler Bestandteil des WWW und kommen auf jeder bekannten Website wie beispielsweise Google, Twitter, Youtube, Amazon und vielen mehr in Massenhafter Form zum Einsatz. Damit der Umgang mit diesen Eingaben leichter und bequemer wird speichert der Browser jene Eingaben ab und bietet sie dem Nutzer beim Ausfüllen des selben Feldes erneut zum automatischen Ausfüllen an.



Diese Besonderheit ermöglicht es schnell und bequem Logins und Eingaben durchzuführen. Der Nutzer kann nun mithilfe der Pfeil-Tasten „Hoch“ sowie „Runter“ die einzelnen bereits gespeicherten Eingaben auswählen und mit der Enter-Taste bestätigen um diese automatisch in des entsprechende Eingabe-Feld (HTML-Input-Field) einzufügen.

Um diese gespeicherten Daten stehlen zu können sind nun 2 Faktoren wichtig.

Erstens: Der Angreifer muss es schaffen, dass sein Opfer die zu stehlenden Eingaben auch auf einer seiner eigenen und manipulierten Webseiten angezeigt bekommt. Das ist notwendig um Eingaben, welche auf externen Seiten wie beispielsweise Twitter oder Youtube gemacht worden stehlen zu können ohne wirklich diese Seiten an zu greifen oder Schwachstellen bei diesen kennen zu müssen.

Zweitens: Das Auswahlfeld betreffend der automatischen Eingabe darf während der Eingabe des Opfers nicht für das Opfer sichtbar sein. Könnte man das Opfer beispielsweise dazu bringen einfach nur einmal die Pfeil-Taste „Hoch“ oder „Runter“ zu drücken, gefolgt von der Enter-Taste, so könnte man eine beliebige zwischengespeicherte Eingabe wie z.B. die persönliche E-Mail-Adresse oder Telefonnummer des Opfers ermitteln. Da das Opfer aber auf diese Auswahl hingewiesen wird und explizit eine vorgeschlagene Auswahl treffen kann aber nicht muss wird es sensible Daten niemals auf diesem Weg auf unbekannten Adressen/Webseiten eingeben.

Bei dieser Schwachstelle werden beide Faktoren ermöglicht und ausgenutzt, wobei der Angreifer zunächst das entsprechende Input-Feld mit dem dazugehörigen Feldnamen nachstellt und den Nutzer dann Aktionen durchführen lässt die einen völlig anderen Sinn aufzeigen sowie die optische Komponente der automatischen Eingabe vollkommen unsichtbar macht.

Dementsprechend wurden zur Veranschaulichung 2 Beispielangriffe programmiert wobei einer auf sehr einfache Weise den Nutzer auffordert seine Eingaben aufgrund eines Captcha zu machen und der zweite die notwendigen Eingaben in einem kleinen Spiel versteckt.

Captcha

Spiel

Betroffene Browser

Betroffen von dieser Schwachstelle sind alle Browser, welche das Verstecken der automatischen Eingabe ermöglichen sowie mit Standard-Einstellungen das Speichern von Eingaben überhaupt ermöglichen. Hierzu zählen sich nach eigenen Tests der Google Chrome Browser sowie Mozilla Firefox und der Opera Browser, wobei neben diesen Browsern nur noch der Microsoft Edge getestet wurde, welcher allerdings das Speichern von Nutzereingaben nicht standardmäßig ermöglicht. Die Schwachstelle ist dementsprechend auf den gängigen und beliebtesten Browsern des WWW ausnutzbar.

Unter den folgenden Eigenschaften ist die Schwachstelle ausnutzbar (Mac OS X wurde nicht getestet).

Linux (KDE & Gnome):
Google Chrome *ohne Einschränkung*
Mozilla Firefox *nur im Vollbild (F11)*
Opera *ohne Einschränkung*

Windows:
Google Chrome *ohne Einschränkung*
Mozilla Firefox *als maximiertes Fenster und im Vollbild*
Opera *ohne Einschränkung*

Ausnutzung der Schwachstelle

Um die Schwachstelle auszunutzen wird wie folgt vorgegangen:

1. Ermittlung des Namens der zu stehlenden Eingabe.

Um den Name des entsprechenden Input-Feldes zu ermitteln besucht man einfach das entsprechende Eingabefeld auf der Webseite. Möchte man beispielsweise den Feldname des Nutzernamens vom Login bei Twitter ermitteln, dann besucht man einfach den Twitter-Login und kopiert sich dort den Feld-Namen aus dem Seiten-Quelltext.

2. Unsichtbar machen des nachgestellten Eingabefeldes.

Im nächsten Schritt muss das Eingabefeld erstellt und unsichtbar gemacht werden. Hierzu wird die Höhe des Eingabefeldes auf mindestens die doppelte Höhe der Desktop-Auflösung sowie die Position des Feldes im Top-Bereich auf mindestens 100 Pixel im negativen Bereich angesetzt. Danach sollte es noch in den Hintergrund gerückt werden und möglichst eine Transparenz von 100% erhalten. Das fertige CSS könnte dann wie folgt aussehen:
width: 100px; height: 5120px; position: fixed; left: 0px; top: -100px; opacity: 0; z-index: -1

3. Die bisherige Arbeit sinnvoll tarnen

Damit der Nutzer die nötigen Eingaben ohne bedenken tätigt ist es notwendig dem geschehen ein komplett neues Aussehen sowie einen völlig neuen Sinn zu verleihen! Dafür kann man beispielsweise ein Script erstellen, welches sich als Captcha oder Spiel tarnt.

Das Beispiel-Script für den Captcha liegt bei als arrows.html.
Passwort: ic9Ak02.1-912c89
Link: https://mega.nz/#!EAgnilCa!yeKwVdgi95bH73Sj2--7wvRDJLVwoB3vXcvASeYtYm0

Das Beispiel-Script für das Spiel liegt bei als arrows-game.html.
Passwort: c89Akc-uaok9801.
Link: https://mega.nz/#!pBxRXAib!xuNBwHVDu-1zkQG0FW2g4gj2J5zhJFgtHX9ehJLM8p8

Das Beispiel-Video vom Spiel-Script liegt bei als game-vid.mp4.
Passwort: juCug-uF!2189014
Link: https://mega.nz/#!0VwFWYTK!b3GcroYQxKMU6bv_a6_kOX8htw9z_W4LTiCqsksPctM