Προγραμματισμός Διαδικτύου

Δρ. Μηνάς Δασυγένης
mdasygenis@uowm.gr

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Έγχυση SQL (Injection)

Εισαγωγή, επίδειξη και επισκόπηση

Τι είναι το SQL Injection;

- Εισαγωγή των δηλώσεων SQL σε εισόδους εφαρμογών για την καταστροφή, εκμετάλλευση ή άλλη βλάβη μιας βάσης δεδομένων εφαρμογών.

- Συνήθως γίνεται απευθείας μέσω διαδικτυακής φόρμας , αλλά μπορεί να κατευθύνεται μέσω hacking διευθύνσεων URL, να ζητάει hacking χρησιμοποιώντας εργαλεία εντοπισμού σφαλμάτων ή χρησιμοποιώντας bots που εξομοιώνουν προγράμματα περιήγησης και χειρίζονται διαδικτυακά αιτήματα.

Τι κάνει?

- Τα πιο απλά χάκς (hacks) χρησιμοποιούνται για την παράκαμψη του ελέγχου ταυτότητας επιπέδου βάσης δεδομένων.
- Οι επιθέσεις Bot σε δημόσιες αντιμέτωπες ιστοσελίδες παρουσιάζονται συχνά σε δύο στάδια:
  - Στάδιο 1: εξαναγκασμός SQL στην εφαρμογή για την εμφάνιση μεταδεδομένων βάσης δεδομένων, όπως δομές πίνακα και στήλης.
  - Στάδιο 2: Χρησιμοποιήστε τα μεταδεδομένα για να επιτεθείτε στη βάση δεδομένων.

    Οι άνθρωποι συχνά εφαρμόζουν την κατάσταση 2 μετά από χτύπημα από ένα bot.
- Το πιο κοινό αποτέλεσμα είναι η εισαγωγή ετικετών HTML σε πίνακες βάσεων δεδομένων:
  - Το κατεστραμμένο περιεχόμενο εμφανίζεται στη διαδικτυακή εφαρμογή.
  - Οι κακόβουλες ετικέτες έχουν ως αποτέλεσμα τη μεταφόρτωση κακόβουλου λογισμικού σε οποιονδήποτε επισκέπτεται έναν ιστότοπο χρησιμοποιώντας ενσωματωμένο JavaScript, iFrames κ.λπ., ενσωματωμένο σε περιεχόμενο βάσης δεδομένων.

Γενικό πρόβλημα: Επικύρωση εισόδου

- Πολύ παλαιό πρόβλημα:

- Τα HTTP get και post δεν είναι ασφαλείς μηχανισμοί αιτήματος και δεν έχουν σχεδιαστεί για να είναι ασφαλείς.
- Το Perl CGI ήταν ευάλωτο σε ενσωματωμένα κομμάτια Perl.
- Όλες οι μορφές υπερχείλισης ρυθμιστών (buffer) είναι αποτυχίες για την επικύρωση της εισόδου του χρήστη.

- Πολύ συνηθισμένο πρόβλημα:

- Τα διαδικτυακά αιτήματα τυπικά μεταβιβάζονται σε διεργασίες διακομιστή (server) ως συσσωρευτική συστοιχιών συμβολοσειρών.
- Ο γρηγορότερος τρόπος απάντησης σε ένα αίτημα είναι να χρησιμοποιήσετε την ακατέργαστη είσοδο χρήστη.
- Τα αποτελέσματα των injection προέρχονται από τεμπέλες πρακτικές προγραμματισμού σε εφαρμογές και παραλείψεις στο λογισμικό διακομιστή (server).

Συναφείς παρενέργειες της επικύρωσης ανεπαρκούς εισαγωγής

- Cross script scripting
- Έγχυση κακόβουλου λογισμικού
- Επιθέσεις υπερχείλισης ρυθμιστή (buffer)
- Άρνηση παροχής υπηρεσίας
- SQL Injection

Πρόληψη της έγχυσης SQL

Η έγχυση SQL (Injection) μπορεί να προληφθεί χρησιμοποιώντας:

- Αρχές σχεδιασμού:
  - Αποφυγή δομών εφαρμογής που αφήνουν τις εφαρμογές ευάλωτες.

- Πρακτικές κωδικοποίησης:
  - Αποτροπή της εκτέλεσης κακών αποσπασμάτων SQL.
  - Αποκλεισμός κακής εισαγωγής / αποχέτευσης εισόδου (input sanitation).

- Πρακτικές βάσης δεδομένων:
  - Κάνοντας τη βάση δεδομένων λιγότερο ευάλωτη σε οποιοδήποτε είδος επίθεσης.

- Υποστήριξη Υποδομής:
  - Αποτροπή επιθέσεων σε οποιαδήποτε εφαρμογή.

Αρχή σχεδιασμού - Δεν υπάρχουν δεδομένα ανώνυμου χρήστη

- Αναγκάστε τους χρήστες να δημιουργήσουν έναν λογαριασμό, ο οποίος επαληθεύεται με ένα μήνυμα ηλεκτρονικού ταχυδρομείου.

- Χρησιμοποιήστε το Captcha ή παρόμοια γραφικά για την καταχώρηση κειμένου για να αποτρέψετε την εισαγωγή αυτοματοποιημένων / bot δεδομένων σε συστήματα.

- Καταγράψτε όλες τις καταχωρήσεις δεδομένων μέσω αιτήματος ιστού - ποιος, τι, πού, πότε και από ποια διεύθυνση IP.

Αρχή σχεδιασμού - Έλεγχος ταυτότητας

- Εξαλείψτε όλα τα αποθηκευμένα ονόματα χρηστών και κωδικών πρόσβασης βάσει βάσης δεδομένων.
  - Μια σελίδα σύνδεσης είναι το σημείο εισόδου σε μια εφαρμογή και πρέπει να επιτρέπει την καταχώριση ανώνυμων δεδομένων.
  - Η έγχυση SQL χρησιμοποιείται συχνά για να παρακάμψει την ασφάλεια.

- Πολλές ανέξοδες και δωρεάν εναλλακτικές λύσεις υπάρχουν για καταστήματα ελέγχου ταυτότητας
  - Το OpenLDAP είναι εύκολο, δωρεάν και η πρόσβαση γίνεται μέσω κλήσεων LDAP και όχι μέσω SQL.
  - Τα ονόματα χρηστών και οι κωδικοί πρόσβασης μπορούν να επαναχρησιμοποιηθούν σε διαφορετική εφαρμογή χωρίς να επαναχρησιμοποιηθεί μια βάση δεδομένων.

Προσοχή: Μην αναμειγνύετε εσωτερικούς και εξωτερικούς χρήστες στο ίδιο κατάστημα LDAP αν είναι δυνατόν.

Αρχή σχεδιασμού - Αποφύγετε το ελεύθερο κείμενο όπου είναι εφικτό και ποτέ μην δεχτείτε ετικέτες HTML

- Ο περιορισμός των εισροών στις αναπτυσσόμενες και τα μορφοποιημένα πλαίσια κειμένου απλοποιούν τις επικυρώσεις που είναι απαραίτητες για την απόπειρα των προσπαθειών έγχυσης SQL.

- Οι ετικέτες HTML είναι ένα πολύ κοινός φορέας κακόβουλου λογισμικού.
  - Καλύτερα να σπάσετε τις εισροές σε πολλαπλά πεδία κειμένου.
  - Χρησιμοποιήστε τις επιλογές μορφοποίησης μέσω αναπτυσσόμενων πεδίων, πλαισίων ελέγχου και άλλων σταθερών πεδίων εισαγωγής.

Προσοχή: Η πλευρά του πελάτη HTML και JavaScript παρακάμπτονται πολύ εύκολα.

Πρακτική κωδικοποίησης - Καμία δυναμική SQL

- Η καθαρή δυναμική SQL χρησιμεύει ως η πιο κοινή μορφή προσβολών SQL:

```
sqlString = "SELECT… From [myTable]
WHERE name ='".myInputValue."' ";
```

- Πολύ εύκολη στην εισαγωγή εισόδου σε μια δομημένη μορφή χρησιμοποιώντας κάποια παραλλαγή τιμών δέσμευσης ή παραμετροποιημένες εντολές SQL.
  - Οι προετοιμασμένες δηλώσεις της Java.
  - .NET χρησιμοποιώντας τη δέσμευση sql με τις παραμέτρους τιμών.
  - mySQLi και Pear μέσα στην PHP.
  - Οι μεταβλητές δέσμευσης αντιμετωπίζονται ως καθαρή είσοδος και δεν εκτελούνται γενικά.

Προσοχή: Δεν είναι μια αλεξίσφαιρη προσέγγιση, απλώς μια βέλτιστη πρακτική.

Πρακτική κωδικοποίησης - Έλεγχος ισχυρού τύπου πριν από την αλληλεπίδραση με τη βάση δεδομένων

- Στο διακομιστή, ένας επεξεργαστής αιτήματος πρέπει να εκτελέσει έναν ισχυρό έλεγχο τύπου:

  - Βεβαιωθείτε ότι οι αριθμοί είναι αριθμοί, οι ημερομηνίες είναι ημερομηνίες, οι τιμές από τα στοιχεία της φόρμας είναι σωστές, όπως οι δείκτες από τα πτυσσόμενα πεδία (drop downs) κ.λπ.

  - Περιορίστε το εύρος των αποδεκτών τιμών αν είναι δυνατόν.

  - Χρησιμοποιήστε τις λειτουργίες ανίχνευσης που έρχονται εγγενώς με πολλές γλώσσες προγραμματισμού, αν είναι διαθέσιμες, όπως σε Java ή .NET .

  - Είναι πολύ σημαντικό στις γλώσσες που έχουν πληκτρολογηθεί σε ασθενώς πληκτρολογημένες γλώσσες, όπως η PHP, για να εξαναγκάσουν τον έλεγχο τύπου.

Πρακτική κωδικοποίησης - Εκτέλεση μήκους και μορφών εισαγωγής

- Περιορίστε το μέγεθος όλων των συμβολοσειρών τόσο στον πελάτη όσο και στον διακομιστή (server).

  - Ο καθορισμός του μήκους σε ένα πλαίσιο εισαγωγής ή η χρήση του JavaScript παρακάμπτεται εύκολα.
  - Απορρίψτε οποιοδήποτε αίτημα όπου οποιαδήποτε τιμή υπερβαίνει ένα μέγιστο αναμενόμενο μήκος.

- Βεβαιωθείτε ότι κάθε είσοδος συμβολοσειρών συμμορφώνεται με τις μάσκες εισόδου όποτε είναι δυνατόν, για παράδειγμα:

  - Όταν οι συμβολοσειρές χρησιμοποιούνται για να περιγράψουν τους αριθμούς για να διατηρήσουν τα αρχικά μηδενικά, βεβαιωθείτε ότι όλοι οι χαρακτήρες είναι αριθμοί.
  - Φιλτράρετε για τυχόν μη αναμενόμενους χαρακτήρες.

Προσοχή: Οι κανόνες πρέπει να εφαρμόζονται στον πελάτη και στον διακομιστή.

Πρακτική κωδικοποίησης - Καθαρίστε όλες τις εισροές χρηστών πριν από οποιαδήποτε άλλη επεξεργασία

- Η ασφαλέστερη και πιο ακίνδυνη πρακτική είναι η επανάληψη μέσω διαδικτυακού αιτήματος και η διήθηση (filter) όλων των μη αναμενόμενων χαρακτήρων.
    - Αν αφαιρεθούν όλοι οι ειδικοί χαρακτήρες, απενεργοποιούνται απλά οι κλήσεις λειτουργίας, η κωδικοποίηση URI και άλλοι συνηθισμένοι τρόποι προσθήκης ευρετηρίων SQL ή ενσωματωμένων ετικετών HTML.

- Αφαιρέστε τους ανεπιθύμητους χώρους.

- Απορρίψτε αιτήματα με ανωμαλίες και καταγράψτε τη δραστηριότητα για ανάλυση.

Πρακτική κωδικοποίησης - Κάλυψη όλων των σφαλμάτων από το χρήστη με φιλική προς το χρήστη έξοδο

- Ποτέ μην εμφανίζετε σφάλματα SQL ή άλλα ακατάλληλα σφάλματα συστήματος πίσω στον χρήστη.
  - Μπορεί να παρέχει επιπλέον φορείς επίθεσης για χάκερ.

- Κάθε φορά που εμφανίζεται μια εξαίρεση, εμφανίζεται ένα γενικό μήνυμα και καταγράψτε το πραγματικό σφάλμα και την είσοδο του χρήστη.

- Κάθε φορά που μια αίτηση αποτυγχάνει ελέγχους επικύρωσης ή εξυγίανσης, χρησιμοποιήστε μια γενική απάντηση, τερματίστε τη συνεδρία του χρήστη και καταγράψτε λεπτομερώς το σφάλμα.

Προσοχή: ΠΟΤΕ μην επαναλάβετε την είσοδο του χρήστη πίσω στον χρήστη χωρίς απολύμανση του αιτήματος. Αυτή είναι η πιο συνηθισμένη μορφή διαδικτυακής δέσμης ενεργειών.

Πρακτική κωδικοποίησης - Χρήση λεπτομερών αρχείων καταγραφής

- Η λεπτομερής καταγραφή εισάγει πρόσθετα έξοδα αποθήκευσης και επεξεργασίας, αλλά είναι ανεκτίμητο για την ανίχνευση σφαλμάτων και τον εντοπισμό αδυναμιών ασφάλειας.

- Οι απροσδόκητες συνθήκες, τα απορριφθέντα αιτήματα και παρόμοια σφάλματα είναι συνήθως το πρώτο σημάδι στην επίθεση της εφαρμογής σας στο διαδίκτυο.

Πρακτικές κωδικοποίησης - Χρησιμοποιήστε πλαίσια

- Κάθε δημοφιλής πλατφόρμα ανάπτυξης ιστού έχει πλαίσια επικύρωσης.

- Αξιοποίηση υφιστάμενων πλαισίων για την εφαρμογή επικυρώσεων:
  - Zend για PHP.
  - Αρκετά πλαίσια για την Java.

- Τα πλαίσια μπορούν να συγκεντρώσουν πολλά καθήκοντα που σχετίζονται με την ασφάλεια.

- Το .NET έχει ένα αναδυόμενο μοντέλο MVC το οποίο θα βοηθήσει στη συγκέντρωση εργασιών όπως η αποχέτευση εισόδου (input sanitation).

Πρακτική βάσης δεδομένων - Χρησιμοποιήστε δύο λογαριασμούς

- Δημιουργία δύο λογαριασμών:
  - Ιδιοκτήτης βάσης δεδομένων.
    - Έχει δικαιώματα πάνω σε όλα τα αντικείμενα σε μια βάση δεδομένων ή ένα σχήμα.
    - Ισοδύναμο με την πρόσβαση σε επίπεδο DBA για μια βάση δεδομένων / σχήμα.
    - Χρησιμοποιείται για τη δημιουργία και τη διατήρηση μιας βάσης δεδομένων εφαρμογών.
    - Ποτέ δεν χρησιμοποιείται από διαδικτυακές εφαρμογές.

  - Λογαριασμός εφαρμογών / λογαριασμός μεσολάβησης βάσης δεδομένων.
    - Διαθέτει ελάχιστα δικαιώματα για την εφαρμογή:
      - Όλα τα δικαιώματα σε κάθε αντικείμενο δηλώνονται ρητά.
      - Δεν κατέχει άμεσα αντικείμενα.
      - Δεν υπάρχει πρόσβαση σε μεταδεδομένα στην πλατφόρμα βάσης δεδομένων.
      - Περιορισμένες τοποθεσίες σύνδεσης εάν είναι δυνατόν.

Πρακτικές Βάσεων Δεδομένων - Ισχυρή Πληκτρολόγηση

- Οι στήλες πρέπει να πληκτρολογούνται έντονα:

  - Αριθμοί ως αριθμοί.
  - Οι χαρακτήρες περιορίζονται στο ακριβές μέγιστο που απαιτείται.
  - Ημερομηνίες αποθηκευμένες ως ημερομηνίες

- Εάν η απόδοση είναι αποδεκτή, χρησιμοποιήστε τους περιορισμούς ή τους παράγοντες ενεργοποίησης:

  - Οι μάσκες μορφοποίησης δύναμης και τα εύρη χαρακτήρων όπως το 0-9 για το SSN κ.λπ.

- Χρησιμοποιήστε τους πίνακες αναζήτησης για τιμές αναφοράς και επιβάλλετε ξένα πλήκτρα.

Πρακτικές βάσης δεδομένων - αποθηκευμένες διαδικασίες και προβολές

- Προβολές:
  - Εκθέστε μόνο τις στήλες που απαιτούνται από την εφαρμογή.
  - Επιτρέψτε σε πιο λεπτομερείς στήλες ανά δικαιώματα στήλης.

- Αποθηκευμένες διαδικασίες:
  - Ο λογαριασμός εφαρμογής αποκτά δικαιώματα εκτέλεσης μόνο.
    - Όλοι οι πίνακες και οι προβολές είναι αόρατοι.
  - Μπορεί να μειώσει τον αριθμό αλληλεπιδράσεων βάσης δεδομένων.
  - Απλοποιεί τη διαχείριση συναλλαγών.
  - Δεν είναι κατάλληλο για όλα τα περιβάλλοντα εφαρμογών/εργαλεία.

Πρακτική Υποδομής - Εγκαταστήστε ένα IDS που ελέγχει συγκεκριμένα την έγχυση SQL (Injection)

- Αρκετά συστήματα IDS υπάρχουν για να παρακολουθούν ειδικά την κυκλοφορία ιστού για SQL Injection.
  - Κάθε αίτηση εξετάζεται για υπογραφές SQL injection.
  - Τα κακά αιτήματα φιλτράρονται και καταγράφονται.

- Προστατεύει όλες τις εφαρμογές από τα πιο κοινά σφάλματα.

- Εξαιρετικό πρώτο βήμα έως ότου όλες οι διαδικτυακές εφαρμογές μπορούν να αναθεωρηθούν για ευπάθειες.

Προσοχή: Οι υπογραφές θα είναι πάντα τελικά νικημένες.

Πρακτική υποδομής - Κάνει αυτόματη σάρωση αρχείων καταγραφής

- Χρησιμοποιήστε ένα κεντρικό εργαλείο καταγραφής που ψάχνει για συμπεριφορά SQL Injection:

  - Σφάλματα SQL.
  - Ερωτήματα ενάντια των μεταδεδομένων από διαδικτυακές εφαρμογές.

- Χρησιμοποιήστε εργαλεία χειροκίνητης σάρωσης, όπως grep, awk και parsers καταγραφής, για να αναζητήσετε μη εξουσιοδοτημένα ερωτήματα / αιτήσεις sql.

Πρακτική υποδομής - Χρησιμοποιήστε εργαλεία σάρωσης ασφαλείας

- Το καλύτερο μέτρο ασφάλειας είναι αυτό που συλλαμβάνει τα προβλήματα πριν αποκαλυφθούν.

- Ως μέρος των δοκιμών QAT ή UAT, οι εφαρμογές πρέπει να σαρώνονται αυτόματα για ευπάθειες.

- Αποκαλύπτει ευπάθειες πέραν και πέρα από απλές δοκιμές διείσδυσης.

- Πολλά εξαιρετικά προϊόντα:
  - Ορθολογικό APPSCAN από την IBM (παρακολούθησε την παρακολούθηση)
  - Acunetix.

Ειδικά Θέματα της πλατφόρμας

- Η PHP, η Java και άλλες γλώσσες προγραμματισμού ιστού μπορούν να καθαρίσουν εύκολα τις αιτήσεις των χρηστών.
  - Τα αιτήματα μπορούν να τροποποιηθούν και οι τιμές να αντικατασταθούν.
  - Το MVC πληκτρολογεί μοντέλα στη Java και περιλαμβάνει οδηγίες σε άλλες γλώσσες δέσμης ενεργειών καθιστώντας αυτήν ασήμαντη.

- Το .NET 2.0 και ανώτερα θέτουν μια πιο δύσκολη πρόκληση, διότι τα αντικείμενα αιτήματος μόνο διαβάζονται.
  - Η εφαρμογή υγιεινής καταργεί πολλά από τα πλεονεκτήματα του .NET προγραμματισμού.
  - Χρησιμοποιώντας φίλτρα τύπου ISAPI, τα αιτήματα προ-επεξεργασίας απαιτούν μεγάλη περιττή κωδικοποίηση.

- Η αδύναμη πληκτρολόγηση σε πολλές από τις διαδικτυακές γλώσσες προγραμματισμού ανοιχτής πηγής (Open Source) δημιουργεί πρόσθετες απαιτήσεις για επαγρύπνηση σε αυτές τις πλατφόρμες.

Συμπέρασμα

- Η ασφάλεια είναι μια συνεχής διαδικασία επαγρύπνησης.
  - Κάθε τεμάχιο / μέτρο έχει τελικά έναν μετρητή.

- Οι καλές συνήθειες στο πλαίσιο της ανάπτυξης και του σχεδιασμού των εφαρμογών μπορούν να αποτρέψουν τα περισσότερα εκμεταλλεύματα τύπου έγχυσης (injection) / υπερχείλισης.

- Η ασφάλεια εφαρμογών απαιτεί δράση από προγραμματιστές, σχεδιαστές και διαχειριστές.

- Οι περισσότερες επιθέσεις SQL Injection μπορούν να μπλοκαριστούν με πολύ απλά μέτρα, τα οποία μπορεί να είναι εντατικής εργασίας για την υλοποίηση.

Βιβλιογραφία

Κανονικές εκφράσεις για φιλτράρισμα:

http://www.securityfocus.com/infocus/1768

Παράδειγμα πειρατείας προέλευσης:

http://video.google.com/videosearch?q=sql+injection&oe=utf-8&rls=org.mozilla:en-US:official&client=firefox-a&um=1&ie=UTF-8&sa=X&oi=video_result_group&resnum=5&ct=title#

Λογισμικό σάρωσης εφαρμογών:

http://www-01.ibm.com/software/awdtools/appscan/