Präparierte Webseiten und Links — Version 2022/I

Egg, 7. Februar 2022: *Eine kurze Mail, dann ein Telefonat. Im Oktober 2021 tritt eine Firma an die Archivista GmbH heran. Sie hätten für einen Kunden ein Security-Audit vorgenommen. WebDMS und WebAdmin seien über präparierte Webseiten bzw. Links angreifbar. Der Kunde könne nicht genannt werden, der Angriff sei jedoch dokumentiert und das Dokument könne über einen Link (mit Passwort) abgerufen werden. Was darauf folgt, ist eine längere Odysee. Von dieser Reise handelt dieser Blog.*

Worum geht es bei Cross-Site-Scripting?

Alle moderneren Web-Applikationen arbeiten mit zwei Schichten. Da ist einmal die Präsentation der Inhalte im Web-Browser bei den End-Geräten. Auf der anderen Seite stehen die Server (z.B. die ArchivistaBox), auf denen die Daten verarbeitet werden. Bei der Darstellung der Inhalte kommt dabei meistens JavaScript zum Einsatz. Damit können beim User-Interface (z.B. WebDMS) Informationen sehr flexibel (quasi on-the-fly) dargestellt und verändert werden, ohne dass jeweils sämtliche Daten bei jedem Aufruf zwischen Client (WebDMS) und dem Server (ArchivistaBox) neu übermittelt werden müssen.
Nachteilig im Sinne der Sicherheit von JavaScript ist, dass die entsprechenden Programme direkt im Web-Browser abgearbeitet werden.Dies deshalb, weil damit die Hoheit an einen externen Partner (Web-Browser) delegiert wird. Und auch wenn JavaScript-Programme unleserlich dargestellt werden, die Sourcen liegen quelloffen vor. Der Code kann dabei ohne grossen Aufwand im Web-Browser eingesehen und (deutlich “unangenehmer”) auch verändert werden. Der aufgerufene Server kann dabei nicht feststellen, ob sein externer Partner in guter oder schlechter Mission, d.h. mit geänderten Quellen arbeitet.
Die einzige Kontrolle des Servers über seinen Klienten (Web-Browser) besteht darin, dass die Sourcen beim Aufruf vom Server geladen werden. Ob das ausgelieferte Programm in korrekter Weise abgearbeitet wird, darüber entscheidet nicht der Server (ArchivistaBox), sondern der jeweilige Web-Browser. Beim sogenannten Cross-Site-Scripting, auch bekannt unter XSS-Attacken, passiert genau dies. Der JavaScript-Code wird im Web-Browser kompromittiert.

Reflektierte Attacken

Eine Unterart von solchen Attacken liegt bei den reflektierten Angriffen vor. Der “Angriff” erfolgt dabei über einen externen Link, der z.B. über präparierte PDF-Dokumenten oder Mails ausgeliefert wird. Klicken ahnungslose Anwender/innen

darauf und geben dabei unbedarft Benutzername und Passwort ein, so “fängt” der externe Server die Anmeldedaten ab und kann danach mit den so gewonnen Daten problemlos auf die gewünschten Web-Applikation zugreifen.
Wichtig dabei zu wissen ist, die Anmeldedaten werden dabei auf einem externen Rechner eingegeben. Wer bei Eingabe von Daten überprüft, ob in der Navigationsleiste des Web-Browsers die Adresse des gewünschten Servers steht, kann in dieser Form nicht “übertölpelt” werden. Eine Unterart des reflektierten Angriffs besteht darin, dass zwar die gewünschte Website (z.B. ArchivistaBox) aufgerufen wird, dass dabei jedoch direkt JavaScript-Code übermittelt wird. Anbei ein Beispiel:

```
http://192.168.0.99/perl/avclient/index.pl?target=_top&host=
localhost"><script>alert("Hacking 1")</script>&db=
"><script>alert("Hacking 2")</script>&uid=
"><script>alert("Hacking 3")</script>
&uid=Admin&db=archivista&pwd=xxxx
```

Um das Beispiel ausführen zu können, muss bei der IP-Adresse 192.168.0.99 die IP-Adresse einer ArchivistaBox eingegeben werden. Abgesehen davon, dass der obige Link relativ lang ist, sollte einem im übrigen nicht primär “Hacking X” stutzig machen (ein “bösartiger” Hacker würde einem nicht mit ‘Hacking X’ begrüssen, sondern das Fragment ‘<script>…</script>’, da darin ja ganz bewusst ein externer Code “eingeschleust” werden soll.
Im Prinzip sollte jeder Server überprüfen, ob derartige Codefragmente an ihn übermittelt werden. Korrekt geparst würde das Fragment ‘<script>…</script>’ dann einfach als Text im entsprechenden Feld der Anwender/innen ausgegeben, d.h. der “eingeschleuste” Code wird als Text und nicht als Script verarbeitet. Diese Kontrolle erfolgte in früheren Version von WebDMS leider mangelhaft. Zwar war es nicht möglich, beim Arbeiten in WebDMS nach dem Anmelden derartige Code-Fragmente abzusetzen, doch beim Login-Formular erfolgte leider keine solche Kontrolle.

Ist meine ArchivistaBox davon betroffen?
Grundsätzlich sind alle bisherigen ArchivistaBoxen (Versionen vor 2022/I) davon betroffen, da beim Login-Formular das Fragment ‘<script>…</script>’ nicht zwingend in Text umgewandelt wurde. Auch wenn entsprechende Angriffe ohne die Mithilfe unbedarfter Dritter nicht möglich sind, so soll die Maschine (hier ArchivistaBox) natürlich trotzdem alle möglichen Checks durchführen, damit solche Angriffe möglichst vermieden werden können. Mit ‘möglichst vermieden’

sei zum Ausdruck gebracht, dass nur das vermieden werden kann, was auch bekannt ist.

Was gilt es zu tun?

Die Antwort lautet: Update “bestellen” und über WebConfig einspielen. “Bestellen” bedeutet, dass Kunden mit Wartungsvertrag jederzeit kostenfrei eine aktualisierte Version erhalten, sofern sie dies wünschen (Mitteilung per Telefon, Mail oder in Briefform genügt).

An dieser Stelle darf die Frage aufgeworfen werden, warum nicht einfach alle ArchivistaBox-Systeme automatisch aktualisiert werden. Die Antwort lautet, dass dies im Konzept der ArchivistaBox so nicht vorgesehen ist, da solche Updates einen automatisierten Zugriff auf die ArchivistaBox erfordern würden. So praktisch solche Updates auch sein mögen, Systeme, welche jederzeit automatisiert aktualisiert werden können, sind in weit grösserem Masse “angreifbar”, als wenn Updates durch Menschen “angestossen” werden.

Wie bereits oben ausgeführt, reflektierte Cross-Site-Scripting-Angriffe erfordern immer ein (mehr oder minder) naives Mittun der Benutzer/innen. Wer unbedarft auf Links klickt bzw. sich bei Eingabe von Benutzername und Passwort nicht darum schert, wem diese Informationen anvertraut werden, muss sich nicht wundern, wenn ein Takeover erfolgt. Die Übernahme des Systems erfolgt ja erst dadurch, dass auf einer Drittseite die Anmeldedaten “preisgegeben” werden. Die gleiche Situation liegt vor, wenn die gleichen Benutzerdaten in sämtlichen möglichen und unmöglichen Situation verwendet werden.

Selbst das beste Passwort nützt herzlich wenig, wenn es tausendfach bei sämtlichen Systemen zur Anwendung gelangt. Ebenso sollten natürlich die Standardpasswörter gerade nicht zur Anwendung gelangen. In diesem Sinne stellen reflektierte Cross-Site-Scripting-Angriffe ein eher kleineres Problem dar. Dennoch soll und darf darüber gesprochen werden.

Wenn ein unbekannter Dritter zum Download einlädt

Zurück zur Frage, warum eingangs von einer Odysee die Rede ist. Der erste Anruf erfolgte im Oktober 2021. Eine Person gibt sich als Security-Forscher aus und verweist auf eine Firmen-Webseite mit Sitz in Deutschland. Mit bestimmter Tonalität möchte der Anrufer, dass die Firma Archivista ein Dokument (passwortgeschützt) ab einer Webseite herunterlädt. Die Sicherheitslücke betreffe einen Kunden, dies habe ein Audit ergeben.

Da weder der Name des Kunden genannt wird, da keine Geschäftsbeziehung zum Anrufer besteht, da der Anrufer ab einer unbekannten und nicht verifizierbaren Mobilfunknummer aus einem Drittland anruft, da ein Blick auf die angegebene Firmen-Homepage nicht im engeren Sinne eine Firma mit hunderten von Mitarbeitern (wie im Telefonat erwähnt) vermuten lässt, wird der Anrufer eingeladen, die “angebliche” Lücke per Mail zuzusenden. Dies wiederum möchte der Anrufende um keinen Fall, dies sei unsicher.

Der Einwand, dass bei Open Source gut und gerne auch öffentlich über Sicherheitsprobleme diskutiert werden dürfe bzw. müsse, überzeugt den Anrufenden nicht. Er beharrt darauf, dass die Lücke nur vertraulich übermittelt werden könne. Selbst der Vorschlag, das Dokument in Briefform (mit/ohne USB-Stick) zuzustellen, findet keine Gnade. Irgendwann verweist die Firma Archivista auf die **Meldestelle für Cyberattacken des Bundes.** Der Anrufende betont, dass sollte das Dokument nicht heruntergeladen werden, eine solche Meldung erfolgen würde und dass die Sicherheitsmeldung danach öffentlich publiziert würde.

Nationale Meldestelle nur per Web-Formular erreichbar
Anfangs Februar erfolgt ein Anruf eines Herrn Knoepfel, der beim nationalen Zentrum für Cybersicherheit arbeite. Dieser bittet um einen Rückruf. Leider gelingt es nicht, den Herrn Knoepfel zu erreichen. Weiter ist es relativ schwierig zu überprüfen, ob Herr Knoepfel wirklich Mitarbeiter der besagten Bundesstelle ist, denn auf der Homepage der **Meldestelle für Cyberattacken des Bundes** gibt es einzig ein Web-Formular, um Meldungen zu erfassen. Wer nach einer Telefonnummer sucht, findet eine solche nicht.
Wer erfahren möchte, wer bei besagtem Amt arbeitet, müsste sich über Linkedin.com anmelden, der Link **https://www.linkedin.com/company/ncsc-ch** befindet sich ganz unten auf der entsprechenden NCSC-Homepage. Da Linkedin.com vor einigen Jahren dadurch aufgefallen war, dass über 100 Millionen Passwörter abgekupfert wurden, wird auf das Eröffnen eines Kontos verzichtet. Herr Knoepfel meldet sich einige Tage später per Mail. Es sei eine Sicherheitslücke bei der Meldestelle eingegangen, die Forscher würden die Lücke wohl in naher Zukunft veröffentlichen.
In der Mail findet sich ein Verweis auf eine weitere Mail, worin behauptet wird, die Firma Archivista habe am 17. Dezember 2021 die Sicherheitslücke erhalten. Da dies mit den Logs bzw. dem Spam-Ordner nicht verifiziert werden kann, erfolgt abermals ein Anruf bei besagter Nummer. Erneut lädt der Anrufbeantworter zur Hinterlegung einer Nachricht ein.
NSCS-Telefonnummer “nur” bei help.ch
Eine Web-Recherche fördert die Telefonnummer der NCSC bei **help.ch** zutage. Beim Anruf unter dieser Nummer nimmt eine Frau Minder ab. Sie sei Kommunikationsbeauftragte des Finanzdepartementes. Die Nachfrage, ob ein Herr Knoepfel bei der NSCS arbeite, beantwortet sie zunächst damit, es sei kein Herr Knoepfel bekannt. Erst auf den Einwand, dass, wenn es den Herrn Knoepfel nicht gebe, es einer Meldung bei der NCSC bedürfe, wonach sich jemand als NSCS-Mitarbeiter mit dem Namen Knoepfel ausgebe, fragt Frau Minder nach, wie der Herr Knoepfel geschrieben werde. Unter ‘Knöpfel’ gebe es niemanden, aber unter Knoepfel, doch, dieser existiere.
Die telefonisch angegebene Nummer stimmt mit der Nummer in der Mail überein. Erneut wird versucht, Herrn Knoepfel telefonisch zu kontaktieren. Da dies nicht gelingt, wird eine länger Mail mit Schilderung der Sachlage verfasst. Ehe die Mail verschickt wird, klingelt das Telefon. Herr Knoepfel ruft zurück und

meint, ich hätte mehrfach angerufen.
Es wird die Situation geschildert, wonach der Firma Archivista bis heute keine Sicherheitsmeldung vorliege, obwohl darum per Mail mehrfach gebeten wurde. Herr Knoepfel bestätigt, dass eine Meldung bei der NCSC eingegangen sei. Auf Nachfrage hin wird die Sicherheitsmeldung per Mail durch Herrn Knoepfel zugestellt.
Es handelt sich um ein achtseitiges PDF-Dokument, welches eine reflektierte Cross-Siting-Script-Lücke (wie oben beschrieben) in der Anmeldemaske protokolliert. Ebenfalls wird recht gut beschrieben, wie das Ausführen von JavaScript über manipulierte Links vermieden werden kann bzw. in unserer Applikation verifiziert werden kann, dass beim Anmelden die Werte nicht korrekt verarbeitet werden.

Der “ominöse” Kunde mit Version 5.2

Stutzig in besagtem Dokument macht ein Screenshot, welcher Archivista WebClient mit Version 5.2 ausweist. Die Version 5.2 datiert aus den Jahren 2005 bis 2007. Folglich wird klar, dass der “mysteriöse” Kunde keine je bei Archivista erworbene ArchivistaBox betreibt. Vielmehr kommt eine Open Source Version zum Einsatz, die mittlerweile irgendwo zwischen 15 und 20 Jahre auf dem Buckel hat.
Selbstverständlich beinhaltet das Recht bei der GPLv2-Lizenz, eine jede Software in beliebiger Form einzusetzen. Allerdings erstaunt es doch sehr, dass sich eine Firma ein Security-Audit (solche Überprüfungen sind nicht ganz “günstig”) leistet, um eine ca. 15 bis 20 jährige Software (hier Archivista WebClient, aktuell WebDMS) überprüfen zu lassen.
Nun könnte die betreffende Firma, welche die ArchivistaBox über offensichtlich diesen Zeitpunkt ohne Wartung und Support im Einsatz stehen hat, das Problem ja selber angehen — immerhin liegen die Sourcen ja vor. Genau dies allerdings erfolgte nicht. Stattdessen wurde die Audit-Firma beauftragt, die Lücke mit Klassifikation “vertraulich” zu dokumentieren.
Dies führte dazu, dass besagte Audit-Firma das Dokument um keinen Preis per Mail oder in Briefform zustellen wollte. Immerhin, nach vielen Um- und Irrwegen über die **Meldestelle für Cyberattacken des Bundes** erhielt die Firma Archivista GmbH endlich Einblick in Problematik — und konnte schliesslich handeln.

Die Moral von der Geschichte…

Angriffe mit reflektiertem Cross-Siting-Sripting sind letztlich so effektiv wie die Nutzer/innen unbedarft auf irgendwelche Links klicken bzw. ihre Kennwörter ungeniert eingeben. Über solche Lücken soll und darf diskutiert werden. Sie dürfen bzw. sollten aber nicht überbewertet werden.
Ganz unabhängig davon bedeutet Open Source Offenheit auf der Basis von Vertrauen und Fairness. Wo die Sourcen offen gelegt sind, bringt “Geheimhaltung” wenig bis nichts. Allfällige Schwachstellen sind ohnehin öffentlich im Code einsehbar. Und darum dürfen alle jederzeit allfällige Schwachstellen gerne kommunizieren. Im Sinne der Offenheit werden solche Meldungen aber auch in Zukunft vorzugsweise **per Mail** entgegen genommen.


Facebook

Twitter