OWASP
TOP TEN

Μετάφραση: **Καζακώνης Αναστάσιος**

ΙΟΥΝΙΟΣ 2005

Εισαγωγή

Το Open Web Application Security Project (OWASP) Είναι αφοσιωμένο στο να βοηθάει τους οργανισμούς να καταλάβουν και να βελτιώσουν την ασφάλεια των δικτυακών τους εφαρμογών και των δικτυακών τους υπηρεσιών. Η λίστα αυτή δημιουργήθηκε για να κάνει τους οργανισμούς και τις κυβερνητικές υπηρεσίες να εστιάσουν στα πιο σοβαρά από αυτά τα προβλήματα ασφαλείας. Η ασφάλεια των δικτυακών εφαρμογών έχει γίνει ένα καυτό θέμα καθώς οι εταιρείες σπεύδουν να κάνουν τις υπηρεσίες και το υλικό που προσφέρουν προσβάσιμα μέσα από το internet. Την ίδια ώρα οι επιτιθέμενοι στρέφουν την προσοχή τους στις απλές αδυναμίες που δημιουργούνται κατά την ανάπτυξη των προγραμμάτων.

Όταν ένας οργανισμός ανεβάζει μια δικτυακή εφαρμογή καλεί τον κόσμο να στείλει HTTP αιτήματα. Επιθέσεις που βασίζονται σε αυτά τα αιτήματα περνάνε μέσα από firewalls, φίλτρα, πλατφόρμες και συστήματα ανίχνευσης επιθέσεων χωρίς να κοπούν γιατί εντάσσονται μέσα στο λογικό και αναμενόμενο HTTP αίτημα. Ακόμη και οι ασφαλείς δικτυακοί τόποι που χρησιμοποιούν SSL απλά δέχονται τα αιτήματα που φτάνουν μέσα από την κωδικοποιημένη οδό χωρίς να το σταματήσουν. **Αυτό σημαίνει ότι ο κώδικας της δικτυακής σας εφαρμογής είναι μέρος της περιμέτρου ασφάλειά σας.** Καθώς ο αριθμός το μέγεθος και η πολυπλοκότητα των δικτυακών εφαρμογών αυξάνει τόσο αυξάνει και η περίμετρος ασφαλείας που πρέπει να προτάξουμε.

Αυτά τα θέματα ασφάλειας που δημιουργούνται εδώ δεν είναι καινούρια. Στην πραγματικότητα, κάποια από αυτά τα έχουμε αντιληφθεί εδώ και δεκαετίες. Αλλά ακόμη, για πολλούς λόγους, σε μεγάλα project ανάπτυξης λογισμικού γίνονται ακόμη αυτά τα λάθη προσβάλλοντας, όχι μόνο την ασφάλεια των πελατών τους, αλλά και την ασφάλεια ολόκληρου του Διαδικτύου. Δεν υπάρχει καμιά χρυσή λύση για να θεραπεύσουμε το πρόβλημα αυτό. Η σημερινή τεχνολογία για την προστασία των δικτυακών εφαρμογών συνεχώς εξελίσσεται, αλλά αυτή τη στιγμή μπορεί μόνο να αντιμετωπίσει έναν περιορισμένο αριθμό υποπεριπτώσεων των θεμάτων που εμφανίζονται, στην καλύτερη περίπτωση. Για να συνοψίσουμε τα θέματα που περιγράφονται σε αυτό το δοκίμιο, οι οργανισμοί θα χρειαστεί να αλλάξουν τον τρόπο με τον οποίο αναπτύσσουν τις εφαρμογές τους, να εκπαιδεύσουν τους προγραμματιστές τους, να ανανεώσουν τις διαδικασίες ανάπτυξης του λογισμικού τους και να χρησιμοποιούσαν την τεχνολογία όπου αυτό χρειάζεται.

Η OWASP Top Ten είναι μια λίστα προβλημάτων ασφαλείας που απαιτούν άμεση αντιμετώπιση. Ο ήδη υπάρχων κώδικας πρέπει να ελεγχθεί για αυτά τα προβλήματα ασφάλειας άμεσα, καθώς τα σημεία αυτά αποτελούν πρωτεύοντες στόχους για τους επιτιθέμενους. Τα προγράμματα ανάπτυξης λογισμικού πρέπει να αναφέρουν τα προβλήματα αυτά στα συνοδευτικά τους έγραφα, να σχεδιάζονται, να υλοποιούνται και να ελέγχουν τις εφαρμογές τους για να επιβεβαιώσουν ότι δεν κινδυνεύουν από κάποιο από αυτά. Οι διευθυντές των προγραμμάτων αυτών πρέπει να αφιερώνουν χρόνο και χρήμα για δραστηριότητες σχετικά με την ασφάλεια των εφαρμογών συμπεριλαμβάνοντας εκπαίδευση των προγραμματιστών, ανάπτυξη πολιτικής ασφαλείας για τις εφαρμογές, σχεδιασμό μηχανισμού ασφαλείας, έλεγχο κατά των επιθέσεων, και εξέταση του κώδικα.

Στηρίζουμε τους οργανισμούς που συμπεριλαμβάνονται στην συνεχώς διευρυνόμενη λίστα των εταιρειών που έχουν υιοθετήσει το **OWASP Top Ten** σαν ένα ελάχιστο πρότυπο ασφαλείας και έχουν δεσμευτεί να δημιουργούν δικτυακές εφαρμογές απαλλαγμένες από τα προβλήματα αυτά.

Διαλέξαμε να παρουσιάσουμε τη λίστα αυτή σε μια μορφή παρόμοια με την εξαιρετικά επιτυχημένη SANS/FBI Top Twenty List με στόχο να διευκολύνουμε τη χρήση και την κατανόησή της. Η SANS/FBI Top Twenty List εστιάζει σε συγκεκριμένα προβλήματα ευρέως χρησιμοποιούμενων δικτύων και δικτυακών προϊόντων. Καθώς κάθε δικτυακός τόπος είναι μοναδικός το κείμενο αυτό οργανώνεται σύμφωνα με συγκεκριμένους τύπους ή κατηγορίες προβλημάτων ασφαλείας που συχνά δημιουργούνται σε δικτυακές εφαρμογές. Οι κατηγορίες αυτές ορίζονται στο OASIS Web Application Security (WAS) XML Project.

Η λίστα αυτή παρουσιάζει τη συνδυασμένη σοφία των ειδικών του OWASP, των οποίων η εμπειρία περιλαμβάνει πολλά χρόνια δουλειάς για την ασφάλεια κυβερνητικών εφαρμογών, οικονομικών, φαρμακευτικών και βιομηχανικών υπηρεσιών, καθώς και εργαλείων ανάπτυξης και της τεχνολογίας. Το έγγραφο αυτό έχει σχεδιαστεί με στόχο να παρουσιάσει τα πιο σοβαρά προβλήματα ασφάλειας των δικτυακών εφαρμογών. Υπάρχουν πολλά βιβλία και οδηγοί που περιγράφουν τα προβλήματα αυτά με περισσότερες λεπτομέρειες και παρέχουν λεπτομερή καθοδήγηση σχετικά με το πως θα απαλλαγούμε από αυτά. Ένας τέτοιος οδηγός είναι ο OWASP Guide που είναι διαθέσιμος στο http://www.owasp.org

Το OWASP Top Ten είναι ένα δυναμικό έγγραφο που συνεχώς εξελίσσεται. Περιλαμβάνει οδηγίες και links σε επιπλέον πληροφορίες χρήσιμες για να διορθώσουμε αυτούς τους τύπους των προβλημάτων ασφαλείας. Ανανεώνουμε συνεχώς τη λίστα και τις οδηγίες καθώς περισσότερες και πιο κρίσιμες απειλές εμφανίζονται συνεχώς, ενώ ολοένα και περισσότερο πρόσφατα ενημερωμένες μέθοδοι ανακαλύπτονται κάθε τόσο. Ενθαρρύνουμε διαρκώς την προσφορά σας στην προσπάθεια αυτή. Το έγγραφο αυτό στηρίζεται σε μια κοινότητα και η εμπειρία σας στην αντιμετώπιση των επιτιθέμενων και στην εξάλειψη των προβλημάτων ασφαλείας που παρουσιάζουμε μπορούν να βοηθήσουν αυτούς που θα έρθουν μετά από μας. Μπορείτε να στείλετε προτάσεις στο topten@owasp.org με θέμα "OWASP Top Ten Comments."

Υπόβαθρο

Η πρόκληση του να αναγνωρίσουμε τα κορυφαία προβλήματα ασφάλειας των δικτυακών εφαρμογών από μια πρώτη ματιά είναι κάτι το ανέφικτο. Δεν υπάρχει καν μια κοινώς αποδεκτή συμφωνία σχετικά με το τι συμπεριλαμβάνεται στον όρο «ασφάλεια δικτυακής εφαρμογής». Κάποιοι υποστηρίζουν ότι θα έπρεπε να εστιάσουμε μόνο σε προβλήματα ασφάλειας που επηρεάζουν τους προγραμματιστές που γράφουν κώδικα δικτυακών εφαρμογών. Άλλοι πάλι υποστηρίζουν ένα πιο διευρυμένο ορισμό που θα καλύπτει ολόκληρο το επίπεδο της εφαρμογής συμπεριλαμβάνοντας τις βιβλιοθήκες, το στήσιμο του διακομιστή και το επίπεδο των πρωτοκόλλων της εφαρμογής. Με στόχο να ορίσουμε τους κυριότερους κινδύνους που αντιμετωπίζουν οι οργανισμοί, αποφασίσαμε να δώσουμε μια σχετικά διευρυμένη ερμηνεία του όρου «ασφάλεια δικτυακών εφαρμογών» ενώ αποστασιοποιούμεθα από θέματα ασφάλειας δικτύων και δικτυακών κατασκευών.

Ακόμη μια πρόκληση για την προσπάθεια αυτή είναι το γεγονός ότι κάθε διακριτό πρόβλημα ασφάλειας είναι μοναδικό για τον δικτυακό τόπο ενός συγκεκριμένου Οργανισμού. Θα χάναμε το στόχο εάν ασχολούμασταν με συγκεκριμένα προβλήματα ασφάλειας των δικτυακών εφαρμογών για κάθε οργανισμό ξεχωριστά ειδικά τη στιγμή που συνήθως αυτά διορθώνονται άμεσα, αφού ένα μεγάλο κοινό μαθαίνει άμεσα για την ύπαρξή τους. Παρόλα αυτά διαλέξαμε να επικεντρωθούμε στις κορυφαίες κλάσεις, τύπους ή κατηγορίες προβλημάτων ασφαλείας των δικτυακών εφαρμογών.

Στην πρώτη έγκριση του κειμένου αυτού, αποφασίσαμε να αναφέρουμε μια ευρεία γκάμα προβλημάτων δικτυακών εφαρμογών σε κατηγορίες που είχαν μικρή σημασία. Μελετήσαμε μια ποικιλία κατηγοριών ασφαλείας και φτάσαμε σε μια ομάδα κινδύνων. Παράγοντες που χαρακτηρίζουν μια κατηγορία σαν αυτές που αναφέρουμε, περιλαμβάνουν κριτήρια όπως αν οι διαρροές σχετίζονται άμεσα με παρόμοια μέτρα ασφαλείας, ή αν τα προβλήματα αυτά συχνά διαπιστώνονται σε συγκεκριμένες αρχιτεκτονικές εφαρμογών. Στην έκδοση αυτή παρουσιάζουμε ένα ανανεωμένο σχήμα. Αυτό αναπτύχθηκε με τη συνεχιζόμενη δουλειά μας στην τεχνική επιτροπή του OASIS WAS στην οποία περιγράφουμε έναν Θησαυρό θεμάτων από τα οποία οι ερευνητές που ασχολούνται με την ασφάλεια μπορούν να περιγράψουν υπογραφές σε XML format.

Το να διαλέξουμε τους δέκα κυριότερους κινδύνους από μια μεγάλη η λίστα υποψηφίων, έχει τις δικές του δυσκολίες. Απλά δεν υπάρχουν αξιόπιστες πηγές στατιστικών σχετικά με τα προβλήματα ασφάλειας που αντιμετωπίζουν οι δικτυακές εφαρμογές. Στο μέλλον θα θέλαμε να συλλέξουμε στατιστικά στοιχεία σχετικά με τη συχνότητα συγκεκριμένων διαρροών σε κώδικα δικτυακών εφαρμογών και να χρησιμοποιήσουμε τα νούμερα αυτά έτσι ώστε να βοηθηθούμε να θέσουμε κάποιες προτεραιότητες σε αυτούς τους δέκα κυριότερους κινδύνους. Παρόλα, αυτά για κάποιους λόγους αυτός ο τρόπος μετρήσεων δεν προβλέπεται να υλοποιηθεί στο κοντινό μέλλον.

Αναγνωρίζουμε ότι δεν υπάρχει μια σωστή απάντηση στο ερώτημα «ποιες κατηγορίες κινδύνων θα πρέπει να βρίσκονται μέσα στις δέκα κυριότερες». Κάθε οργανισμός θα πρέπει να σκεφτεί τους κινδύνους που έχει να αντιμετωπίσει βασισμένος στην

πιθανότητα να έχει κάποια από τις διαρροές που περιγράφουμε, και τους κινδύνους των επιπτώσεων που θα υποστεί από κάτι τέτοιο. Στο μεταξύ δίνουμε αυτή τη λίστα σαν ένα σύνολο προβλημάτων που εκπροσωπούν ένα συγκεκριμένο αριθμό κινδύνων που ίσως αντιμετωπίσει ένα μεγάλο σύνολο οργανισμών. Οι δέκα αυτοί οι κίνδυνοι δεν βρίσκονται σε μια συγκεκριμένη σειρά καθώς θα ήταν σχεδόν αδύνατο να αποφασίσουμε ποιοι από αυτούς εκπροσωπούν τον πιο σημαντικό κίνδυνο.

Το OWASP Top Ten project είναι μια προσπάθεια που συνεχίζεται για να κάνουμε τις πληροφορίες σχετικά με τις διαρροές ασφάλειας των δικτυακών εφαρμογών διαθέσιμες σε ένα μεγάλο κοινό. Στοχεύουμε να ανανεώνουμε το έγγραφο αυτό κάθε χρόνο βασισμένοι στη συζήτηση που διεξάγεται στην OWASP κοινότητα μας, μέσα από τη mailing list και στα σχόλια που έρχονται στο topten@owasp.org

Τι άλλαξε από την προηγούμενη έκδοση;

Το OWASP έχει αλλάξει αρκετά από την τελευταία φορά που εκδόθηκε τον Ιανουάριο του 2003. Αυτή η ανανεωμένη έκδοση περιλαμβάνει όλες τις συζητήσεις που έγιναν μέχρι σήμερα, τις απόψεις και τις αντιθέσεις που διατυπώθηκαν στην κοινότητα του OWASP τους τελευταίους δώδεκα μήνες. Πάνω απ' όλα, έχουν γίνει μεγάλες βελτιώσεις σε όλα τα τμήματα του και μόνο μερικές μικρές αλλαγές.

- **WAS-XML Alignment** – Ένα από τα νέα projects που ξεκίνησαν το 2003 είναι το Web Application Security Technical Committee (WAS TC) στο OASIS. Ο στόχος του WAS TC είναι να παράγει ένα σχήμα που να ταξινομεί – κωδικοποιεί την δικτυακή ασφάλεια και τα προβλήματα που μπορεί να παρουσιάσει, ένα μοντέλο για να καθοδηγήσει αρχικού βεληνεκούς απειλές, να συγκρουστεί και να εκτιμήσει τους κινδύνους, αλλά και να δημιουργήσει ένα XML σχήμα για να περιγράψει τις συνθήκες της δικτυακής ασφάλειας που μπορεί να χρησιμοποιηθούν από εργαλεία ελέγχου και εργαλεία προστασίας από κοινού. Το OWASP Top Ten project χρησιμοποιεί το WAS TC ως μια αναφορά για να αναπροσαρμόσει την εικόνα του Top Ten και να παρέχει μια κανονικοποιημένη προσέγγιση στην ταξινόμηση των μειονεκτημάτων ασφαλείας των δικτυακών εφαρμογών. Το WAS Thesaurus ορίζει μια κανονικοποιημένη γλώσσα για τη συζήτηση σχετικά με τη δικτυακή ασφάλεια και υιοθετούμε το εν λόγω λεξιλόγιο και εδώ.
- **Πρόσθεση του κεφαλαίου άρνησης παροχής υπηρεσιών** – Η μόνη κατηγορία του Top Ten που άλλαξε ήταν η προσθήκη του κεφαλαίου 9 που αναφέρεται στην άρνηση παροχής υπηρεσιών. Η έρευνά μας έδειξε ότι ένα μεγάλο εύρος οργανισμών είναι δεκτικό σε αυτό το είδος των κινδύνων. Βασισμένοι στην πιθανότητα μιας επίθεσης άρνησης παροχής υπηρεσιών και στις συνέπειες που μπορεί να προκύψουν εάν η επίθεση αυτή πετύχει αποφασίσαμε ότι το κεφάλαιο αυτό είναι αναγκαίο να συμπεριληφθεί στη λίστα. Για να φιλοξενήσουμε τη νέα αυτή είσοδο συνδυάσαμε το κεφάλαιο 9 του προηγούμενου χρόνου σχετικό με τις διαρροές ασφαλείας από την απομακρυσμένη διαχείριση με το κεφάλαιο 2 που αναφέρεται στην κατηγορία ελέγχου των παράνομων προσβάσεων καθώς αυτό αποτελεί ειδική περίπτωση της κατηγορίας αυτής. Πιστεύουμε κάτι τέτοιο είναι λογικό καθώς οι τύποι των διαρροών στο δεύτερο κεφάλαιο τυπικά είναι ίδιοι με αυτούς στο ένατο και απαιτούν ίδιους τρόπους αντίδρασης.

Ο πίνακας παρακάτω εμφανίζει τη σχέση ανάμεσα στην καινούρια λίστα Top Ten, στην περσινή λίστα Top Ten και στο νέο WAS TC Thesaurus.

Νέο Top Ten 2004	**Top Ten του 2003**	**Νέο WAS Thesaurus**
A1 Μη έγκυρη είσοδος	A1 Μη έγκυρες παράμετροι	Εξακρίβωση της εγκυρότητας της εισόδου
A2 Έλεγχος παράνομης πρόσβασης	A2 Έλεγχος παράνομης πρόσβασης *(Περιλαμβάνει το A9 Διαρροές ασφαλείας από απομακρυσμένη πρόσβαση για τη διαχείριση)*	Έλεγχος των προσβάσεων

A3 Παράνομη πιστοποίηση και διαχείριση των συνδέσεων	A3 Σπασμένοι λογαριασμοί και διαχείριση των συνδέσεων	Πιστοποίηση χρηστών και διαχείριση συνδέσεων
A4 Cross Site Scripting και (XSS) Flaws	A4 Cross Site Scripting (XSS) Flaws	Πιστοποίηση της εισόδου→ Cross site scripting
A5 Υπερχείλιση του Buffer	A5 Υπερχείλιση του Buffer	Υπερχείλιση του Buffer
A6 Injection Flaws	A6 Διαρροές με εμφώλευση εντολών	Πιστοποίηση της εισόδου→ Injection
A7 Λανθασμένος έλεγχος λαθών	A7 Προβλήματα ελέγχου λαθών	Χειρισμός λαθών
A8 Μη ασφαλής αποθήκευση δεδομένων	A8 Μη ασφαλής χρήση της κρυπτογραφίας	Προστασία Δεδομένων
A9 Άρνηση παροχής υπηρεσιών	*Δεν διατίθεται*	Διαθεσιμότητα
A10 Μη ασφαλής διαχείριση παραμετροποίησης	A10 Κακή παραμετροποίηση των Web Server και των διακομιστών εφαρμογών	Διαχείριση της παραμετροποίησης των εφαρμογών και υποδομών

Η λίστα του Top Ten

Τα σημαντικότερα προβλήματα ασφαλείας των δικτυακών εφαρμογών		
A1	Μη πιστοποιημένη είσοδος	Δεδομένα που προέρχονται από αιτήματα μέσω δικτύου δεν πιστοποιούνται πριν χρησιμοποιηθούν από μια δικτυακή εφαρμογή. Οι επιτιθέμενοι μπορούν να χρησιμοποιήσουν αυτή τη διαρροή ασφαλείας για να επιτεθούν στα στοιχεία που την απαρτίζουν μέσα από μια εφαρμογή
A2	Έλεγχος πρόσβασης	Περιορισμοί σχετικά με τις επιτρεπόμενες ενέργειες των πιστοποιημένων χρηστών δεν επιβάλλονται όπως θα έπρεπε. Οι επιτιθέμενοι μπορούν να ανακαλύψουν τα κενά αυτά για να πάρουν πρόσβαση στους λογαριασμούς άλλων χρηστών, να δουν ευαίσθητα δεδομένα ή να κάνουν χρήση συναρτήσεων για τις οποίες δεν έχουν δικαιώματα.
A3	Διαχείριση προσβάσεων και συνδέσεων	Ιδιότητες των λογαριασμών και συνδέσεις που έχουν γίνει δεν προστατεύονται επαρκώς. Επιτιθέμενοι που μπορούν να χρησιμοποιήσουν κωδικούς, κλειδιά, session cookies, ή άλλα κεκτημένα μπορούν να ξεπεράσουν τη διαδικασία πιστοποίησης και τους περιορισμούς που αυτή επιβάλλει και να συνδεθούν προσποιούμενοι ότι πρόκειται για άλλους χρήστες.
A4	Διαρροές μέσω Cross Site Scripting (XSS)	Η δικτυακή εφαρμογή μπορεί να χρησιμοποιηθεί σαν ένας μηχανισμός μεταφοράς επιθέσεων στον browser του τελικού χρήστη. Μια επιτυχημένη επίθεση μπορεί να κρατήσει ανοιχτή τη σύνδεση ενός χρήστη, να μπει στο τοπικό μηχάνημα ή να αλλάξει δεδομένα και περιεχόμενα για να ξεγελάσει τον χρήστη.
A5	Υπερχείλιση των Buffer	Τα στοιχεία μιας δικτυακής εφαρμογής σε μερικές γλώσσες που δεν επικυρώνουν την είσοδο μπορεί crashάρουν και, σε μερικές περιπτώσεις, να χρησιμοποιηθούν για να πάρει ένας επιτιθέμενος τον έλεγχο της εφαρμογής. Τα στοιχεία αυτά μπορεί να περιλαμβάνουν CGI, βιβλιοθήκες, οδηγούς συσκευών, και στοιχεία δικτυακών εφαρμογών διακομιστών.
A6	Διαρροές μέσω Injection	Οι δικτυακές εφαρμογές περνούν παραμέτρους όταν παίρνουν πρόσβαση σε εξωτερικά συστήματα ή σε τοπικά λειτουργικά συστήματα. Αν ένας επιτιθέμενος μπορέσει να εμβολιάσει κακόβουλες εντολές μέσα σε αυτές τις παραμέτρους, το εξωτερικό σύστημα θα εκτελέσει τις εντολές αυτές για λογαριασμό της εφαρμογής.
A7	Μη αρμόζουσα διαχείριση λαθών	Καταστάσεις λάθους που προκύπτουν κατά τη διάρκεια κανονικών λειτουργιών δεν διαχειρίζονται ορθά. Εάν ένας επιτιθέμενος μπορεί να προκαλέσει σφάλματα για να δώσει την εικόνα ότι η εφαρμογή δεν συμπεριφέρεται σωστά, μπορεί να συλλέξει λεπτομερείς πληροφορίες σχετικές με το σύστημα, να προκαλέσει άρνηση παροχής υπηρεσιών, να ρίξει μηχανισμούς ασφαλείας ή να crashάρει τον server.
A8	Μη ασφαλής αποθήκευση	Δικτυακές εφαρμογές συχνά χρησιμοποιούν εντολές κρυπτογράφησης για να προστατεύσουν πληροφορίες και πιστοποιητικά. Οι εντολές αυτές και ο κώδικας που τις ενοποιεί έχει αποδειχθεί ότι είναι δύσκολο να κωδικοποιηθούν σωστά. Το

		γεγονός αυτό συχνά μας οδηγεί σε αδύναμα μέτρα ασφαλείας.
A9	Άρνηση παροχής υπηρεσιών	Οι επιτιθέμενοι μπορούν να καταναλώσουν πόρους της δικτυακής εφαρμογής σε σημείο που άλλοι νόμιμοι χρήστες να μην μπορούν πλέον να πάρουν πρόσβαση σε αυτή. Οι επιτιθέμενοι μπορούν επίσης να κλειδώσουν τους λογαριασμούς άλλων χρηστών ή ακόμη να προκαλέσουν πτώση ολόκληρης της εφαρμογής.
A10	Μη ασφαλής διαχείριση της παραμετροποίησης	Το να έχουμε ένα ισχυρό πρότυπο παραμετροποίησης των διακομιστών μας είναι σημαντικό για την ασφάλεια των δικτυακών μας εφαρμογών. Οι διακομιστές αυτοί έχουν πολλές επιλογές παραμετροποίησης που επηρεάζουν την ασφάλεια.

A1 Μη πιστοποιημένη είσοδος

A1.1 Περιγραφή

Οι δικτυακές εφαρμογές χρησιμοποιούν εισόδους από HTTP αιτήματα, και μερικές φορές από αρχεία, για να αποφασίσουν τον τρόπο που θα αντιδράσουν. Οι επιτιθέμενοι μπορούν να ασχοληθούν με οποιοδήποτε μέρος ενός τέτοιου HTTP αιτήματος συμπεριλαμβανομένων των url, των συμβολοσειρών που περιέχουν την ερώτηση, των headers, των cookies, των πεδίων της φόρμας και των κρυμμένων πεδίων για να προσπαθήσουν να προσπεράσουν τους μηχανισμούς ασφαλείας του δικτυακού τόπου. Απλοί όροι που αφορούν επιθέσεις μέσω της "ενασχόλησης" με την είσοδο που ο δικτυακός τόπος ζητάει, αποτελούν οι: forced browsing, εισαγωγή εντολών (command insertion), cross site scripting, υπερχείλιση buffer, επίθεση μέσω μορφοποίησης συμβολοσειρών (format string attacks), SQL injection, cookie poisoning και χρήση κρυμμένων πεδίων (hidden field manipulation). Κάθε μια από αυτές τις μεθόδους επίθεσης περιγράφεται με περισσότερες λεπτομέρειες αργότερα στο δοκίμιο αυτό.

- A4 – Διαρροές μέσω Cross Site Scripting: Το κεφάλαιο αυτό πραγματεύεται επιθέσεις μέσω εισόδων που περιέχουν scripts που στόχο έχουν να εκτελεστούν στον browser άλλων χρηστών
- A5 – Υπερχείλιση Buffer: Το κεφάλαιο αυτό πραγματεύεται επιθέσεις μέσω εισόδων που σχεδιάζονται για να γράψουν πάνω στο σημείο εκτέλεσης του προγράμματος
- A6 – Διαρροές με χρήση Injection: Το κεφάλαιο αυτό πραγματεύεται επιθέσεις μέσω εισόδων που μορφοποιούνται κατάλληλα έτσι ώστε να περιέχουν εκτελέσιμες εντολές.

Μερικοί δικτυακοί τόποι προσπαθούν να προστατευτούν φιλτράροντας όποια κακόβουλη είσοδό τους δοθεί. Το πρόβλημα είναι ότι υπάρχουν τόσοι πολλοί διαφορετικοί τρόποι κωδικοποίησης της πληροφορίας. Αυτές οι μορφές κωδικοποίησης δεν ομοιάζουν με κρυπτογράφηση καθώς είναι πολύ εύκολο να αποκωδικοποιηθούν. Ακόμη οι προγραμματιστές συχνά ξεχνούν να αποκωδικοποιήσουν όλες τις παραμέτρους στην πιο απλή μορφή τους πριν τις χρησιμοποιήσουν. Οι παράμετροι πρέπει να μεταφερθούν στην πιο απλή μορφή πριν πιστοποιηθούν, διαφορετικά κακόβουλη είσοδος μπορεί να διαπεράσει τα φίλτρα. Η διαδικασία της απλοποίησης αυτών των κωδικοποιήσεων αποκαλείται κανονικοποίηση. Καθώς σχεδόν όλες οι HTTP είσοδοι μπορούν να αναπαρασταθούν σε διαφορετικές μορφές η τεχνική αυτή μπορεί να χρησιμοποιηθεί για να αποκρουστεί η οποιαδήποτε επίθεση που στόχο έχει τα προβλήματα ασφάλειας που περιγράφονται στο έγγραφο αυτό. Το γεγονός αυτό κάνει το φιλτράρισμα δύσκολη εργασία.

Ένας εκπληκτικός αριθμός δικτυακών εφαρμογών χρησιμοποιούν μηχανισμούς μόνο από την πλευρά του client για να πιστοποιήσουν την είσοδο. Οι μηχανισμοί πιστοποίησης από τη μεριά του client εύκολα προσπερνώνται αφήνοντας τη δικτυακή εφαρμογή απροστάτευτη απέναντι σε χρήση κακόβουλων παραμέτρων. Οι επιτιθέμενοι μπορούν να δημιουργήσουν τα δικά τους HTTP αιτήματα

χρησιμοποιώντας εργαλεία τόσο απλά όσο το telnet. Έτσι, δεν χρειάζεται να υπερκεράσουν μέτρα ασφαλείας που ο προγραμματιστής έχει πάρει από την πλευρά του client. Σημειώστε ότι η πιστοποίηση από τη μεριά του client είναι μια πολύ καλή ιδέα όσον αφορά την απόδοση και τη χρηστικότητα αλλά παρόλα αυτά δεν έχει κανένα προτέρημα σχετικά με την ασφάλεια. Οι έλεγχοι από την πλευρά του διακομιστή θεωρούνται απαραίτητοι για να αμυνθούμε εναντίον των επιθέσεων μέσω της μεταβολής των παραμέτρων. Από τη στιγμή που είμαστε καλυμμένοι σε αυτό το σημείο ο έλεγχος από τη μεριά του client μπορεί επίσης να συμπεριληφθεί για να αυξήσουμε το ποσοστό ασφάλειας που παρέχουμε και να μειώσουμε το ποσό της άκυρης κίνησης προς τον διακομιστή.

Οι επιθέσεις αυτές αυξάνουν διαρκώς καθώς αυξάνονται τα εργαλεία που υποστηρίζουν την μεταβολή των παραμέτρων. Οι επιπτώσεις από τη χρήση μη πιστοποιημένης εισόδου δεν πρέπει να υποτιμόνται. Ένας μεγάλος αριθμός επιθέσεων θα ήταν δύσκολο να συμβεί, ή ακόμη και αδύνατο, εάν οι προγραμματιστές προσέθεταν τη δυνατότητα να πιστοποιείται η είσοδος πριν αυτή χρησιμοποιηθεί. Εάν μια δικτυακή εφαρμογή δεν έχει ένα δυνατό κεντρικό μηχανισμό για την πιστοποίηση όλων των εισόδων από HTTP αιτήματα, και οποιεσδήποτε άλλες πηγές, είναι πολύ πιθανό να εμφανιστούν και να ασφαλείας κατά τη διάρκεια επιθέσεων βασισμένων σε κακόβουλες εισόδους

A1.2 Περιβάλλοντα που προσβάλλονται

Όλοι οι web server, οι application server, και τα περιβάλλοντα δικτυακών εφαρμογών θεωρούνται υποψήφιοι δέκτες επιθέσεων από μεταβολή παραμέτρων.

A1.3 Παραδείγματα και Παραπομπές

- Οδηγός του OWASP για να χτίσετε ασφαλείς δικτυακές εφαρμογές και υπηρεσίες. Κεφάλαιο 8: Πιστοποίηση δεδομένων:
http://www.owasp.org/documentation/guide/

- modsecurity project (Apache module for HTTP validation)
http://www.modsecurity.org

- Πως να δημιουργήσετε μια μηχανή πιστοποίησης HTTP αιτημάτων (J2EE validation with Stinger)
http://www.owasp.org/columns/jeffwilliams/jeffwilliams2

- Have Your Cake and Eat it Too (.NET validation)
http://www.owasp.org/columns/jpoteet/jpoteet2

A1.4 Πως να αντιληφθείτε αν είστε ευάλωτος σε επιθέσεις

Οποιοδήποτε μέρος ενός HTTP αιτήματος που χρησιμοποιείται από μια δικτυακή εφαρμογή χωρίς να έχει πιστοποιηθεί προσεκτικά αποκαλείται “tainted” παράμετρος. Ο πιο απλός τρόπος για να εντοπίσουμε τέτοιες παραμέτρους είναι να έχουμε μια λεπτομερή αναφορά του κώδικα και να ψάξουμε μέσα σε αυτήν τα σημεία όπου τα δεδομένα προέρχονται από ένα HTTP αίτημα. Για παράδειγμα σε μια J2EE εφαρμογή αυτές είναι οι μέθοδοι στην HttpServletRequest class. Μετά μπορούμε να παρακολουθήσουμε τον κώδικα για να δούμε που χρησιμοποιούνται οι μεταβλητές

αυτές. Εάν η μεταβλητή που δεν ελέγχεται πριν χρησιμοποιηθεί είναι πολύ πιθανό να υπάρξει πρόβλημα. Στην γλώσσα Perl καλό είναι να χρησιμοποιήσουμε την επιλογή "taint" (-T).

Είναι επίσης πιθανό να εντοπίσουμε τη χρήση τέτοιων παραμέτρων *(tainted)* με τη βοήθεια εργαλείων όπως το OWASP WebScarab. Δίνοντας μη αναμενόμενες τιμές σε HTTP αιτήματα και παρατηρώντας τις αντιδράσεις της δικτυακής εφαρμογής στις εισόδους αυτές, μπορούμε να αναγνωρίσουμε σημεία όπου τέτοιες παράμετροι μπορεί να χρησιμοποιούνται.

A1.5 Πως να προστατευτείτε

Ο καλύτερος τρόπος για να προστατευτούμε από επιθέσεις μέσω αλλαγής των παραμέτρων είναι να βεβαιωθούμε ότι όλες οι παράμετροι πιστοποιούνται πριν χρησιμοποιηθούν. Ένα κεντρικό μέρος του προγράμματος (συνάρτηση) ή μια βιβλιοθήκη θα μπορούσαν να είναι πιο αποτελεσματικά καθώς ο κώδικας που διεξάγει τον έλεγχο πρέπει να βρίσκεται καθ' ολοκληρίαν σε ένα σημείο. Κάθε παράμετρος πρέπει να ελέγχεται σύμφωνα με μια αυστηρή μορφή που πρέπει να έχει, η οποία περιγράφει απόλυτα ποια είναι η είσοδος που επιτρέπεται να δοθεί. Αρνητικές προσεγγίσεις που περιλαμβάνουν αποκλεισμό μέσω φίλτρων συγκεκριμένων κακόβουλων εισόδων και οι προσεγγίσεις που στηρίζονται σε ψηφιακές υπογραφές δεν θεωρούνται αποτελεσματικές και ίσως είναι δύσκολο να διατηρηθούν.

Οι παράμετροι θα πρέπει να πιστοποιηθούν σύμφωνα με κάποια θετικά χαρακτηριστικά που ορίζονται από:

- τον τύπο δεδομένων (string, integer, real, etc…)
- τους επιτρεπόμενος χαρακτήρες
- το ελάχιστο και το μέγιστο μήκος
- εάν το κενό *(null)* επιτρέπεται
- εάν η παράμετρος επιβάλλεται να συμπληρωθεί η όχι
- εάν διπλοεγγραφές επιτρέπονται
- το εύρος των αριθμών που χρησιμοποιείται
- συγκεκριμένες δεκτές τιμές
- συγκεκριμένους τύπους δεδομένων

Μια νέα τάξη εργαλείων ασφαλείας γνωστά ως firewalls δικτυακών εφαρμογών μπορούν να παρέχουν υπηρεσίες πιστοποίησης παραμέτρων. Παρόλα αυτά, με στόχο οι συσκευές αυτές να είναι αποτελεσματικές, η συσκευή πρέπει να παραμετροποιηθεί με έναν αυστηρό ορισμό του τι θεωρείται σωστό για μια παράμετρο για τον εκάστοτε δικτυακό τόπο. Αυτό περιλαμβάνει σωστή προστασία από όλους τους τύπους εισόδου σε ένα HTTP αίτημα συμπεριλαμβανομένων των URLs, φορμών εισαγωγής δεδομένων, cookies, συμβολοσειρών ερωτήσεων, κρυμμένων πεδίων και παραμέτρων.

Το project των OWASP φίλτρων παράγει επαναχρησιμοποιούμενα τμήματα σε πολλές γλώσσες για να βοηθήσει στην προστασία από επιθέσεις μέσω τροποποίησης των παραμέτρων. Η μηχανή πιστοποίησης HTTP αιτημάτων Stinger (stinger.sourceforge.net) επίσης αναπτύχθηκε από την OWASP για J2EE περιβάλλοντα.

Α2 Έλεγχος προσβάσεων

Α2.1 Περιγραφή

Ο έλεγχος της πρόσβασης που κάποιες φορές αποκαλείται πιστοποίηση του χρήστη είναι ο τρόπος με τον οποίο μια δικτυακή εφαρμογή δίνει πρόσβαση για κάποιους χρήστες στο περιεχόμενό της και τις λειτουργίες της, και σε κάποιους όχι. Αυτοί οι έλεγχοι πραγματοποιούνται μετά την πιστοποίηση και ορίζουν σε τι ενέργειες επιτρέπεται να προβούν οι πιστοποιημένοι χρήστες. Ο όρος "έλεγχος πρόσβασης" ακούγεται σαν να πρόκειται για πρόβλημα, αλλά περιέργως είναι δύσκολο να το αναφέρουμε πιο σωστά για να γίνει αντιληπτό. Το μοντέλο ελέγχου πρόσβασης μιας δικτυακής εφαρμογής είναι στενά συνδεδεμένο με το περιεχόμενο και τις λειτουργίες που ο δικτυακός τόπος παρέχει. Επιπροσθέτως οι χρήστες μπορεί να ανήκουν σε ένα σύνολο ομάδων η ρόλων με διαφορετικές ικανότητες η δικαιώματα.

Οι προγραμματιστές συχνά υποτιμούν τη δυσκολία δημιουργίας ενός αξιόπιστου μηχανισμού ελέγχου πρόσβασης. Πολλά από αυτά τα σχήματα δεν σχεδιάστηκαν εσκεμμένα αλλά απλά εξελίχθηκαν παράλληλα με τους δικτυακούς τόπους. Στις περιπτώσεις αυτές, οι κανόνες ελέγχου της πρόσβασης εισάγονται σε διάφορες θέσεις μέσα στον κώδικα. Καθώς ο δικτυακός τόπος προστατεύεται από επιθέσεις το σύνολο των κανόνων διευρύνεται τόσο πολύ ώστε είναι σχεδόν αδύνατο αυτοί να γίνουν αντιληπτοί.

Πολλές φορές δεν είναι δύσκολο να ανακαλύψουμε και να σπάσουμε τις αδυναμίες ασφαλείας πολλών τέτοιων σχημάτων ελέγχου. Συχνά το μόνο που χρειάζεται είναι να δημιουργήσουμε ένα έξυπνο αίτημα για συναρτήσεις το περιεχόμενο του οποίου δεν θα έπρεπε να επιτρέπεται. Από τη στιγμή που μια διαρροή ασφαλείας ανακαλύπτεται, τα αποτελέσματα μιας μη επιτρεπόμενης πρόσβασης μπορεί να είναι καταστρεπτικά. Εκτός από την περίπτωση που κάποιος επιτιθέμενος δει περιεχόμενο για το οποίο δεν είναι πιστοποιημένος, ίσως καταφέρει και μπορέσει να αλλάξει ή να αλλάξει μέρος του περιεχομένου, να εκτελέσει συναρτήσεις στις οποίες δεν έχει πρόσβαση η ακόμη και να πάρει τον έλεγχο της διαχείρισης του δικτυακού τόπου.

Ένα συγκεκριμένο είδος προβλήματος ελέγχου πρόσβασης είναι τα διαχειριστικά περιβάλλοντα που επιτρέπουν στους διαχειριστές των δικτυακών τόπων να ελέγχουν το site από το internet. Τέτοια χαρακτηριστικά συχνά χρησιμοποιούνται για να επιτρέπουν στους διαχειριστές του δικτυακού τόπου να διαχειρίζονται επαρκώς τους χρήστες, τα δεδομένα και το περιεχόμενο του site. Σε πολλές περιπτώσεις, τα site υποστηρίζουν μια ποικιλία διαχειριστικών ρόλων για να πετύχουν μια διαβαθμισμένη διαχείριση. Εξαιτίας της δύναμης που έχουν αυτά τα περιβάλλοντα αποτελούν πρωτεύοντες στόχους για επίθεση.

Α2.2 Περιβάλλοντα που επηρεάζονται

Όλοι οι γνωστοί web server, application server, και περιβάλλοντα δικτυακών εφαρμογών είναι οπωσδήποτε τρωτοί σε κάποια από τα προβλήματα αυτά. Ακόμη και αν ένας δικτυακός τόπος είναι απολύτως στατικός, αν δεν έχει παραμετροποιηθεί σωστά, επιτιθέμενοι μπορούν να αποκτήσουν τον έλεγχο σε ευαίσθητα αρχεία και να τον καταστρέψουν ή να τον τροποποιήσουν κακόβουλα.

A2.3 Παραδείγματα και Παραπομπές

- Οδηγός του OWASP για να φτιάξετε ασφαλείς δικτυακές εφαρμογές και υπηρεσίες Κεφάλαιο 8: Έλεγχος πρόσβασης:

http://www.owasp.org/guide/

- Έλεγχος πρόσβασης (aka Authorization) στην J2EE εφαρμογή σας

http://www.owasp.org/columns/jeffwilliams/jeffwilliams3

- Security Architecture - Layered Insecurity - BY Richard Mackey

http://www.infosecuritymag.com/2002/jun/insecurity.shtml

A2.4 Πως διαπιστώνετε αν είστε ευάλωτος σε επιθέσεις

Εικονικά όλοι οι δικτυακοί τόποι απαιτούν την ύπαρξη κάποιων μηχανισμών ελέγχου πρόσβασης. Ωστόσο μια πολιτική ελέγχου πρόσβασης θα πρέπει να είναι σαφώς ορισμένη. Επίσης το έγγραφο συνοδευτικό υλικό *(documentation)* που περιγράφει τον σχεδιασμό του θα πρέπει να περιέχει μια προσέγγιση για να επιβάλει την ύπαρξη μιας πολιτικής. Εάν αυτό το έγγραφο δεν υπάρχει τότε είναι πολύ πιθανό ένας δικτυακός τόπος να είναι ευάλωτος σε επιθέσεις.

Ο κώδικας που υλοποιεί την πολιτική ελέγχου πρόσβασης πρέπει να ελέγχεται. Τέτοιος κώδικας θα πρέπει να είναι καλά δομημένος και ακόμη περισσότερο εντοπισμένος σε ένα συγκεκριμένο σημείο. Μια λεπτομερής ανάλυση του κώδικα θα πρέπει να έχει γίνει για να πιστοποιήσουμε την ορθότητα του μηχανισμού ελέγχου πρόσβασης. Επιπροσθέτως θα ήταν εξαιρετικά χρήσιμο ο έλεγχος πιθανών διασπάσεων του κλοιού ασφάλειας έτσι ώστε να δούμε εάν υπάρχουν κάποια προβλήματα στο σχήμα ελέγχου πρόσβασης.

Βρείτε πως ο δικτυακός τόπος διαχειρίζεται. Ανακαλύψετε πόσες αλλαγές γίνονται στις σελίδες, πως αυτές ελέγχονται και πως μεταφέρονται από και προς τον διακομιστή. Εάν οι διαχειριστές μπορούν να κάνουν αλλαγές από μακριά θα πρέπει να ξέρετε πως προστατεύονται αυτά τα κανάλια επικοινωνίας. Προσεκτικά δείτε κάθε περιβάλλον για να βεβαιωθείτε ότι μόνο πιστοποιημένοι διαχειριστές έχουν πρόσβαση σε αυτά. Επίσης εάν υπάρχουν διαφορετικοί τύποι ομάδων δεδομένων που μπορεί να τεθούν υπό διαχείριση μέσα από διαφορετικά περιβάλλοντα βεβαιωθείτε ότι μόνο τα εξουσιοδοτημένα δεδομένα μπορεί να διαχειριστούν επίσης. Εάν τέτοια περιβάλλοντα ελέγχονται από εξωτερικές εντολές ελέγξτε τη χρήση τέτοιων εντολών για να βεβαιωθείτε ότι δεν γίνονται στόχος επιθέσεων μέσω εμφώλευσης εντολών όπως περιγράφεται στο δοκίμιο αυτό

A2.5 Πως να προστατευτείτε

Το πιο σημαντικό βήμα είναι σκεφθείτε τις απαιτήσεις του ελέγχου πρόσβασης μιας εφαρμογής και να τις συλλέξετε σε μια πολιτική δικτυακής ασφάλειας. Προτείνουμε τη χρήση ενός πίνακα ελέγχου πρόσβασης για να ορίσουμε τους όρους της πρόσβασης. Αν δεν γράψουμε την πολιτική ασφάλειας δεν υπάρχει ορισμός του τι σημαίνει ασφάλεια για το εκάστοτε site. Η πολιτική θα πρέπει να αναφέρει τις ομάδες των χρηστών που μπορούν να πάρουν πρόσβαση στο σύστημα αλλά και τις εντολές και στο είδος του περιεχομένου στο οποίο θα έχουν πρόσβαση οι ομάδες αυτές. Ο

μηχανισμός ελέγχου της πρόσβασης πρέπει να ελεγχθεί εκτενώς για να βεβαιωθούμε ότι δεν υπάρχει τρόπος να το υπερκεράσουμε. Ο έλεγχος αυτός απαιτεί μια ποικιλία λογαριασμών και μεγάλης έκτασης προσπάθειες για να πάρουμε πρόσβαση σε υλικό ή εντολές στα οποία δεν έχουμε πρόσβαση.

Κάποια συγκεκριμένα θέματα που αφορούν έλεγχο πρόσβασης περιλαμβάνουν:

- Μη ασφαλή Id's – Τα περισσότερα sites χρησιμοποιούν μια μορφή id, κλειδιού, ή ευρετηρίου ως έναν τρόπο για να αναφερθούν στους χρήστες, τους ρόλους που αυτοί έχουν, το περιεχόμενο, τα αντικείμενα ή τις εντολές. Αν ένας επιτιθέμενος μπορεί να μαντέψει αυτά τα id's, και οι δοθείσες τιμές δεν πιστοποιούνται για τον τωρινό χρήστη, ο επιτιθέμενος μπορεί να δοκιμάσει ελεύθερα το σχήμα ελέγχου πρόσβασης για να δει σε τι έχουν πρόσβαση οι χρήστες και σε τι όχι. Οι δικτυακές εφαρμογές δεν μπορούν να βασίζονται στη μυστικότητα οιονδήποτε id's για την προστασία τους.

- Forced Browsing Past Access Control Checks – πολλοί δικτυακοί τόποι ζητούν από τους χρήστες να περάσουν διάφορους ελέγχους πριν τους δοθεί πρόσβαση σε συγκεκριμένα URLs που τυπικά θεωρούνται "βαθύτερα" όσον αφορά τη χωροθέτησή τους στο site. Οι έλεγχοι αυτοί δεν πρέπει να προσπερνώνται εύκολα από κάποιον χρήστη που απλά προσπερνά τη σελίδα και τον έλεγχο ασφαλείας της.

- Path Traversal – Το είδος αυτής της επίθεσης προϋποθέτει πληροφορίες σχετικές με το path (πχ, "../../target_dir/target_file") σαν μέρος ενός αιτήματος για πληροφορίες. Τέτοιες επιθέσεις, στόχο έχουν να πάρουν πρόσβαση σε αρχεία που κανονικά δεν είναι προσβάσιμα απευθείας από κάποιον ή σε διαφορετική περίπτωση η πρόσβαση σε αυτά θα απαγορευόταν αν ήταν να ζητηθεί έτσι. Τέτοιες επιθέσεις υποβάλλονται μέσω αιτημάτων σε URLs όπως επίσης και σε οποιαδήποτε άλλη είσοδο.

- File Permissions – Πολλοί web servers και application servers στηρίζονται σε λίστες ελέγχου πρόσβασης που παρέχονται από το σύστημα αρχείων της πλατφόρμας που τρέχει στο background. Ακόμα και αν όλα τα δεδομένα αποθηκεύονται σε άλλους διακομιστές, υπάρχουν πάντα δεδομένα που αποθηκεύονται τοπικά στον web server και τον application server που δεν θα πρέπει να είναι προσβάσιμα σε όλους, ιδιαίτερα τα αρχεία παραμετροποίησης, τα αρχεία του συστήματος και τα scripts που είναι εγκατεστημένα στους περισσότερους web και application servers. Μόνο τα αρχεία που έχει σχεδιαστεί να είναι προσβάσιμα από τους χρήστες του διαδικτύου πρέπει να μαρκαριστούν σαν διαθέσιμα για ανάγνωση, χρησιμοποιώντας τις άδειες που δίνει το λειτουργικό σύστημα. Τα περισσότερα directories δεν πρέπει να είναι αναγνώσιμα και μόνο λίγα αρχεία, αν αυτό είναι απαραίτητο, πρέπει να χαρακτηριστούν ως εκτελέσιμα.

- Client Side Caching – Πολλοί χρήστες έχουν πρόσβαση σε δικτυακές εφαρμογές από υπολογιστές που βρίσκονται σε δημόσιο χώρο προς χρήση του κοινού, όπως αυτοί σε βιβλιοθήκες, σχολεία, αεροδρόμια, και άλλα σημεία πρόσβασης. Οι browser αποθηκεύουν συχνά ιστοσελίδες που μπορούν να προσπελαστούν από επιτιθεμένους για να αποκτήσουν πρόσβαση σε σημεία

δικτυακών τόπων που σε άλλες περιπτώσεις δεν θα είχαν. Οι προγραμματιστές πρέπει να χρησιμοποιήσουν πολλαπλούς μηχανισμούς, συμπεριλαμβανομένων των HTTP headers και των meta tags, για να βεβαιωθούν ότι οι σελίδες που περιέχουν τις ευαίσθητες πληροφορίες δεν αποθηκεύονται από τον browser του χρήστη.

- Υπάρχουν μερικά εργαλεία ασφάλειας του επιπέδου της εφαρμογής που μπορούν να βοηθήσουν στην σωστή επιβολή κάποιων σημείων του σχεδίου ελέγχου πρόσβασής. Πάλι, όπως για την επικύρωση παραμέτρου, για να είναι αποτελεσματικό, το σημείο πρέπει να διαμορφωθεί καθορίζοντας αυστηρά τα αιτήματα θεωρούνται νόμιμα για το εκάστοτε site. Κατά χρησιμοποίηση ενός τέτοιου τμήματος λογισμικού, πρέπει να είστε προσεκτικοί για να καταλάβετε ακριβώς ποια σημεία ελέγχου πρόσβασης μπορεί αυτό να παρέχει για τη δεδομένη πολιτική ασφάλειας του δικτυακού σας τόπου, και ποιο μέρος της πολιτικής ελέγχου πρόσβασής σας δεν μπορεί να ελέγξει, και επομένως πρέπει να εξεταστεί κατάλληλα με δικό σας ειδικά παραμετροποιημένο κώδικα.

Για τις λειτουργίες διαχείρισης, η πρώτη συμβουλή που έχουμε να δώσουμε είναι να μην επιτρέπεται στον διαχειριστή η πρόσβαση στο διαχειριστικό εργαλείο μέσα από την πρώτη σελίδα, εάν αυτό είναι δυνατόν. Λαμβάνοντας υπόψη τη δύναμη αυτών των περιβαλλόντων, οι περισσότεροι οργανισμοί δεν πρέπει εκτεθούν στον κίνδυνο να βγάλουν στον αέρα αυτά τα interface στην πρώτη σελίδα. Εάν η απομακρυσμένη πρόσβαση διαχειριστών είναι απολύτως απαραίτητη, αυτό μπορεί να επιτευχθεί χωρίς να ανοίξουμε την πρόσβαση από την πρώτη σελίδα. Η χρήση της τεχνολογίας VPN θα μπορούσε να χρησιμοποιηθεί για να παρέχει μια εξωτερική πρόσβαση του διαχειριστή στο εσωτερικό δίκτυο επιχείρησης (ή του δικτυακού τόπου) από το οποίο ένας διαχειριστής μπορεί έπειτα να έχει πρόσβαση στο site μέσω μιας προστατευμένης σύνδεσης.

Α3 Παράνομη πιστοποίηση και διαχείριση των συνδέσεων

Α3.1 Περιγραφή

Η πιστοποίηση και η διαχείριση των συνδέσεων περιλαμβάνει όλες τις μορφές χειρισμού πιστοποίησης του χρήστη και διαχείρισης ενεργών συνδέσεων. Η πιστοποίηση είναι μια κρίσιμη πλευρά αυτής της διαδικασίας, αλλά ακόμη και αξιόπιστοι μηχανισμοί πιστοποίησης είναι δυνατό να υπονομευθούν από λειτουργίες διαχείρισης πιστοποιητικών με διαρροές ασφάλειας, συμπεριλαμβανομένης της password change, forgot my password, remember my password, account update, και άλλων σχετικών λειτουργιών. Επειδή οι επιθέσεις "walk by" είναι πιθανές για πολλές δικτυακές εφαρμογές, όλες οι λειτουργίες διαχείρισης λογαριασμών θα πρέπει να ζητούν επαναπιστοποίηση, ακόμη και αν ο χρήστης έχει έγκυρο id σύνδεσης.

Η πιστοποίηση χρήστη στο διαδίκτυο εμπεριέχει συνήθως τη χρήση ενός user id και password. Ισχυρότερες μέθοδοι πιστοποίησης είναι διαθέσιμες στο εμπόριο, όπως κρυπτογραφικές συσκευές ή βιομετρικά στοιχεία βασισμένα σε software και hardware, αλλά τέτοιοι μηχανισμοί είναι απαγορευτικοί λόγω κόστους για τις περισσότερες δικτυακές εφαρμογές. Ένα ευρύ σύνολο διαρροών ασφάλειας λογαριασμών και διαχείρισης συνδέσεων μπορεί να έχει ως αποτέλεσμα να τεθούν σε κίνδυνο (λογαριασμοί χρηστών ή διαχείρισης συστημάτων). Οι προγραμματιστών συχνά υποτιμούν την πολυπλοκότητα του σχεδιασμού ενός σχεδίου πιστοποίησης και διαχείρισης συνδέσεων, το οποίο να προστατεύει επαρκώς τα πιστοποιητικά σε όλες τις εκφάνσεις του site. Οι δικτυακές εφαρμογές πρέπει να εγκαθιστούν συνδέσεις, ώστε να εντοπίζουν το ρεύμα των αιτημάτων από κάθε χρήστη. Το HTTP δεν παρέχει αυτή τη δυνατότητα, έτσι οι δικτυακές εφαρμογές πρέπει να τη δημιουργούν οι ίδιες. Συχνά, το περιβάλλον των δικτυακών εφαρμογών παρέχει δυνατότητα συνδέσεων, αλλά πολλοί προγραμματιστές προτιμούν να δημιουργούν τις δικές τους συνδέσεις από την αρχή. Σε κάθε περίπτωση, αν οι συνδέσεις δεν προστατεύονται επαρκώς, ένας επιτιθέμενος μπορεί να κάνει πάρει τον έλεγχο μιας ενεργής σύνδεσης και να αποσπάσει το user id. Η δημιουργία ενός σχεδίου ώστε να δημιουργηθούν ισχυρές συνδέσεις και να προστατευτούν κατά τη διάρκεια του κύκλου ζωής τους έχει αποδειχθεί απατηλή για πολλούς προγραμματιστές.

Εκτός και αν όλα τα πιστοποιητικά και οι συνδέσεις προστατεύονται με SSL συνεχώς από την αποκάλυψή τους από άλλες διαρροές ασφάλειας, όπως cross site scripting, ένας επιτιθέμενος μπορεί να κάνει πειρατεία στις συνδέσεις ενός χρήστη και να αποσπάσει την ταυτότητά τους.

Α3.2 Περιβάλλοντα που επηρεάζονται

Όλοι οι γνωστοί web servers, application servers και περιβάλλοντα δικτυακών εφαρμογών είναι τρωτοί σε θέματα παράνομης πιστοποίησης και διαχείρισης συνδέσεων.

A3.3 Παραδείγματα και Παραπομπές

- Οδηγός του OWASP για να Φτιάξετε Ασφαλείς Δικτυακές Εφαρμογές και Υπηρεσίες, Κεφάλαιο 6: Πιστοποίηση και Κεφάλαιο 7: Διαχείριση Συνδέσεων:

http://www.owasp.org/guide/

- Λευκή Βίβλος για τα Προβλήματα Ασφαλείας ... Συνδέσεων σε Δικτυακές Εφαρμογές

http://www.acros.si/papers/session_fixation.pdf

- Λευκή Βιβλός για την Ανακτηση Password για Δικτυακές Εφαρμογές

http://fishbowl.pastiche.org/archives/docs/PasswordRecovery.pdf

A3.4 Πως να προσδιορίσετε αν έχετε Προβλήματα Ασφαλείας

Τόσο ο λεπτομερής έλεγχος του κώδικα όσο και οι δοκιμές παράνομης πρόσβασης μπορούν να χρησιμοποιηθούν για τη διάγνωση προβλημάτων πιστοποίησης διαχείρισης συνδέσεων. Ελέγξτε προσεκτικά κάθε πλευρά των μηχανισμών πιστοποίησης, ώστε να διασφαλίσετε ότι τα πιστοποιητικά χρήστη προστατεύονται συνεχώς, ενώ βρίσκονται σε παύση (π.χ. σε δίσκο) και ενώ βρίσκονται σε μετάβαση (π.χ. κατά τη διάρκεια login). Ελέγξτε κάθε διαθέσιμο μηχανισμό για την αλλαγή συνθηματικού του χρήστη για να διασφαλίσετε ότι μόνο ένας εξουσιοδοτημένος χρήστης μπορεί να τα αλλάξει. Ελέγξτε τον μηχανισμό διαχείρισης συνδέσεων για να διασφαλίσετε ότι οι ταυτοποιητές συνδέσεων προστατεύονται πάντα και χρησιμοποιούνται με τέτοιο τρόπο ώστε να ελαχιστοποιούν την πιθανότητα τυχαίας ή εχθρικής έκθεσης.

A3.5 Πως να προστατευτείτε

Προσεκτική και κατάλληλη χρήση των συνηθισμένων μηχανισμών πιστοποίησης και διαχείρισης συνδέσεων θα πρέπει να μειώνουν σημαντικά την πιθανότητα προβλήματος σ' αυτήν την περιοχή. Ένα καλό πρώτο βήμα είναι να προσδιορίζεις και να τεκμηριώνεις την πολιτική του site σου όσον αφορά την ασφαλή διαχείριση των πιστοποιητικών των χρηστών. Το να διασφαλίζεις ότι η εφαρμογή σου επιβάλλει αυτήν την πολιτική με συνέπεια είναι το κλειδί για έναν ασφαλή και εύρωστο μηχανισμό πιστοποίησης και διαχείρισης συνδέσεων. Μερικές κρίσιμες περιοχές περιλαμβάνουν:

- Αντοχή του password – Τα passwords θα πρέπει να έχουν περιορισμούς που απαιτούν ένα ελάχιστο μέγεθος και πολυπλοκότητα για το password. Η πολυπλοκότητα απαιτεί συνήθως τη χρήση ελάχιστων συνδυασμών αλφαβητικών, αριθμητικών και/ή μη-αλφααριθμητικών χαρακτήρων στο password ενός χρήστη (π.χ. τουλάχιστον ένα από το καθένα). Θα πρέπει να ζητείται από τους χρήστες να αλλάζουν το password τους κατά διαστήματα. Οι χρήστες θα πρέπει να αποτρέπονται από το να ξαναχρησιμοποιούν προηγούμενα passwords.
- Χρήση password – Οι χρήστες θα πρέπει να περιορίζονται σε έναν προσδιορισμένο αριθμό αποπειρών login ανά μονάδα χρόνου και επανειλημμένες αποτυχημένες απόπειρες login θα πρέπει να καταχωρούνται. Τα passwords που δίνονται κατά τη διάρκεια των αποτυχημένων αποπειρών login δεν θα πρέπει να καταγράφονται, καθώς αυτό μπορεί να εκθέσει το password του χρήστη σε οποιονδήποτε μπορεί να αποκτήσει πρόσβαση σε

αυτήν την καταχώρηση. Το σύστημα δεν θα πρέπει να υποδεικνύει αν ήταν το username ή το password αυτό που ήταν λάθος αν μια απόπειρα login αποτύχει. Οι χρήστες θα πρέπει να ενημερώνονται για την ημερομηνία/ώρα του τελευταίου επιτυχημένου login και τον αριθμό των αποτυχημένων αποπειρών πρόσβασης στο λογαριασμό τους έκτοτε.

- Έλεγχοι Αλλαγής Password: Ένας μόνο μηχανισμός αλλαγής password πρέπει να χρησιμοποιείται οπουδήποτε επιτρέπεται στους χρήστες να αλλάζουν password, ανεξάρτητη από την κατάσταση. Θα πρέπει να ζητείται πάντα από τους χρήστες να παρέχουν και το παλιό και το νέο password τους όταν αλλάζουν password (όπως και όλες τις πληροφορίες του λογαριασμού). Αν τα ξεχασμένα passwords αποστέλλονται στους χρήστες με e-mail, το σύστημα θα πρέπει να ζητά από το χρήστη να επαναπιστοποιήσει κάθε φορά που αλλάζει διεύθυνση e-mail, αλλιώς ένας επιτιθέμενος που έχει πρόσβαση στη σύνδεσή τους προσωρινά (π.χ. by walking up στον υπολογιστή τους όταν είναι logged in) μπορεί απλά να αλλάξει την διεύθυνσή τους και να ζητήσει να τους αποσταλεί ένα ξεχασμένο password.
- Αποθήκευση password – Όλα τα passwords πρέπει να αποθηκεύονται είτε σε hashed είτε σε κωδικοποιημένη μορφή ώστε να προστατεύονται από την έκθεση, ανεξαρτήτως του πού είναι αποθηκευμένα. Η hashed μορφή προτιμάται, αφού δεν είναι δυνατόν να αντιστραφεί έτσι ώστε να προκύψει η αρχική. Η κωδικοποίηση θα πρέπει να χρησιμοποιείται όταν χρειάζεται το plaintext password, όπως όταν χρησιμοποιούμε το password για να κάνουμε login σε ένα άλλο σύστημα. Τα passwords δεν θα πρέπει ποτέ να γίνονται hardcoded σε οποιοδήποτε source code. Τα κλειδιά αποκωδικοποίησης πρέπει να προστατεύονται σθεναρά για να διασφαλίζουμε ότι δεν μπορεί να αρπαχτούν και να χρησιμοποιηθούν για να αποκωδικοποιηθεί το password file.
- Προστατεύοντας Πιστοποιητικά σε Μετάβαση – Η μόνη αποτελεσματική τεχνική είναι να κωδικοποιήσουμε ολόκληρη τη συναλλαγή login χρησιμοποιώντας κάτι σαν το SSL. Απλές μεταμορφώσεις του password, όπως το hashing στον πελάτη πριν την μετάδοση παρέχουν μικρή προστασία καθώς η hashed εκδοχή μπορεί απλά να υποκλαπεί και να μεταδοθεί ακόμη και αν το ίδιο το plaintext password δεν μπορεί να γίνει γνωστό.
- Προστασία ID σύνδεσης – Ιδανικά, μια ολόκληρη σύνδεση ενός χρήστη θα πρέπει να προστατεύεται μέσω SSL. Αν αυτό έχει γίνει, τότε το ID σύνδεσης (π.χ. το session cookie) δεν μπορεί να υφαρπαχθεί από το δίκτυο, κάτι που αποτελεί το μεγαλύτερο κίνδυνο έκθεσης για ένα ID σύνδεσης. Αν το SSL δεν είναι εφικτό να πραγματοποιηθεί, τότε τα ίδια τα IDs σύνδεσης πρέπει να προστατευτούν με άλλους τρόπους. Πρώτα, δεν θα πρέπει ποτέ να περιλαμβάνονται στο URL, καθώς μπορούν να κρυφτούν από το browser, να σταλούν στο referrer header ή να προωθηθούν τυχαία σε κάποιον «φίλο». Οι IDs σύνδεσης θα πρέπει να είναι μεγάλοι, πολύπλοκοι, τυχαίοι αριθμοί που δεν μπορούν εύκολα να υποτεθούν. Μπορούν, επίσης, να αλλάζονται συχνά κατά τη διάρκεια μιας σύνδεσης, ώστε να μειωθεί η διάρκεια κατά την οποία είναι έγκυροι. Θα πρέπει να αλλάζονται όταν γυρνάμε σε SSL, πιστοποιούμε ή γίνεται κάποια άλλη μεγάλη μετάβαση. Οι IDs σύνδεσης που επιλέγονται από ένα χρήστη δεν θα πρέπει ποτέ να γίνονται αποδεκτοί.
- Λίστες Λογαριασμών – Τα συστήματα θα πρέπει να σχεδιάζονται, ώστε να αποφεύγουν το να επιτρέπουν στους χρήστες να αποκτούν πρόσβαση σε μια λίστα των ονομάτων του λογαριασμού στο site. Αν οι λίστες των χρηστών

πρέπει να παρουσιαστούν, συνιστάται να δίνεται αντίθετα κάποια μορφή ψευδωνύμου (screen name) που maps στον συγκεκριμένο λογαριασμό. Με αυτόν τον τρόπο, το ψευδώνυμο δεν μπορεί να χρησιμοποιηθεί κατά τη διάρκεια μιας απόπειρας login ή κάποια άλλη λαθραία διείσδυση που βάλλει κατά ενός λογαριασμού χρήστη.

- Browser Caching – Δεδομένα πιστοποίησης και σύνδεσης δεν θα πρέπει ποτέ να υποβάλλονται σαν μέρος ενός GET, απεναντίας θα πρέπει πάντα να χρησιμοποιείται POST. Οι σελίδες πιστοποίησης θα πρέπει να σημειώνονται με όλες τις ποικιλίες του no cache tag ώστε να εμποδίζεται κάποιος απ' το να χρησιμοποιεί το πλήκτρο πίσω στο browser ενός χρήστη για να επανέλθει στη σελίδα login και να επαναυποβάλει τα προηγουμένως πληκτρολογημένα πιστοποιητικά. Πολλοί browsers υποστηρίζουν τώρα το autocomplete=false σημείο για να αποτρέψουν την αποθήκευση πιστοποιητικών σε autocomplete caches.
- Σχέσεις Εμπιστοσύνης – Η αρχιτεκτονική του site σας θα πρέπει να αποφεύγει την ανεπιφύλακτη εμπιστοσύνη μεταξύ συστατικών όποτε αυτό είναι δυνατό. Κάθε συστατικό θα πρέπει να πιστοποιείται προς κάθε άλλο συστατικό με το οποίο αλληλεπιδρά, εκτός κι αν υπάρχει ισχυρός λόγος να μην το κάνει (όπως απόδοση ή έλλειψη χρησιμοποιήσιμου μηχανισμού). Αν απαιτούνται σχέσεις εμπιστοσύνης, ισχυροί διαδικαστικοί και αρχιτεκτονικοί μηχανισμοί θα πρέπει να βρίσκονται σε θέση να διασφαλίζουν ότι δεν μπορεί να γίνεται κατάχρηση αυτής της εμπιστοσύνης καθώς η αρχιτεκτονική του site εξελίσσεται με το χρόνο.

A4 Διαρροές ασφαλείας από Cross-Site Scripting (XSS)

A4.1 Περιγραφή

Οι επιθέσεις μέσω Cross-site scripting (μερικές φορές καλούμενες XSS) εμφανίζονται όταν ένας επιτιθέμενος χρησιμοποιεί μια δικτυακή εφαρμογή για να στείλει τον κακόβουλο κώδικα, στο πεδίο συμπλήρωσης ενός script, σε έναν διαφορετικό χρήστη. Αυτές οι διαρροές ασφαλείας είναι αρκετά διαδεδομένες και εμφανίζονται οπουδήποτε μια δικτυακή εφαρμογή χρησιμοποιεί την είσοδο που δίνει ένας χρήστης χωρίς να ελέγξει την εγκυρότητά της.

Ένας επιτιθέμενος μπορεί να χρησιμοποιήσει το cross site scripting για να στείλει ένα κακόβουλο script σε κάποιον χρήστη. Ο browser του χρήστη δεν έχει κανέναν τρόπο να ξέρει ότι το script δεν πρέπει να το εμπιστευθεί, και θα το εκτελέσει. Επειδή σκέφτεται ότι το script προήλθε από μια εμπιστευτική πηγή, το κακόβουλο script μπορεί να έχει πρόσβαση σε οποιαδήποτε cookies, στα session tokens, ή άλλες ευαίσθητες πληροφορίες που αποθηκεύονται από τον browser και που χρησιμοποιούνται σε εκείνο το site. Αυτά τα script μπορούν ακόμη και να ξαναγράψουν το περιεχόμενο της σελίδας HTML.

Οι επιθέσεις XSS μπορούν γενικά να ταξινομηθούν σε δύο κατηγορίες: Στις stored και στις reflected. Οι stored επιθέσεις είναι εκείνες όπου ο injected κώδικας αποθηκεύεται μόνιμα στους διακομιστές που αποτελούν στόχο, π.χ. σε μια βάση δεδομένων, σε ένα φόρουμ μηνυμάτων, ένα log file επισκέψεων, έναν πεδίο για σχόλια, κλπ. Το θύμα ανακτά έπειτα το κακόβουλο script από τον διακομιστή όταν ζητά τις αποθηκευμένες πληροφορίες. Οι reflected επιθέσεις είναι εκείνες όπου ο injected κώδικας απεικονίζεται από τον web server, όπως σε ένα μήνυμα λάθους, το αποτέλεσμα αναζήτησης, ή οποιαδήποτε άλλη απάντηση που περιλαμβάνει κάποιο μέρος ή ολόκληρη την είσοδο που στέλνεται στον διακομιστή ως τμήμα του αιτήματος. Οι reflected επιθέσεις γίνονται στα θύματα μέσω μιας άλλης διαδρομής, όπως σε ένα e-mail, ή σε κάποιο άλλο server του δικτύου. Όταν ένας χρήστης πέφτει θύμα απάτης κάνοντας κλικ σε κάποιο κακόβουλο link ή υποβάλλει στοιχεία σε κάποια ειδικά επεξεργασμένη φόρμα, ο injected κώδικας περνάει στον web server που αποτελεί στόχο, ο οποίος στέλνει την επίθεση πίσω στον browser του χρήστη. Ο browser εκτελεί έπειτα τον κώδικα επειδή πιστεύει ότι προήλθε από έναν εμπιστευτικό server.

Η συνέπεια μιας επίθεσης XSS είναι η ίδια ανεξάρτητα από εάν πρόκειται για stored ή reflected επίθεση. Η διαφορά είναι στο πώς η ζημιά που προκαλείται φθάνει στον διακομιστή. Μην επαναπαύεστε με τη λογική ότι κάποιο site είναι "read only" ή "brochureware" και συνεπώς δεν είναι τρωτό σε σοβαρές reflected επιθέσεις XSS. Το XSS μπορεί να προκαλέσει ποικίλα προβλήματα για τον χρήστη που κυμαίνονται από μια απλή ενόχληση έως το να χάσει τον ίδιο τον λογαριασμό του. Οι πιο μεγάλες επιθέσεις XSS μπορούν να προκαλέσουν τη δημοσίευση του session cookie του χρήστη, που επιτρέπει σε έναν επιτιθέμενο για να καταλάβει τη σύνδεση του χρήστη και μαζί με αυτή και τον λογαριασμό του. Άλλες καταστρεπτικές επιθέσεις περιλαμβάνουν την δημοσίευση των αρχείων των χρηστών, εγκατάσταση Trojan horses, που πηγαίνει αυτόματα τον χρήστη σε κάποια άλλη σελίδα (redirect), ή που

τροποποιεί το περιεχόμενο του δικτυακού τόπου. Μια διαρροή ασφαλείας, ευάλωτη σε XSS, που επιτρέπει σε έναν επιτιθέμενο να τροποποιήσει ένα δελτίο τύπου ή κάποιο site με ειδήσεις θα μπορούσε να έχει επιπτώσεις στην εικόνα μιας επιχείρησης και να μειώσει την εμπιστοσύνη των πελατών της σε αυτήν. Μια διαρροή ασφαλείας, ευάλωτη σε XSS, σε μια φαρμακευτική περιοχή θα μπορούσε να επιτρέψει σε έναν επιτιθέμενο να τροποποιήσει τις πληροφορίες της δόσης του φαρμάκου με συνέπεια να δοθεί υπερβολική δόση σε κάποιον ασθενή.

Οι επιτιθέμενοι χρησιμοποιούν συχνά διάφορες μεθόδους για να κωδικοποιήσουν το κακόβουλο τμήμα του tag, όπως η χρήση Unicode, έτσι ώστε το αίτημα να φαίνεται λιγότερο ύποπτο στον χρήστη. Υπάρχουν εκατοντάδες παραλλαγές αυτών των επιθέσεων, συμπεριλαμβανομένων εκδόσεων αυτών που δεν απαιτούν σύμβολα του τύπου < >. Για αυτόν τον λόγο, το να προσπαθήσουμε να φιλτράρουμε αυτά τα scripts δεν σημαίνει ότι σίγουρα θα πετύχει σαν μέθοδος. Αντί κάτι τέτοιου προτείνουμε να τσεκάρουμε την εγκυρότητα της εισόδου με κάποια πολύ αυστηρά κριτήρια που σχετίζονται με το είδος της εισόδου που θα έπρεπε να αναμένουμε. Οι επιθέσεις με XSS παρουσιάζονται συνεχώς με τη μορφή εμφωλευμένου JavaScript κώδικα. Παρόλα αυτά, οποιοδήποτε ενσωματωμένο active content είναι μια πιθανή πηγή κινδύνου, που περιλαμβάνει: ActiveX (OLE), VBscript, shockwave, Flash και περισσότερα.

Τα προβλήματα που προκύπτουν από XSS επιθέσεις μπορούν επίσης να δημιουργηθούν σε web και application servers. Οι περισσότεροι web και application servers παράγουν απλές ιστοσελίδες που βγάζουν κάποια μηνύματα στην περίπτωση λάθους, όπως μια σελίδα 404 ‘page not found’ ή ένα 500 ‘internal server error’. Εάν αυτές οι σελίδες περιέχουν οποιεσδήποτε πληροφορίες από το αίτημα του χρήστη, όπως το URL στο οποίο προσπαθούσαν να έχουν πρόσβαση, μπορούν να είναι ευάλωτες σε μια reflected επίθεση XSS.

Η πιθανότητα κάποιο site να είναι ευάλωτο σε XSS είναι εξαιρετικά υψηλή. Υπάρχουν πολλοί τρόποι να προσβληθούν από τέτοιες επιθέσεις δικτυακές εφαρμογές στέλνοντάς τους κακόβουλα scripts. Οι προγραμματιστές, προσπαθώντας να φιλτράρουν κακόβουλα τμήματα αυτών των αιτημάτων είναι πολύ πιθανό να αγνοήσουν άλλες πιθανές επιθέσεις ή κωδικοποιήσεις. Η εύρεση αυτών των διαρροών ασφαλείας δεν είναι παρά πολύ δύσκολη για τους επιτιθεμένους, καθώς το μόνο που χρειάζονται είναι ένας browser και λίγος χρόνος. Υπάρχουν πολλά δωρεάν εργαλεία διαθέσιμα στο δίκτυο που βοηθούν τους επιτιθέμενους να βρουν αυτές τις διαρροές καθώς επίσης και τον τρόπο προσεκτικά που θα μπορέσουν να κάνουν τις επιθέσεις XSS σε κάποιο site στόχο.

Α4.2 Περιβάλλοντα που επηρεάζονται

Όλοι οι web servers, application servers, και περιβάλλοντα δικτυακών εφαρμογών είναι τρωτά σε επιθέσεις μέσω cross site scripting.

Α4.3 Παραδείγματα και Παραπομπές

- To Cross Site Scripting FAQ:

http://www.cgisecurity.com/articles/xss-faq.shtml

- CERT Advisory on Malicious HTML Tags:

http://www.cert.org/advisories/CA-2000-02.html

- CERT "Understanding Malicious Content Mitigation"

http://www.cert.org/tech_tips/malicious_code_mitigation.html

- Cross-Site Scripting Security Exposure Executive Summary:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/topics/ExSumCS.asp

- Understanding the cause and effect of CSS Vulnerabilities:

http://www.technicalinfo.net/papers/CSS.html

- OWASP Guide to Building Secure Web Applications and Web Services, Chapter 8: Data Validation

http://www.owasp.org/documentation/guide/

- How to Build an HTTP Request Validation Engine (J2EE validation with Stinger)

http://www.owasp.org/columns/jeffwilliams/jeffwilliams2

- Have Your Cake and Eat it Too (.NET validation)

http://www.owasp.org/columns/jpoteet/jpoteet2

Α4.4 Πως να διαπιστώσετε αν είστε ευάλωτος σε τέτοιες επιθέσεις

Οι διαρροές ασφαλείας λόγω XSS μπορούν να είναι δύσκολο να εντοπιστούν και να αφαιρεθούν από μια δικτυακή εφαρμογή. Ο καλύτερος τρόπος να βρεθούν είναι να δημιουργηθεί μια αναφορά ασφάλειας του κώδικα και να αναζητηθούν όλες οι θέσεις όπου δίνεται είσοδος από ένα HTTP αίτημα από το οποίο μπορεί να εξαχθεί ένα αποτέλεσμα σε HTML. Σημειώστε ότι διάφορα HTML tags μπορούν να χρησιμοποιηθούν για να περάσουν ένα κακόβουλο JavaScript. Το Nessus, Nikto, και μερικά άλλα διαθέσιμα εργαλεία μπορούν να βοηθήσουν στην εξερεύνηση ενός site για τέτοιου είδους κενά ασφαλείας αλλά μόνο σε επιφανειακό επίπεδο. Εάν ένα σημείο του site είναι τρωτό, υπάρχει υψηλή πιθανότητα ότι υπάρχουν κι άλλα προβλήματα.

Α4.5 Πως να προστατευτείτε

Ο καλύτερος τρόπος για να προστατεύσουμε μια δικτυακή εφαρμογή από τις επιθέσεις XSS είναι να βεβαιωθούμε ότι η εφαρμογή επικυρώνει όλα τα headers, cookies, query strings, form fields και hidden fields (πχ. όλες τις παραμέτρους) σύμφωνα με αυστηρές προδιατυπωμένες προδιαγραφές. Η επικύρωση δεν πρέπει να προσπαθήσει να προσδιορίσει το active content και να το αφαιρέσει, να το φιλτράρει ή να το αποστειρώσει. Υπάρχουν πάρα πολλοί τύποι active content και πάρα πολλοί τρόποι για να κωδικοποιηθεί για να έχουμε τη δυνατότητα να δημιουργήσουμε φίλτρα για αυτά. Προτείνουμε μια θετική πολιτική ασφάλειας που διευκρινίζει τι επιτρέπεται. Οι αρνητικές ή οι πολιτικές που βασίζονται σε ψηφιακές υπογραφές είναι δύσκολο να επικρατήσουν και πιθανόν να αποδειχθούν μη αποτελεσματικές.

Η μέθοδος της κωδικοποίησης της εισόδου που δίνει ο χρήστης μπορεί επίσης να νικήσει τις διαρροές ασφαλείας λόγω του XSS εμποδίζοντας τη μετάδοση των scripts

σε εκτελέσιμη μορφή προς τους χρήστες. Μπορούμε να προστατεύσουμε τις εφαρμογές από javascript επιθέσεις με τη μετατροπή των ακόλουθων χαρακτήρων στην έξοδο, στην αντίστοιχη HTML γλώσσα:

Αντί	Χρησιμοποιούμε
<	&lt;
>	&gt;
(	&#40;
)	&#41;
#	&#35;
&	&#38;

Το OWASP Filters project παράγει επαναχρησιμοποιήσιμα τμήματα σε διάφορες γλώσσες για να αποτρέψει πολλές μορφές επίθεσης μέσω μεταβολής των παραμέτρων, συμπεριλαμβανομένων των επιθέσεων XSS. Η OWASP έχει βγάλει επίσης το CodeSeeker, μια εφαρμογή που ανήκει στο είδος των firewall. Επιπλέον, το επιμορφωτικό πρόγραμμα OWASP WebGoat έχει μαθήματα πάνω στο Cross Site Scripting και την κωδικοποίηση στοιχείων.

A5 Υπερχείλιση Buffer

A5.1 Περιγραφή

Οι επιτιθέμενοι χρησιμοποιούν την τεχνική της υπερχείλισης των buffers για να παραποιήσουν το σωρό στον οποίο εκτελείται μια δικτυακή εφαρμογή. Στέλνοντας μια προσεκτικά παραποιημένη είσοδο στη δικτυακή εφαρμογή, ο επιτιθέμενος μπορεί να κάνει την εφαρμογή να εκτελέσει ένα κομμάτι κώδικα – πετυχαίνοντας έτσι να πάρει τον έλεγχο του μηχανήματος. Οι υπερχειλίσεις των buffers δεν είναι εύκολο να ανακαλυφθούν, αλλά ακόμα και αν ανακαλυφθεί κάποια, είναι εξαιρετικά δύσκολο να την εκμεταλλευτεί κανείς προς όφελός του. Ωστόσο, οι επιτιθέμενοι έχουν κατορθώσει να εντοπίσουν τις υπερχειλίσεις buffer σε μια σειρά τμημάτων. Μια άλλη πολύ παρόμοια κατηγορία διαρροών ασφαλείας είναι γνωστή ως format string attacks.

Οι διαρροές ασφαλείας λόγω υπερχείλισης των buffers μπορεί να παρουσιαστούν και σε web και application servers που εξυπηρετούν τα στατικά και δυναμικά τμήματα ενός δικτυακού τόπου. Οι υπερχειλίσεις των buffer που βρίσκονται στους πιο συχνά χρησιμοποιημένους server είναι πιθανό να γίνουν ευρέως γνωστές και μπορούν να θέσουν έναν σημαντικό κίνδυνο για τους χρήστες αυτών των προϊόντων. Όταν οι δικτυακές εφαρμογές χρησιμοποιούν βιβλιοθήκες, όπως μια βιβλιοθήκη γραφικών για να παραστήσουν κάποιες εικόνες, εκτίθενται σε πιθανές επιθέσεις υπερχείλισης buffer.

Οι υπερχειλίσεις buffer μπορούν επίσης να εντοπιστούν σε συνηθισμένο κώδικα δικτυακών εφαρμογών. Διαρροές ασφαλείας λόγω υπερχείλισης του buffer σε διάφορες δικτυακές εφαρμογές είναι λιγότερο πιθανό να εντοπιστούν καθώς είναι πολύ λίγοι εκείνοι οι επιτιθέμενοι που θα προσπαθήσουν να βρουν και να εκμεταλλευτούν τέτοια κενά ασφάλειας. Εάν ανακαλυφθεί μια τέτοια σε κάποια εφαρμογή, η δυνατότητα να χρησιμοποιηθεί αυτή κακόβουλα (εκτός από το να crashάρει την εφαρμογή) μειώνεται σημαντικά από το γεγονός ότι ο κώδικας πηγής και τα λεπτομερή μηνύματα λάθους για την εφαρμογή δεν είναι κανονικά διαθέσιμα στον επιτιθέμενο.

A5.2 Περιβάλλοντα που επηρεάζονται

Σχεδόν οι περισσότεροι γνωστοί web servers, application servers, και περιβάλλοντα δικτυακών εφαρμογών είναι τρωτά σε buffer υπερχείλιση. Η αξιοσημείωτη εξαίρεση είναι τα περιβάλλοντα σε Java και J2EE, που δεν προσβάλλονται από τέτοιου είδους επιθέσεις (εκτός από περίπτωση υπερχείλισης της ίδιας της JVM).

A5.3 Παραδείγματα και παραπομπές

- OWASP Guide to Building Secure Web Applications and Web Services, Chapter 8: Data Validation

http://www.owasp.org/documentation/guide/

- Aleph One, “Smashing the Stack for Fun and Profit”,

http://www.phrack.com/show.php?p=49&a=14

- Mark Donaldson, “Inside the Buffer Overflow Attack: Mechanism, Method, & Prevention”,

http://rr.sans.org/code/inside_buffer.php

Α5.4 Πώς να εξακριβώσετε αν έχετε προβλήματα ασφάλειας

Για server και βιβλιοθήκες, κρατηθείτε ενήμεροι για τις τελευταίες αναφορές ιών για τα προϊόντα που χρησιμοποιείτε. Για συνηθισμένες εφαρμογές λογισμικού, κάθε κώδικας που δέχεται εισροές από χρήστες μέσω αιτήματος HTTP πρέπει να επιθεωρείται για να διασφαλίζεται ότι μπορεί να χειριστεί καταλλήλως αυθαίρετα μεγάλες εισροές.

Α5.5 Πώς να προστατευτείτε

Κρατηθείτε ενήμεροι για τις τελευταίες αναφορές ιών για τα προϊόντα σας δικτύου και εφαρμογών server και άλλα προϊόντα στην δικτυακή σας υποδομή. Εφαρμόστε τα τελευταία μέτρα προστασίας σε αυτά τα προϊόντα. Κατά διαστήματα, σκανάρετε το site σας με ένα ή περισσότερα από τα κοινά διαθέσιμα scanner που ψάχνουν για διαρροές ασφάλειας buffer overflow στα προϊόντα του server σας και τις συνήθεις δικτυακές σας εφαρμογές.

Για το συνήθη κώδικα εφαρμογής σας, πρέπει να επιθεωρήσετε κάθε κώδικα που δέχεται εισροές από χρήστες μέσω αιτήματος HTTP και να σιγουρέψετε ότι παρέχει κατάλληλο έλεγχο μεγέθους σε όλες τις παρόμοιες εισροές. Αυτό θα πρέπει να γίνει ακόμη και για περιβάλλοντα που δεν είναι τρωτά σε τέτοιες επιθέσεις καθώς υπερβολικά μεγάλες εισροές που δεν πιάνονται μπορεί να προκαλέσουν παρόλα αυτά απόρριψη παροχής υπηρεσίας ή άλλα λειτουργικά προβλήματα.

Α6 Διαρροές ασφάλειας μέσω εμφώλευσης εντολών – Injection flaws

Α6.1 Περιγραφή

Οι διαρροές ασφάλειας μέσω injection flaws επιτρέπουν στους επιτιθεμένους να στείλουν κακόβουλο κώδικα μέσω μιας δικτυακής εφαρμογής σε ένα άλλο σύστημα. Αυτές οι επιθέσεις περιλαμβάνουν κλήσεις στο λειτουργικό σύστημα μέσω των system calls, τη χρήση εξωτερικών προγραμμάτων μέσω των εντολών του shell, καθώς επίσης και κλήσεις σε βάσεις δεδομένων μέσω της SQL (δηλ., έγχυση SQL). Ολόκληρα τα scripts που γράφονται στην perl, python, και άλλες γλώσσες μπορούν να γίνουν injected σε κακοσχεδιασμένες δικτυακές εφαρμογές και να εκτελεσθούν. Όταν μια δικτυακή εφαρμογή χρησιμοποιεί έναν διερμηνέα οποιουδήποτε τύπου υπάρχει ο κίνδυνος μιας επίθεσης με injection.

Πολλές δικτυακές εφαρμογές χρησιμοποιούν στοιχεία των λειτουργικών συστημάτων και εξωτερικά προγράμματα για να εκτελέσουν τις συναρτήσεις τους. Το Sendmail είναι πιθανώς το πολύ συχνά κληθέν εξωτερικό πρόγραμμα, αλλά χρησιμοποιούνται επίσης ακόμη πολλά προγράμματα. Όταν μια δικτυακή εφαρμογή περνά τις πληροφορίες από ένα αίτημα HTTP κατευθείαν ως τμήμα ενός εξωτερικού αιτήματος, πρέπει να διεκπεραιωθεί προσεκτικά. Διαφορετικά, ο επιτιθέμενος μπορεί να εμφωλεύσει τους ειδικούς χαρακτήρες (meta), τις κακόβουλες εντολές, ή μεθόδους αλλαγής – τροποποίησης εντολών στις πληροφορίες και η δικτυακή εφαρμογή θα τα περάσει τυφλά στο εξωτερικό σύστημα όπου και θα εκτελεστούν.

Η έγχυση SQL είναι μια ιδιαίτερα διαδεδομένη και επικίνδυνη μορφή injection. Για να εκμεταλλευτεί μια διαρροή ασφαλείας μέσω εγχύσεων SQL, ο επιτιθέμενος πρέπει να βρει μια παράμετρο μέσω της οποίας η δικτυακή εφαρμογή περνά σε μια βάση δεδομένων. Ο επιτιθέμενος μπορεί να εξαπατήσει τη δικτυακή εφαρμογή στέλνοντας μια κακόβουλη ερώτηση στη βάση δεδομένων, ενσωματώνοντας προσεκτικά κακόβουλο κώδικα σε εντολές SQL, μέσα στο περιεχόμενο της παραμέτρου. Αυτές οι επιθέσεις δεν είναι δύσκολο να γίνουν και πολλά εργαλεία έχουν τη δυνατότητα να ανιχνεύσουν δικτυακούς τόπους για τέτοιου είδους προβλήματα. Οι συνέπειες είναι ιδιαίτερα καταστρεπτικές, δεδομένου ότι ένας επιτιθέμενος μπορεί να λάβει, να αλλοιώσει, ή να καταστρέψει το περιεχόμενο βάσεων δεδομένων.

Οι επιθέσεις εγχύσεων πολλές φορές είναι πολύ εύκολο να ανακαλυφθούν από κάποιον επιτιθέμενο ο οποίος μπορεί και να τις εκμεταλλευτεί, αλλά μπορούν επίσης να είναι εξαιρετικά δύσκολο να βρεθούν. Οι συνέπειες μπορεί επίσης να είναι εξαιρετικά δριμείες (μέχρι του σημείο να crashάρουν το σύστημα), ή ακόμη και ασήμαντες. Εν πάση περιπτώσει, η χρήση των εξωτερικών κλήσεων είναι αρκετά διαδεδομένη, έτσι η πιθανότητα μιας δικτυακής εφαρμογής που έχει μια ρωγμή εγχύσεων εντολών πρέπει να θεωρηθεί υψηλή.

Α6.2 Περιβάλλοντα που επηρεάζονται

Κάθε περιβάλλον δικτυακής εφαρμογής επιτρέπει την εκτέλεση εξωτερικών εντολών όπως των system calls, εντολών κελύφους και SQL ερωτήσεων. Η πιθανότητα μια

εξωτερική κλήση να υποστεί έγχυση εντολής εξαρτάται από το πως γίνεται η κλήση και το συγκεκριμένο τμήμα που καλείται. Όμως, σχεδόν όλες οι εξωτερικές κλήσεις μπορούν να δεχθούν επίθεση αν ο κώδικας της δικτυακής εφαρμογής δεν έχει γραφτεί σωστά.

Α6.3 Παραδείγματα και παραπομπές

Παραδείγματα: Μια κακόβουλη παράμετρος μπορεί να μετατρέψει τις δράσεις μιας κλήσης του συστήματος που κανονικά θα έπρεπε να αντλήσει ένα αρχείο του χρήστη και να πάρει πρόσβαση στο λογαριασμό του, σε ένα άλλο αρχείο. (πχ. Συμπεριλαμβάνοντας στο path"../" χαρακτήρες σαν ένα μέρος αιτήματος από ένα όνομα αρχείου). Επιπλέον εντολές μπορούν να προστεθούν στο τέλος των παραμέτρων που περνώνται σε ένα shell script για να εκτελέσουν ένα επιπλέον shell command (πχ., "; rm –r *") μαζί με τη δοσμένη εντολή. Ερωτήματα SQL μπορούν να διαφοροποιηθούν προσθέτοντας επιπλέον αναγκαστικές παραμέτρους σε μια where clause (πχ., "OR 1=1") για να πάρουν πρόσβαση ή να τροποποιήσουν δεδομένα στα οποία δεν θα έπρεπε να έχουν πρόσβαση.

Παραπομπές:

- OWASP Guide to Building Secure Web Applications and Web Services, Chapter 8: Data Validation

http://www.owasp.org/documentation/guide/

- How to Build an HTTP Request Validation Engine (J2EE validation with Stinger)

http://www.owasp.org/columns/jeffwilliams/jeffwilliams2

- Have Your Cake and Eat it Too (.NET validation)

http://www.owasp.org/columns/jpoteet/jpoteet2

- White Paper on SQL Injection:

http://www.spidynamics.com/papers/SQLInjectionWhitePaper.pdf

Α6.4 Πως να ελέγξετε αν είστε τρωτοί σε τέτοιου είδους επιθέσεις

Ο καλύτερος τρόπος να εντοπίσουμε εάν είμαστε τρωτοί στις επιθέσεις εγχύσεων εντολών είναι να ελέγξουμε τον πηγαίο κώδικα για όλες τις κλήσεις στους εξωτερικούς πόρους (πχ., system, exec, fork, Runtime.exec, ερωτήσεις SQL, ή οποιοσδήποτε σύνταξη είναι ικανή να υποβάλει αιτήματα στον διερμηνέα του περιβάλλοντος). Σημειώστε ότι πολλές γλώσσες έχουν πολλούς τρόπους να τρέξουν εξωτερικές εντολές. Οι προγραμματιστές πρέπει να ελέγχουν τον κώδικα και να αναζητούν όλα τα σημεία όπου η είσοδος από ένα HTTP αίτημα μπορεί πιθανώς να περάσει μέσα σε αυτές τις κλήσεις. Πρέπει να εξετάσουμε προσεκτικά κάθε μια από αυτές τις κλήσεις για να είμαστε βέβαιοι ότι τα βήματα προστασίας που περιγράφονται πιο κάτω ακολουθούνται πιστά.

Α6.5 Πως να προστατευτείτε

Ο απλούστερος τρόπος να προστατευτούμε από την έγχυση είναι να αποφύγουμε τη χρήση εξωτερικών διερμηνέων όπου αυτό είναι εφικτό. Για πολλές εντολές του shell και μερικές system calls, υπάρχουν συγκεκριμένες γλωσσικές βιβλιοθήκες που εκτελούν τις ίδιες λειτουργίες. Η χρήση τέτοιων βιβλιοθηκών δεν περιλαμβάνει τον

shell interpreter του λειτουργικού συστήματος, και επομένως αποφεύγει έναν μεγάλο αριθμό προβλημάτων με τις εντολές του κελύφους.

Για κάποιες κλήσεις που πρέπει ακόμα να προσέξουμε, όπως οι κλήσεις στις backend βάσεις δεδομένων, πρέπει προσεκτικά να ελέγξουμε τα στοιχεία που παρέχονται για να βεβαιωθούμε ότι δεν περιέχει οποιοδήποτε κακόβουλο κώδικα μέσα του. Μπορούμε επίσης να φτιάξουμε πολλά αιτήματα με έναν τρόπο που να εξασφαλίζει ότι όλες οι παρεχόμενες παράμετροι αντιμετωπίζονται ως δεδομένα, παρά το γεγονός ότι μπορεί να αποτελούν εκτελέσιμο υλικό. Η χρήση αποθηκευμένων διαδικασιών ή των έτοιμων δηλώσεων παρέχει σημαντική προστασία, που εξασφαλίζει ότι η είσοδος που δίνεται αντιμετωπίζεται ως δεδομένα. Αυτά τα μέτρα μειώνουν, αλλά δεν αποβάλουν πλήρως τον κίνδυνο που εμπεριέχεται σε αυτές τις εξωτερικές κλήσεις. Πρέπει ακόμα πάντα να ελέγχουμε την εγκυρότητα τέτοιων εισόδων για να σιγουρευτούμε ότι ικανοποιεί τα χαρακτηριστικά που θα έπρεπε να έχει μια αναμενόμενη είσοδος.

Μια ακόμη ισχυρή προστασία ενάντια στην έγχυση εντολής είναι να εξασφαλίσουμε ότι η δικτυακή εφαρμογή τρέχει μόνο με χρήση των προνομίων που χρειάζονται για να εκτελεστεί η λειτουργία που πρέπει. Έτσι δεν πρέπει να τρέχει μια εφαρμογή σε κάποιον webserver με δικαιώματα του root ή να δίνουμε πρόσβαση σε μια βάση δεδομένων ως DBADMIN, διαφορετικά ένας επιτιθέμενος μπορεί να κάνει κακόβουλη χρήση αυτών των διαχειριστικών προνομίων που δίνονται στη δικτυακή εφαρμογή. Μερικά από τα J2EE περιβάλλοντα επιτρέπουν τη χρήση της JAVA sandbox, η οποία μπορεί να αποτρέψει την εκτέλεση εντολών του συστήματος.

Εάν μια εξωτερική εντολή πρέπει να χρησιμοποιηθεί, οποιεσδήποτε πληροφορίες χρηστών που εισάγονται σε αυτήν πρέπει να ελεγχθούν αυστηρά. Οι μηχανισμοί πρέπει να θεσπιστούν για να χειριστούν οποιαδήποτε πιθανά λάθη, timeouts ή κολλήματα κατά τη διάρκεια της κλήσης. Όλες οι έξοδοι που επιστρέφουν κωδικούς και λάθη από την κλήση πρέπει να ελέγχονται οι επιστροφής κώδικες και οι κώδικες λάθους από την κλήση πρέπει να ελεγχθούν για να εξασφαλίσουν ότι η αναμενόμενη επεξεργασία συνέβη γιατί έτσι έπρεπε να γίνει και όχι με injection. Στη χειρότερη περίπτωση, αυτό θα μας επιτρέψει να διαπιστώσουμε αν κάτι έχει πάει στραβά. Διαφορετικά, η επίθεση δεν μπορεί να εμφανιστεί και να ανιχνευθεί ποτέ.

Το OWASP Filters project δημιουργεί τμήματα που μπορεί να ξαναχρησιμοποιηθούν σε διάφορες γλώσσες για να εμποδίσει πολλές μορφές έγχυσης. Το OWASP έχει βγάλει επίσης το CodeSeeker, εφαρμογή firewall σε επίπεδο λογισμικού.

A7 Ανάρμοστος χειρισμός λαθών

A7.1 Περιγραφή

Ο ανάρμοστος χειρισμός των λαθών μπορεί να δημιουργήσει μια ποικιλία προβλημάτων ασφάλειας για έναν δικτυακό τόπο. Το πιο κοινό από αυτά δημιουργείται όταν λεπτομερή εσωτερικά μηνύματα λάθους όπως stack traces, database dumps και error codes προβάλλονται στον χρήστη (hacker). Τα μηνύματα αυτά αποκαλύπτουν λεπτομέρειες της εκτέλεσης που δεν πρέπει ποτέ να αποκαλύπτονται. Τέτοιες λεπτομέρειες μπορούν να παρέχουν σημαντικές πληροφορίες στους επιτιθέμενους για πιθανές διαρροές ασφαλείας στον δικτυακό τόπο, ενώ μπορεί επίσης να γίνουν ενοχλητικά για τους κανονικούς χρήστες.

Οι δικτυακές εφαρμογές παράγουν συχνά μηνύματα λάθους κατά τη διάρκεια της κανονικής εκτέλεσής τους. Out of memory, null pointer exceptions, system call failure, database unavailable and network timeout είναι λίγοι μόνο από τους πολλούς κοινούς όρους που μπορούν να οδηγήσουν σε παραγωγή μηνυμάτων λαθών. Αυτά τα λάθη πρέπει να τα χειριστούμε σύμφωνα με ένα καλά σχεδιασμένο πλάνο που θα παράσχει κάποιο ασήμαντο μήνυμα λάθους στο χρήστη, το οποίο όμως δεν θα δίνει κάποια ευαίσθητη πληροφορία.

Ακόμα και όταν τα μηνύματα λάθους δεν παρέχουν πολλές λεπτομέρειες, οι μικρές ασυνέπειες που παρουσιάζονται μπορεί να αποκαλύψουν σημαντικές πληροφορίες σχετικά με στον τρόπο με τον οποίο λειτουργεί κάποιος δικτυακός τόπος, αλλά και ποιες πληροφορίες υπάρχουν κάτω από τα μηνύματα αυτά. Πχ, όταν κάποιος χρήστης προσπαθεί να προσπελάσει ένα αρχείο που δεν υπάρχει, το μήνυμα λάθους είναι, “file not found”. Όταν κάποιος χρήστης προσπαθεί να προσπελάσει ένα αρχείο στο οποίο του έχει απαγορευτεί η πρόσβαση, το μήνυμα λάθους είναι: “access denied”. Ο χρήστης δεν είναι υποχρεωμένος να ξέρει αν το αρχείο υπάρχει ή όχι, αλλά τέτοιου είδους πληροφορίες μπορούν να ενημερώσουν κάποιον επιτιθέμενο σχετικά με την παρουσία ή την απουσία απρόσιτων αρχείων ή της δομής καταλόγου στον οποίο βρίσκεται ο δικτυακός τόπος.

Ένα κοινό πρόβλημα ασφάλειας που παρατηρείται κατά τον ανάρμοστο χειρισμό λάθους είναι ο fail – open έλεγχος ασφάλειας. Όλοι οι μηχανισμοί ασφάλειας πρέπει να αρνηθούν την πρόσβαση μέχρι αυτή να χορηγηθεί ειδικά, να μη χορηγήσουν πρόσβαση αν αυτή γίνει απορριπτέα, κάτι που είναι ένας κοινός λόγος για τον οποίο προκύπτουν fail – open λάθη. Άλλα λάθη μπορούν να αναγκάσουν το σύστημα να πέσει – να crashάρει ή να καταναλώσουν όλους τους πόρους, πράγμα που έχει ως αποτέλεσμα να αρνείται εντελώς ή να μειώνει την απόδοση των υπηρεσιών που παρέχει στους νόμιμους χρήστες.

Οι σωστοί μηχανισμοί χειρισμού λαθών πρέπει να είναι σε θέση να διαχειριστούν οποιοδήποτε σύνολο εισόδων – μικρό ή μεγάλο – επιβάλλοντας τους κατάλληλους κανόνες ασφάλειας. Τα απλά μηνύματα λάθους πρέπει να προβάλλονται και να καταγράφονται έτσι ώστε η αιτία τους, είτε ένα λάθος στον δικτυακό τόπο είτε μια προσπάθεια παράνομης εισβολής, να μπορεί να παρατηρηθεί έτσι ώστε να παρθούν τα κατάλληλα μέτρα. Ο χειρισμός λάθους δεν πρέπει να εστιάσει απλώς στην είσοδο που παρέχεται από το χρήστη, αλλά πρέπει επίσης να συμπεριλάβει οποιαδήποτε

λάθη που μπορούν να παραχθούν από τα εσωτερικά τμήματα όπως οι κλήσεις του συστήματος, οι ερωτήσεις σε βάσεις δεδομένων, ή οποιεσδήποτε άλλες εσωτερικές λειτουργίες.

A7.2 Περιβάλλοντα που επηρεάζονται

Όλοι οι web servers, application servers, και δικτυακές εφαρμογές θεωρούνται τρωτές σε ανάρμοστους χειρισμούς λαθών.

A7.3 Παραδείγματα και παραπομπές

- OWASP discussion on generation of error codes:
http://www.owasp.org/documentation/guide/

A7.4 Πως να διαπιστώσετε αν είστε τρωτός σε αυτούς τους κινδύνους

Τυπικά, μια απλή δοκιμή μπορεί να δείξει πώς αποκρίνεται ο δικτυακός σας τόπος στα διάφορα είδη λαθών στην είσοδο. Ένας πιο λεπτομερής έλεγχος απαιτείται συνήθως για δημιουργήσει εκούσια λάθη στην είσοδο και να δούμε τον τρόπο με τον οποίο συμπεριφέρεται ο δικτυακός τόπος.

Μια άλλη σπουδαία προσέγγιση είναι να υπάρξει ένας λεπτομερής έλεγχος του κώδικα, αναζητώντας μέσα σε αυτόν σημεία που παραπέμπουν σε λογική χειρισμού λαθών. Ο χειρισμός λάθους πρέπει να είναι ενδελεχής μέσα καλύπτοντας όλο το εύρος του δικτυακού τόπου και κάθε κομμάτι πρέπει να αποτελεί μέρος ενός καλά σχεδιασμένου σχήματος. Μια συνοπτική μορφή του κώδικα θα αποκαλύψει πώς το σύστημα έχει σχεδιαστεί να χειριστεί τους διάφορους τύπους λαθών. Εάν διαπιστώνετε ότι δεν υπάρχει καμία οργάνωση στο σχέδιο χειρισμού των λαθών ή ότι φαίνονται να υπάρχουν διάφορα διαφορετικά σχέδια, είναι πολύ πιθανό να υπάρξει πρόβλημα.

A7.5 Πως να προστατευτείτε

Μια συγκεκριμένη πολιτική πάνω στον χειρισμό των λαθών πρέπει να καταγραφεί, και να συμπεριλάβει μέσα της τους τύπους των λαθών που πρέπει να αντιμετωπίζονται και για κάθε ένα, ποιες πληροφορίες πρόκειται να δοθούν πίσω στον χρήστη, αλλά και ποιες πληροφορίες πρόκειται να καταγραφούν. Οι προγραμματιστές πρέπει να καταλάβουν την πολιτική και να εξασφαλίσουν ότι ο κώδικάς τους ακολουθεί πιστά την πολιτική αυτή.

Στην υλοποίηση πρέπει να εξασφαλίσουμε ότι ο δικτυακός τόπος χτίζεται κατάλληλα ώστε να χειριστεί σωστά όλα τα πιθανά λάθη. Όταν τα λάθη εμφανίζονται, το site πρέπει να απαντήσει με ένα από πριν σχεδιασμένο αποτέλεσμα που θα φανεί χρήσιμο στο χρήστη χωρίς όμως να αποκαλύπτει περιττές εσωτερικές λεπτομέρειες. Ορισμένες κατηγορίες λαθών πρέπει να καταγραφούν για να βοηθήσουν στην ανίχνευση διαρροών ασφαλείας ή ακόμη και των προσπαθειών παράνομης εισβολής.

Πολύ λίγοι δικτυακοί τόποι εμπεριέχουν δυνατότητες ανίχνευσης εισβολών στη δικτυακή εφαρμογή τους, αλλά είναι βεβαίως κατανοητό ότι μια δικτυακή εφαρμογή μπορεί να εντοπίσει πιθανές επαναλαμβανόμενες αποτυχημένες προσπάθειες και να

ενεργοποιήσει μηνύματα λάθους για λόγους ασφαλείας. Σημειώστε ότι η μεγάλη πλειοψηφία των επιθέσεων σε δικτυακές εφαρμογές δεν ανιχνεύεται ποτέ επειδή πολύ λίγα site μπορούν να τις ανιχνεύσουν. Επομένως, η διάδοση των επιθέσεων κατά της ασφάλειας των δικτυακών εφαρμογών είναι πιθανό να υποτιμηθεί σοβαρά.

Το OWASP Filters project δημιουργεί επαναχρησιμοποιήσιμα τμήματα σε διάφορες γλώσσες για να παρεμποδίσει την εμφάνιση κωδίκων λάθους στην ιστοσελίδα του χρήστη. Αυτό επιτυγχάνεται φιλτράροντας τέτοιες σελίδες που παράγονται δυναμικά από την εφαρμογή.

A8 Μη ασφαλής αποθήκευση

Οι περισσότερες δικτυακές εφαρμογές χρειάζεται να αποθηκεύσουν τις ευαίσθητες πληροφορίες, είτε σε μια βάση δεδομένων είτε για κάποιο σύστημα αρχείων. Οι πληροφορίες αυτές μπορεί να είναι κωδικοί πρόσβασης, αριθμοί πιστωτικών καρτών, αρχεία λογαριασμών, ή προσωπικά δεδομένα. Συχνά, χρησιμοποιούνται τεχνικές κρυπτογράφησης για να προστατευτούν αυτές τις ευαίσθητες πληροφορίες. Ενώ η κρυπτογράφηση έχει γίνει σχετικά εύκολο να εφαρμοστεί και να χρησιμοποιηθεί, οι προγραμματιστές συχνά κάνουν λάθη ενσωματώνοντας αυτές σε μια δικτυακή εφαρμογή. Οι προγραμματιστές μπορεί να υπερεκτιμήσουν την προστασία που παρέχεται από τη χρήση της κρυπτογράφησης και να μη φανούν προσεκτικοί κατά την εξασφάλιση παραμέτρων ασφαλείας σε άλλα σημεία του δικτυακού τόπου. Μερικές περιοχές όπου γίνονται συνήθως λάθη είναι:

- Αποτυχία να κρυπτογραφηθούν τα κρίσιμα και ευαίσθητα δεδομένα
- Επισφαλής αποθήκευση των κλειδιών, των πιστοποιητικών, και των κωδικών πρόσβασης
- Ανάρμοστη αποθήκευση των μυστικών στη μνήμη
- Εκλογή τυχαίων αριθμών από μικρό εύρος
- Επιλογή λάθος αλγορίθμου
- Προσπάθεια δημιουργίας νέου αλγορίθμου κρυπτογράφησης
- Αποτυχία να περιληφθεί η υποστήριξη για τις βασικές αλλαγές κρυπτογράφησης και άλλες απαραίτητες διαδικασίες συντήρησης

Ο αντίκτυπος αυτών των αδυναμιών μπορεί να είναι καταστρεπτικός στην ασφάλεια ενός ιστοχώρου. Η κρυπτογράφηση χρησιμοποιείται γενικά για να προστατεύσει τα πιό ευαίσθητα στοιχεία που μπορεί να περιλαμβάνει ένας δικτυακός τόπος, τα οποία μπορεί να προσβληθούν καθολικά από κάποια αδυναμία του συνόλου.

A8.2 Περιβάλλοντα που επηρεάζονται

Τα περισσότερα περιβάλλοντα δικτυακών εφαρμογών περιλαμβάνουν κάποια μορφή υποστήριξης κρυπτογραφίας. Στη σπάνια περίπτωση που υποστήριξη κρυπτογραφικών μεθόδων δεν παρέχεται ήδη, υπάρχει μια μεγάλη ποικιλία προϊόντων τρίτων κατασκευαστών που μπορεί να ενσωματωθεί στο λογισμικό μας. Οι ιστοχώροι που χρησιμοποιούν την κρυπτογράφηση για να προστατεύσουν τις πληροφορίες μόνο κατά τη διάρκεια της αποθήκευσης ή της μεταφοράς δεδομένων είναι ευαίσθητοι σε αυτές τις επιθέσεις. Σημειώστε ότι αυτό το τμήμα δεν καλύπτει τη χρήση της SSL, η οποία καλύπτεται στο κεφάλαιο A10 που αναφέρεται στην επισφαλή διαχείριση της παραμετροποίησης του συστήματος. Αυτό το τμήμα ασχολείται μόνο με την προγραμματική κρυπτογράφηση των στοιχείων στρώματος εφαρμογής.

A8.3 Παραδείγματα και παραπομπές

- OWASP Guide to Building Secure Web Applications and Web Services
http://www.owasp.org/documentation/guide/
- Bruce Schneier, “Applied Cryptography”, 2nd edition, John Wiley & Sons, 1995

Α8.4 Πως να διαπιστώσετε αν είστε τρωτοί σε τέτοιες διαρροές ασφαλείας

Η ανακάλυψη των διαρροών ασφαλείας λόγω κακής χρήσης της κρυπτογράφησης χωρίς να έχουμε πρόσβαση στον πηγαίο κώδικα μπορεί να είναι μια εξαιρετικά χρονοβόρα διαδικασία. Ωστόσο, είναι δυνατό να εξεταστούν οι συνδέσεις, τα session IDs, τα cookies και άλλα χαρακτηριστικά για να διαπιστώσουμε εάν αυτά είναι τυχαία. Όλες οι παραδοσιακές προσεγγίσεις της κρυπτογραφίας μπορούν να χρησιμοποιηθούν για να επιχειρήσουμε να αποκαλύψουμε τον τρόπο που κάποιος δικτυακός τόπος χρησιμοποιεί τις κρυπτογραφικές λειτουργίες.

Η μακράν ευκολότερη προσέγγιση είναι να κάνουμε μια προσεκτική μελέτη του κώδικα και να μελετήσουμε τον τρόπο που οι κρυπτογραφικές λειτουργίες εφαρμόζονται. Πρέπει ακόμη να διενεργηθεί μια προσεκτική θεώρηση της δομής, της ποιότητας, και της εφαρμογής των κρυπτογραφικών κανόνων. Ο κριτής πρέπει να έχει ένα ισχυρό γνωσιακό υπόβαθρο στη χρήση του συστήματος κρυπτογραφίας και των διαρροών ασφαλείας που αυτές μπορεί να παρουσιάζουν. Η μελέτη αυτή πρέπει επίσης να περιλαμβάνει και τον τρόπο που τα κλειδιά, οι κωδικοί πρόσβασης, και άλλα μυστικά αποθηκεύονται, προστατεύονται, φορτώνονται στη μνήμη, τρέχουν στον επεξεργαστή, και καθαρίζονται από τη μνήμη.

Α8.5 Πως να προστατευτείτε από τέτοιου είδους διαρροές ασφαλείας

Ο ευκολότερος τρόπος να προστατευτούμε από τυχόν διαρροές που προκύψουν λόγω ελλείψεων σε επίπεδο κρυπτογραφίας είναι να ελαχιστοποιήσουμε τη χρήση της κρυπτογράφησης και να κρατηθούν μόνο οι πληροφορίες που είναι απολύτως απαραίτητες. Πχ, παρά την κρυπτογράφηση των αριθμών της πιστωτικής κάρτας και την αποθήκευση τους, απλά ζητήστε από τους χρήστες να ξανά δώσουν από το πληκτρολόγιο τον αριθμό της κάρτας τους. Επίσης, αντί να αποθηκεύουμε τους κρυπτογραφημένους κωδικούς πρόσβασης, χρησιμοποιήστε μια μονόδρομη συνάρτηση, όπως η SHA-1, για να πάρουμε το hashing των κωδικών πρόσβασης.

Εάν η κρυπτογραφία πρέπει να χρησιμοποιηθεί, επιλέξτε μια βιβλιοθήκη που έχει διερευνηθεί από πολλούς, και συνεπώς είμαστε βέβαιοι για την αξιοπιστία της, και σιγουρευτείτε ότι δεν υπάρχει καμία ανοικτή ευπάθεια. Συγκεντρώστε τις κρυπτογραφικές λειτουργίες που χρησιμοποιούνται και μελετήστε τον κώδικα προσεκτικά. Να είστε βέβαιοι ότι μυστικά, όπως τα κλειδιά, πιστοποιητικά, και κωδικοί πρόσβασης, αποθηκεύονται ασφαλώς. Για δυσκολέψουμε το έργο ενός επιτιθέμενου, κάθε τι κωδικοποιημένο μυστικό (όπως passwords) πρέπει να χωριστεί σε τουλάχιστον δύο μέρη αποθηκευμένα σε διαφορετικά σημεία, τα οποία θα επανενωθούν μόνο όταν χρειαστεί να χρησιμοποιηθούν. Τέτοιες θέσεις μπορεί να περιλαμβάνουν ένα αρχείο διαμόρφωσης, έναν εξωτερικό διακομιστή ή το εσωτερικό του ίδιου του κώδικα.

A9 Άρνηση παροχής υπηρεσιών

A9.1 Περιγραφή

Οι δικτυακές εφαρμογές είναι ιδιαίτερα ευπαθείς σε επιθέσεις άρνησης παροχής υπηρεσιών. Σημειώστε ότι οι επιθέσεις σε δίκτυα άρνησης παροχής υπηρεσιών, όπως οι SYN floods, είναι ένα διαφορετικό πρόβλημα που είναι δεν έχει σχέση με το έγγραφο αυτό.

Μια δικτυακή εφαρμογή δεν μπορεί εύκολα να διακρίνει τη διαφορά μεταξύ μιας επίθεσης και μιας συνηθισμένης κυκλοφορίας δεδομένων. Υπάρχουν πολλοί παράγοντες που συμβάλλουν στην δυσκολία αυτή, αλλά ένας από το σημαντικότερους είναι ότι, για διάφορους λόγους, οι διευθύνσεις IP δεν είναι χρήσιμες ως πιστοποιητικό προσδιορισμού της ταυτότητας ενός τερματικού. Επειδή δεν υπάρχει κανένας αξιόπιστος τρόπος να διακρίνουμε την πηγή ενός HTTP αιτήματος, είναι πολύ δύσκολο να φιλτραριστεί η κυκλοφορία που περιέχει κακόβουλο κώδικα και εντολές. Για ένα σύνολο επιθέσεων, πώς μια εφαρμογή θα μπορούσε να διακρίνει τη διαφορά μεταξύ μιας αληθινής επίθεσης, όταν πολλοί χρήστες κάνουν ταυτόχρονα refresh (πράγμα που μπορεί να συμβεί εάν υπάρξει ένα προσωρινό πρόβλημα με το site);

Οι περισσότεροι web servers μπορούν να χειριστούν εκατοντάδες ταυτόχρονους χρήστες που κάνουν όμως λογική χρήση του site. Κάποιος επιτιθέμενος μπορεί να προκαλέσει αρκετή κυκλοφορία από κάποιο συγκεκριμένο σημείο του διαδικτύου για να πλημμυρίσει πολλές εφαρμογές. Η εξισορρόπηση του φόρτου της κίνησης μπορεί να δυσκολέψει την επιτυχία των επιθέσεων αυτών, χωρίς όμως να τις καταστήσει αδύνατες να συμβούν, ειδικά εάν κάποιες συνδέσεις είναι άρρηκτα δεμένες με κάποιο συγκεκριμένο διακομιστή. Αυτός είναι ένας καλός λόγος να φροντίσουμε ώστε τα στοιχεία συνδέσεων μιας εφαρμογής να είναι όσο το δυνατόν μικρότερα σε μέγεθος και να φροντίσουμε να δυσκολέψουμε τη διαδικασία σύναψης μιας νέας σύνδεσης .

Μόλις ένας επιτιθέμενος μπορέσει να καταναλώσει όλους τους απαραίτητους πόρους, τότε πετυχαίνει αυτόματα και τον αποκλεισμό των νόμιμων χρηστών του δικτυακού τόπου από το να τον χρησιμοποιήσουν. Κάποιοι πόροι που από τη φύση τους είναι περιορισμένοι περιλαμβάνουν το εύρος ζώνης, το σύνολο των συνδέσεων στη βάση δεδομένων, τον αποθηκευτικό χώρο στον δίσκο, την χρήση της ΚΜΕ, τη διαθέσιμη μνήμη, τα νήματα που τρέχουν, ή τους συγκεκριμένους πόρους που η εφαρμογή έχει στη διάθεσή της για να χρησιμοποιήσει. Όλοι αυτοί οι πόροι μπορούν να καταναλωθούν από επιτιθέμενους που στόχο έχουν την εξάντλησή τους έτσι ώστε να μην είναι πλέον η υπηρεσία διαθέσιμη. Πχ, ένα site που επιτρέπει σε μη πιστοποιημένους χρήστες να ζητήσουν την έναν πίνακα μηνυμάτων της κυκλοφορίας, μπορεί να αρχίσει πολλές διαδοχικές ερωτήσεις σε κάποια βάση δεδομένων για κάθε αίτημα HTTP που λαμβάνει. Ένας επιτιθέμενος μπορεί εύκολα να στείλει τόσα πολλά αιτήματα που ο αριθμός των συνδέσεων με τη βάση δεδομένων να γίνει τόσο μεγάλος που δεν θα επιτρέπει πλέον σε κανένα πιστοποιημένο χρήστη να κάνει μια ερώτηση στη βάση.

Άλλες παραλλαγές αυτών των τύπων των επιθέσεων εναντίον των πόρων του συστήματος μπορεί να αφορούν κάποιον συγκεκριμένο χρήστη. Πχ, ένας επιτιθέμενος μπορεί να είναι σε θέση να κλειδώσει έξω από το σύστημα έναν νόμιμο χρήστη με την αποστολή των άκυρων πιστοποιητικών έως ότου κλειδώνει καταφέρει να κλειδώσει τον λογαριασμό του εξαιτίας των συνεχών αποτυχημένων προσπαθειών login στο σύστημα. Ή ο επιτιθέμενος μπορεί να ζητήσει έναν νέο κωδικό πρόσβασης για κάποιον χρήστη, πράγμα που τον αναγκάζει να έχει πρόσβαση σε έναν εναλλακτικό λογαριασμό ηλεκτρονικού ταχυδρομείου για να πάρει τους νέους κωδικούς. Εναλλακτικά, εάν το σύστημα κλειδώνει όλους τους πόρους για έναν συγκεκριμένο χρήστη, κατόπιν ένας επιτιθέμενος θα μπορούσε ενδεχομένως να τους κρατήσει έτσι ώστε άλλοι να μην μπορούν να τους χρησιμοποιήσουν. Μερικές δικτυακές εφαρμογές είναι ακόμα και ευαίσθητες στις επιθέσεις που θα τις βγάλουν offline άμεσα. Οι εφαρμογές που δεν χειρίζονται ορθά τα λάθη μπορούν ακόμη και να ρίξουν το σύστημα στο οποίο τρέχει η δικτυακή εφαρμογή. Αυτές οι επιθέσεις είναι ιδιαίτερα καταστρεπτικές επειδή αποτρέπουν αμέσως όλους τους άλλους χρήστες από τη χρήση της εφαρμογής.

Υπάρχει μια ευρεία ποικιλία αυτών των επιθέσεων, οι περισσότερες από τις οποίες μπορούν να πραγματοποιηθούν εύκολα με μερικές γραμμές κώδικα perl από έναν υπολογιστή χαμηλών δυνατοτήτων. Ενώ δεν υπάρχει καμία τέλεια άμυνα σε αυτές τις επιθέσεις, μπορούμε να πάρουμε μέτρα εναντίον του, καθιστώντας όσο το δυνατόν πιο δύσκολο το έργο των επιτιθέμενων.

A9.2 Περιβάλλοντα που επηρεάζονται

Όλοι οι web servers, οι application servers, και τα περιβάλλοντα δικτυακών εφαρμογών είναι ευάλωτα σε επιθέσεις άρνησης παροχής υπηρεσιών.

A9.3 Παραδείγματα και παραπομπές

- OWASP Guide to Building Secure Web Applications and Web Services

http://www.owasp.org/documentation/guide/

A9.4 Πως να διαπιστώσετε αν είστε ευάλωτοι σε τέτοιου είδους επιθέσεις

Ένα από τα σκληρότερα μέρη της άρνησης των επιθέσεων υπηρεσιών καθορίζει εάν είστε τρωτοί. Τα εργαλεία δοκιμής φορτίων, όπως JMeter μπορούν να παραγάγουν την κυκλοφορία Ιστού έτσι ώστε μπορείτε να εξετάσετε ορισμένες πτυχές για το πώς η περιοχή σας αποδίδει κάτω από το βαρύ φορτίο. Βεβαίως μια σημαντική δοκιμή είναι πόσα αιτήματα ανά δευτερόλεπτο η αίτησή σας μπορεί να τοποθετήσει. Η δοκιμή από μια ενιαία διεύθυνση IP είναι χρήσιμη δεδομένου ότι θα σας δώσει μια ιδέα πόσων αιτημάτων ένας επιτιθέμενος θα πρέπει να παραγάγει προκειμένου να βλαφθεί η περιοχή σας.

One of the hardest parts of denial of service attacks is determining whether you are vulnerable. Load testing tools, such as JMeter can generate web traffic so that you can test certain aspects of how your site performs under heavy load. Certainly one important test is how many requests per second your application can field. Testing from a single IP address is useful as it will give you an idea of how many requests an attacker will have to generate in order to damage your site.

Για να καθορίσετε εάν οποιοιδήποτε πόροι μπορούν να χρησιμοποιηθούν για να δημιουργήσουν μια άρνηση της υπηρεσίας, πρέπει να αναλύσετε καθεμία για να δείτε εάν υπάρχει ένας τρόπος να εξαντληθεί. Πρέπει ιδιαίτερα να εστιάσετε σε αυτό που ένας πλαστός χρήστης μπορεί να κάνει, αλλά εκτός αν εμπιστεύεστε τους όλους χρήστες σας, πρέπει να εξετάσετε τι ένας επικυρωμένος χρήστης μπορεί να κάνει επίσης.

To determine if any resources can be used to create a denial of service, you should analyze each one to see if there is a way to exhaust it. You should particularly focus on what an unauthenticated user can do, but unless you trust all of your users, you should examine what an authenticated user can do as well.

A9.5 How to Protect Yourself

Η υπεράσπιση ενάντια σε επιθέσεις άρνησης παροχής υπηρεσιών είναι δύσκολη υπόθεση, δεδομένου ότι δεν υπάρχει κανένας τέλειος τρόπος για να προστατευθεί κανείς από αυτές. Ακολουθώντας έναν γενικό κανόνα, πρέπει να περιορίσετε τους πόρους που διατίθενται σε οποιοδήποτε χρήστη σε ένα ελάχιστο βαθμό. Για τους επικυρωμένους χρήστες, είναι δυνατό να ορίσουμε τα ποσοστά αυτά έτσι ώστε να περιορίσετε το ποσό του φορτίου που ένας συγκεκριμένος χρήστης μπορεί να βάλει στο σύστημά σας. Ειδικότερα, δώστε δικαιώματα μόνο για ένα αίτημα ανά χρήστη σε έναν συγκεκριμένο χρονικό διάστημα. Επίσης μπορείτε να ορίσετε έναν κανόνα σύμφωνα με τον οποίο οποιοδήποτε αίτημα που επεξεργάζεστε κάποια χρονική στιγμή για έναν χρήστη να ακυρώνεται όταν φθάνει ένα άλλο αίτημα από τον ίδιο χρήστη.

Για τους μη πιστοποιημένους χρήστες, πρέπει να αποφύγετε οποιαδήποτε περιττή πρόσβαση σε βάσεις δεδομένων ή πόρους που καταναλώνουν μεγάλο ποσοστό στη CPU. Δοκιμάστε να οργανώσετε τη ροή των πόρων στον δικτυακό σας τόπο έτσι ώστε ένας μη πιστοποιημένος χρήστης να μην είναι σε θέση να καλέσει οποιεσδήποτε διαδικασίες που απαιτούν εξάντληση των πόρων. Μπορείτε επίσης να cacháρετε το περιεχόμενο που ζητήθηκε από μη πιστοποιημένους χρήστες έτσι ώστε να μη χρειάζεται κάθε φορά που ζητείται το ίδιο πράγμα να ξαναγίνεται ερώτηση στη βάση.

Τέλος, πρέπει να ελέγξετε το σχέδιο χειρισμού λαθών που χρησιμοποιείτε για να εξασφαλίσετε ότι ένα λάθος δεν μπορεί να έχει επιπτώσεις στη γενική λειτουργία της εφαρμογής.

A10 Insecure Configuration Management

A10.1 Description

Οι διαμορφώσεις των web και application server διαδραματίζουν έναν βασικό ρόλο στην ασφάλεια μιας δικτυακής εφαρμογής. Αυτοί οι διακομιστές είναι αρμόδιοι για την εξασφάλιση της διακίνησης των δεδομένων και της κλήσης των εφαρμογών που τα παράγουν. Επιπλέον, πολλοί application server παρέχουν πολλές υπηρεσίες που οι δικτυακές εφαρμογές μπορούν να χρησιμοποιήσουν, συμπεριλαμβανομένης της αποθήκευσης αρχείων, directory services, mail, messaging, και άλλων πολλών. Η αποτυχία να ρυθμιστεί η κατάλληλη διαμόρφωση των διακομιστών σας μπορεί να οδηγήσει σε μια μεγάλη ποικιλία των προβλημάτων ασφάλειας.

Συχνά, οι προγραμματιστές της δικτυακής εφαρμογής διαφέρουν από τους διαχειριστές του δικτυακού τόπου. Στην πραγματικότητα, υπάρχει συχνά ένα μεγάλο χάσμα μεταξύ εκείνων που γράφουν την εφαρμογή και εκείνων που είναι υπεύθυνοι για τη διαχείριση του περιβάλλοντος. Τα προβλήματα ασφάλειας των δικτυακών εφαρμογών αφορούν συχνά αυτό το χάσμα και απαιτούν τη συνεργασία των μελών και από τις δύο πλευρές του project για να εξασφαλιστεί η ασφάλεια της εφαρμογής του δικτυακού τόπου.

Υπάρχει μια ευρεία ποικιλία των προβλημάτων διαμόρφωσης των παραμέτρων των διακομιστών που μπορεί να επηρεάσει την ασφάλεια ενός site. Ανάμεσα σε αυτά διακρίνουμε:

- Ρήγματα ασφαλείας λόγω ύπαρξης μη ενημερωμένων εκδόσεων λογισμικού στον διακομιστή
- Ρωγμές ασφαλείας του λογισμικού ή misconfigurations του διακομιστή που επιτρέπουν επιθέσεις με directory listing και directory traversal
- Περιττές προεπιλογές ρυθμίσεων, backups, ή sample files, συμπεριλαμβανομένων των scripts, των εφαρμογών, των αρχείων διαμόρφωσης και ιστοσελίδων
- Λαθεμένες οδηγίες πρόσβασης αρχείων και καταλόγων
- Περιττές παρεχόμενες υπηρεσίες, συμπεριλαμβανομένης της διαχείρισης περιεχομένου και της απομακρυσμένης πρόσβασης για διαχείριση
- Ύπαρξη προεπιλεγμένων λογαριασμών μαζί με τους default κωδικούς πρόσβασης που συνήθως έχουν
- Διαχειριστικές ή λειτουργίες διόρθωσης λαθών που επιτρέπονται ή προσφέρονται σε όλους
- Μηνύματα λάθους που προσφέρουν υπερβολικά πολλές πληροφορίες που συνήθως είναι περιττές (περισσότερες λεπτομέρειες στο τμήμα χειρισμού λάθους)
- Πιστοποιητικά SSL και παραμετροποιήσεις κρυπτογράφησης που δεν έχουν setαριστεί σωστά
- Χρήση ερασιτεχνικά δημιουργηθέντων πιστοποιητικών για να επιτύχουμε την πιστοποίηση ή πιστοποίηση με μεσολάβηση του ανθρώπινου παράγοντα
- Χρήση των προεπιλεγμένων πιστοποιητικών
- Λαθεμένη διαδικασία επικύρωσης με τα εξωτερικά συστήματα.

Μερικά από αυτά τα προβλήματα μπορούν να ανιχνευθούν εύκολα, με διαθέσιμα εργαλεία ανίχνευσης πιθανών διαρροών ασφαλείας. Μόλις ανιχνευθούν κάποια από αυτά, αυτά τα προβλήματα μπορούν να χρησιμοποιηθούν ως πολύτιμη εμπειρία για την κατασκευή ενός πιο ασφαλούς δικτυακού τόπου. Οι επιτυχείς επιθέσεις μπορούν επίσης να χρησιμοποιηθούν με τον ίδιο τρόπο οδηγώντας σε αποκτηθείσα εμπειρία σχετικά με την ασφάλεια backend βάσεων δεδομένων και άλλων συνεργαζόμενων δικτύων. Το να έχουμε στη διάθεσή μας ασφαλές λογισμικό αλλά και μια ασφαλή παραμετροποίησή του είναι από κοινού απαιτητό για να δημιουργήσουμε έναν ασφαλή δικτυακό τόπο.

A10.2 Environments Affected

Όλοι οι web server, οι application server, και τα περιβάλλοντα δικτυακών εφαρμογών είναι ευαίσθητα σε διαρροές ασφαλείας λόγω κακής παραμετροποίησης.

A10.3 Παραδείγματα και Παραπομπές

- OWASP Guide to Building Secure Web Applications and Web Services
http://www.owasp.org/documentation/guide/
- Web Server Security Best Practices:
http://www.pcmag.com/article2/0,4149,11525,00.asp
- Securing Public Web Servers (from CERT):
http://www.cert.org/security-improvement/modules/m11.html

A10.4 Πως να διαπιστώσετε αν είστε ευάλωτος σε τέτοιες διαρροές ασφαλείας

Εάν δεν έχετε προσπαθήσει να κλειδώσετε τους διακομιστές του δικτύου σας και των εφαρμογών σας είστε πλέον πιθανότατα τρωτοί σε επιθέσεις. Λίγοι, ενδεχομένως, διακομιστές θεωρούνται ασφαλείς από την κατασκευή τους. Μια ασφαλής διαμόρφωση για την πλατφόρμα σας πρέπει να καταγραφεί εγγράφως και να ενημερώνεται συχνά. Ένας έλεγχος του οδηγού παραμετροποίησης πρέπει να καταγραφεί και να ενημερώνεται συχνά για να διασφαλιστεί ότι είναι ενημερωμένος και συνεπής. Μια σύγκριση με τα πραγματικά επεκταμένα συστήματα συστήνεται επίσης.

Επιπλέον, υπάρχουν διάφορα διαθέσιμα προϊόντα ανίχνευσης που θα ελέγξουν εξωτερικά έναν web ή application server για τις γνωστές ευπάθειες, συμπεριλαμβανομένων των Nessus και Nikto. Πρέπει να τρέξετε αυτά τα εργαλεία σε συχνή βάση, τουλάχιστον μηνιαία, για να βρείτε πιθανά προβλήματα όσο πια έγκαιρα γίνεται. Τα εργαλεία πρέπει να τρέξουν και εσωτερικά και εξωτερικά. Οι εξωτερικοί έλεγχοι πρέπει να τρέξουν από έναν host εκτός του δικτύου του διακομιστή. Οι εσωτερικοί έλεγχοι πρέπει να τρέξουν από υπολογιστές στο ίδιο δίκτυο με τους διακομιστές που ίσως να αποτελέσουν στόχους.

A10.5 Πως να προστατευτείτε

Το πρώτο βήμα είναι να δημιουργηθεί μια σκληρή γραμμή άμυνας που να αποτελείται από οδηγίες σχετικά με την παραμετροποίηση ειδικά του δικού σας web

και application server. Αυτή η διαμόρφωση πρέπει να χρησιμοποιείται σε όλους τους hosts που τρέχουν την εφαρμογή όπως και στο περιβάλλον ανάπτυξης εφαρμογών. Προτείνουμε αρχίζοντας από οποιαδήποτε προϋπάρχουσα καθοδήγηση που μπορείτε να βρείτε από τον προμηθευτή σας ή άλλους διαθέσιμους από τις διάφορες υπάρχουσες οργανώσεις ασφάλειας όπως OWASP, CERT, και SANS και να τις προσαρμόσετε στις ιδιαίτερες ανάγκες σας. Οι οδηγίες αυτές πρέπει να αφορούν τα ακόλουθα θέματα:

- Παραμετροποίηση όλων των μηχανισμών ασφάλειας
- Κλείσιμο όλων των services που δεν χρησιμοποιούνται
- Οργάνωση των ρόλων, των αδειών και των λογαριασμών των χρηστών (η διαδικασία αυτή περιλαμβάνει την απενεργοποίηση των default λογαριασμών ή αλλαγή των password που αντιστοιχούν σε αυτούς)
- Δημιουργία αρχείων καταγραφής και μηνυμάτων λάθους

Μόλις καταγραφούν οι οδηγίες σας, μπορείτε να τις χρησιμοποιήσετε για να διαμορφώσετε και να διατηρήσετε τους διακομιστές σας σύμφωνα με τις περιγραφές. Εάν έχετε έναν μεγάλο αριθμό διακομιστών για να διαμορφώσετε, εφαρμόστε μια ημιαυτόματη ή εντελώς αυτόματη διαδικασία. Χρησιμοποιήστε ένα υπάρχον εργαλείο διαμόρφωσης ή αναπτύξτε κάποιο δικό σας. Διάφορα τέτοια εργαλεία υπάρχουν διαθέσιμα στο διαδίκτυο. Μπορείτε επίσης να χρησιμοποιήσετε τα εργαλεία όπως το GHOST για να πάρετε ένα image ενός διακομιστή και, σύμφωνα με αυτό, να στήσετε αυτόματα και τους υπόλοιπους. Μια τέτοια διαδικασία μπορεί να λειτουργήσει ή και να μη λειτουργήσει ανάλογα με το περιβάλλον του δικού σας δικτύου.

Το να διαφυλάξετε τη συγκεκριμένη διαμόρφωση των διακομιστών σας απαιτεί συνεχή επαγρύπνηση. Πρέπει να είστε βέβαιοι ότι η ευθύνη για τη διαμόρφωση των διακομιστών έχει ανατεθεί σε κάποιον που μπορείτε να εμπιστευτείτε, ο οποίος κρατάει τον διακομιστή ενημερωμένο με τις τελευταίες εκδόσεις. Η διαδικασία συντήρησης πρέπει να περιλαμβάνει:

- Την παρακολούθηση των πιο πρόσφατα δημοσιευμένων προβλημάτων ασφαλείας που παρουσιάζονται
- Εφαρμογή των πιο πρόσφατων ενημερωμένων εκδόσεων ασφαλείας
- Ενημέρωση των οδηγιών διαμόρφωσης ασφάλειας
- Συχνό έλεγχο διαρροών ασφαλείας και από τις εσωτερικές και εξωτερικές πλευρές του διακομιστή
- Συχνούς εσωτερικούς ελέγχους της ασφάλειας του διακομιστή σε σύγκριση με την περιγραφή της παραμετροποίησης που έχει γίνει
- Συχνές εξαγωγές εκθέσεων προς τον διαχειριστή του δικτύου που τεκμηριώνει τη γενική στάση ασφάλειας

Συμπεράσματα

Η OWASP έχει δημιουργήσει αυτόν τον κατάλογο για να αυξήσει την πληροφόρηση σχετικά με την ασφάλεια των δικτυακών εφαρμογών. Οι ειδικοί της OWASP έχουν καταλήξει στο συμπέρασμα ότι αυτές οι ευπάθειες αντιπροσωπεύουν έναν σοβαρό κίνδυνο για τους οργανισμούς και τις επιχειρήσεις που έχουν εκθέσει την επιχειρησιακή λογική τους στο Διαδίκτυο. Τα προβλήματα ασφάλειας των δικτυακών εφαρμογών είναι τόσο σοβαρά όσο τα προβλήματα ασφάλειας των δικτύων, αν και αυτά έχουν λάβει παραδοσιακά αρκετά λιγότερη προσοχή. Οι επιτιθέμενοι έχουν αρχίσει να εστιάζουν στα προβλήματα ασφάλειας των δικτυακών εφαρμογών, και αναπτύσσουν συνεχώς εργαλεία και τεχνικές για τον εντοπισμό και την αντιμετώπισή τους.

Αυτός ο κατάλογος του Top Ten είναι μόνο μια αφετηρία. Πιστεύουμε ότι αυτές οι ρωγμές αντιπροσωπεύουν τους σοβαρότερους κινδύνους για την ασφάλεια των δικτυακών εφαρμογών, αλλά υπάρχουν πολλοί άλλοι κρίσιμοι τομείς ασφαλείας που εξετάστηκαν για να ενταχθούν στον κατάλογο και αντιπροσωπεύουν επίσης σημαντικό κίνδυνο για τους οργανισμούς που επεκτείνουν τις δικτυακές εφαρμογές. Αυτοί περιλαμβάνουν ρωγμές στους τομείς:

- Περιττού και κακόβουλου κώδικα
- Σπασμένη ασφάλεια νημάτων και παράλληλος προγραμματισμός
- Μη πιστοποιημένη συλλογή πληροφοριών
- Προβλήματα λογαριασμών και αδύναμη πιστοποίηση
- Παραποίηση δεδομένων
- Παραποίηση προσωρινά αποθηκευμένων στοιχείων, συγκέντρωση και επαναχρησιμοποίησή τους

Χαιρετίζουμε τη συνδρομή σας σε αυτόν κατάλογο του Top Ten. Παρακαλώ συμμετέχετε στη mailing list του OWASP και βοηθήστε να βελτιώσουμε την ασφάλεια των δικτυακών εφαρμογών. Επισκεφθείτε το http://www.owasp.org για να ξεκινήσετε τη συνδρομή σας στην προσπάθειά μας.

We welcome your feedback on this Top Ten list. Please participate in the OWASP mailing lists and help to improve web application security. Visit http://www.owasp.org to get started.