Altes Forum - Neue Software - Eure Meinung!

Beitrag von „AnitaE.“ vom 22. Dezember 2015, 13:31

Hier wurde eine These aufgestellt und eine Unterstellung ausgeübt, über User ,die aus Sicherheitshaltung heraus JavaScript deaktiviert haben. Ich habe ein anderes Szenario hinzugefügt. Nichts weiter. Dies war genau so legitim.

Wäre interessant ob Du / ihr dann auch alle offiziellen Ratgeber (einschlägige Onlinemagazine usw) auch mit derartigen boshaften Andeutungen bezeichnen würdet .
Wenn also Du mich fragst, was ich für eine Antwort erwartet hätte, frage ich dich was Du für eine Antwort erwartet hast ? Einseitig zu argumentieren ist nun mal nicht OK. Es hat immer alles zwei Seiten.

Das hier im Forum keine Kritik erwünscht ist, habe ich nun mittlerweile auch begriffen. Man kann ja lesen wie es Usern hier ergeht die vor mir ebenfalls Kritik an manchen Sachen äusserten.
Ich werde aber nicht wie ein dressiertes Hündchen hier zu allem Applaus klatschen, wenn ich denke dass es falsch ist.

Es stimmt zwar, das manche Foren (nicht alle und nicht viele) Java nutzen , es stimmt jedoch nicht dass diese Softwareumgebung unbedenklich im Netz ist. Egal ob auf Foren oder sonstigen Seiten.

Ihr seid recht dünnhäutig was eure Plattform angeht und man fragt sich immer wieder warum.... es muss hier wohl einiges geschenen sein in den letzten Jahren oder wie lange auch immer dieses Forum existiert.
Ich vermisse auch so etwas wie eine gewisse Souverenität hier. Aber das ist nicht unser Problem.

Der Versuch , meine Gegenargumentation (von deiner Unterstellung vielen Usern im Internet) in eine Richtung zu lenken die angeblich direkt gegen das Forum hier gehen würde, greift nicht. Deswegen schrieb ich , wie Du, dass Du dich nicht persönlich angesprochen fühlen sollst.

Achja,.... bin gespannt ob Du auch das BSI als Deppen hinstellst ... denn auch diese haben sich u.A. mit der Problemkatik befasst...

Zitat:

> Zitat
>
> Gefahren und Risiken im Umgang mit JavaScript/JScript
>
> Es sind verschiedene sicherheitsrelevante Schwachstellen bekannt, durch die bei Ausführung von JScript/JavaScript-Code Schäden auf dem Anwenderrechner entstehen können.
>
> Relativ harmlos ist die missbräuchliche Verwendung einiger JScript/JavaScript-Funktionen,
> durch die "nur" der normale Rechnerbetrieb des Anwenders gestört wird.
> Es ist beispielsweise möglich, unzählig oft weitere Fenster zu öffnen
> und damit eine Art Denial-Of-Service-Angriff
> auf den Anwenderrechner zu verursachen. Die geöffneten Fenster binden
> Rechner-Ressourcen und zwingen den Anwender, diese Fenster entweder zu
> schließen oder den Rechner neu zu starten.
>
> Auch in die harmlosere Kategorie fällt das Problem, dass sich
> Programmierfehler erst zur Laufzeit auf dem Rechner des Anwenders
> auswirken. Da nicht sichergestellt ist, dass der Programmierer sein
> Skript mit allen Browservarianten getestet hat, unterstützt der verwendete Browser möglicherweise nicht alle Konstrukte und kann abstürzen.
>
> Doch es gibt auch kritische Schwachstellen. Ein solches Sicherheitsrisiko kann vom JScript/JavaScript-Interpreter
> selbst ausgehen. Ist dieser fehlerhaft programmiert, entstehen
> Sicherheitslücken, die Angreifer ausnutzen können. Im schlimmsten Fall
> erhält ein Außenstehender vollständigen Zugriff auf den Rechner.

Ebenfalls kritisch sind einige Möglichkeiten, mit JScript/JavaScript-Elementen den Anwender zu täuschen. So können ganze Eingabefenster von vertrauenswürdigen Webseiten
simuliert werden, wodurch beispielsweise Benutzernamen, Passwort oder andere sensible Daten wie Kreditkarteninformationen abgefangen und übertragen werden können.

Auch mit JavaScript/JScript möglich ist, die Anwender über das Ziel von Links zu täuschen. Über spezielle Funktionen kann die Statusleiste des Web-Browsers verändert werden, so dass die dort angezeigte Adresse nicht der der tatsächlich verlinkten Webseite entspricht.

Relativ gefährlich ist es, wenn sich der Anwender aufgrund eingesetzter Schutzprogramme in falscher Sicherheit wiegt. Es gibt beispielsweise Filter-Programme, die an Firewall-Systemen eingesetzt werden können. Diese Filter-Programme haben spezielle Funktionen, um Skriptcode auf bösartige Funktionen zu untersuchen und bei Bedarf diesen heraus zu filtern. Unter JavaScript/JScript können jedoch durch Verschleierung entsprechender HTML-Tags mit JavaScript/JScript-Funktionen beispielsweise Java-Applets auf dem lokalen Rechner des Anwenders ausgeführt werden, auch wenn die Firewall die Applets eigentlich herausfiltern sollte.

Nachfolgend ein Beispiel für die Verschleierung von HTML-Tags:

```
document.write('<APP');

document.write('LET\n');

document.write('CODE=client.Main.class \n>');

document.write('CODEBASE=base \>');
```

> document.write('ARCHIVE=clientapplet.jar \>');
>
> document.write('</APPLET>') ;
>
> Auch die Verwendung von Signaturen birgt Risiken, wenn ihre Sicherheitsfunktionen nicht richtig eingeschätzt werden. Der Anwender erhält zwar die Gewissheit, dass die JavaScript/JScript-Dateien unverändert sind. Bezüglich der Vertrauenswürdigkeit und der Kompetenz des Entwicklers werden jedoch durch die Signatur keine Anhaltspunkte gegeben. Auch wird keine Aussage zum Funktionsumfang der JavaScript/JScript-Dateien und der davon ausgehenden Gefahren getroffen.
>
> Alles anzeigen

.... Quelle: https://www.bsi.bund.de/DE/The...initionen_javascript.html

oder hier..._

> Zitat
>
> Warum Java gefährlich ist
> Grundsätzlich hat jedes größere Softwareprojekt das Problem der Fehleranfälligkeit. Noch schwieriger und komplexer wird die Situation, wenn eine Software im Zusammenspiel mit einer anderen Software oder gar dem Betriebssystem agieren muss - Systemverwalter und Sicherheitsbeauftragte können ein Lied davon singen. Ein weiteres großes Problem bei Java: Hier werden Programme auf dem PC ausgeführt. Um Programme auf einem Rechner (ganz gleich mit welchem Betriebssystem er betrieben wird) auszuführen, benötigen diese Ressourcen des Betriebssystems und in einigen Fällen auch Zugriff auf die dort gespeicherten Dateien. Vielfach werden diese Aktionen dabei mit all den Rechten des jeweiligen Nutzers ausgeführt - arbeitet dieser mit den Zugriffsrechten eines Administrators, so sind auch gefährliche Zugriffe und Änderungen möglich. Obwohl es schon seit Windows Vista für einen Standardnutzer nicht mehr wie unter Windows XP nötig ist, mit den Rechten eines Administrators zu arbeiten, ist dies leider allzu häufig

> noch die Regel.
>
> Nun haben die Java-Entwickler sich aber grundsätzlich ein gutes Konzept einfallen lassen, indem sie die Java-Programme in einer virtuellen Maschine ausführen und deren Zugriffe auf das System damit gehörig abblocken sollten. Die Praxis zeigt leider, dass dies nicht der Fall ist und dass hier auch viele Einflüsse von anderen Programmen einwirken können, die sich dann als veritable Sicherheitslücken entpuppen (in der Computerwissenschaft als "side effects" bezeichnet und häufig mit "Seiteneffekt" nicht zutreffend übersetzt). Fehler in der Implementierung und bei der Umsetzung neuer Funktionen und Bibliotheken tun ein Übriges dazu, dass immer wieder Sicherheitslücken entstehen. Diese müssen dann vom Anbieter - in diesem Fall Oracle - durch entsprechende Patches und Upgrades wieder beseitigt werden.
>
> Grundsätzlich gilt auch hier: Anwender sollte keine Software aus unbekannten und/oder potenziell unsicheren Quellen auf ihren Systemen ausführen. Genau das tun sie häufig aber, wenn sie eine Web-Seite aufrufen, die ein Java-Applet auf dem lokalen PC startet!
>
> Alles anzeigen

Quelle: http://www.computerwoche.de/a/...-risiko-darstellt,2538751

oder hier.....

> Zitat
>
> Durch einen simplen JavaScript-Trick lassen sich Websites so manipulieren, dass User freiwillig ihre Passwörter eingeben. Antiviren-Programme oder andere Vorsichtsmaßnahmen helfen nicht, der Falle zu entgehen.
>
> Bekanntlich sind Versuche, sensible Daten von Internet-Nutzern abzugreifen, immer dann am erfolgreichsten, wenn sie so wie ein bekannter Vorgang aussehen. So gab es beispielsweise eine Zeit lang

> einen regelrechten Boom von Websites, die offiziellen Online-Banking-Seiten der großen Kreditinstitute zum Verwechseln ähnlich sahen, in Wirklichkeit aber nur die eingegebenen Daten abgriffen, statt Zugriff auf das Konto zu gewähren.
>
> Ein jetzt publik gewordener Einsatzbereich der JavaScript-Methode event.preventDefault() schlägt in genau dieselbe Kerbe: Das Document Object Model (DOM) verhindert, dass eine eigentlich als Standard vorgesehene Aktion im Browser ausgeführt wird. Dies wiederum lässt sich nutzen, um dem Nutzer vorzutäuschen, dass die gewünschte Aktion durchgeführt wird, während in Realität ein völlig anderer Aktionsprozess in Gang gesetzt wurde.....
>
> Alles anzeigen

...mehr unter der Quelle: http://www.chip.de/news/JavaSc...hr-Passwort_59217242.html

usw usw usw...

Es ist also fahrlässig , den Usern hier im Forum Javascript als harmloses Umgebung darzustellen.

Da brauche ich auch keinen Freund von jemanden hier, der mir etwas anderes erzählen möchte.

lG.