OWASP: Ασφάλεια στις Web εφαρμογές

Οταν μιλάμε για ασφάλεια στις Web εφαρμογές, δεν χρειάζεται πανικός, αλλά γνώση και κατάλληλα προληπτικά εργαλεία. Ο Κωσταντίνος Παπαπαναγιώτου, εκ μέρους του OWASP, εξηγεί τι πρέπει να γνωρίζει κάποιος, για να μην έχει δυσάρεστες εκπλήξεις.

ΤΟΥ ΚΩΝΣΤΑΝΤΙΝΟΥ ΠΑΠΑΠΑΝΑΓΙΩΤΟΥ*
Ο Κωσταντίνος είναι ερευνητής στον τομέα της ασφάλειας πληροφοριών, πραγματοποιώντας τη διδακτορική διατριβή του στο τμήμα Πληροφορικής & Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών. Ταυτόχρονα, είναι μέλος της συντονιστικής επιτροπής της ελληνικής ομάδας εργασίας του OWASP.

**Στη συγγραφή του άρθρου βοήθησαν επίσης οι Μίλτος Κάνδιας και Αλέξης Κοτσιφάκος.*

Η ασφάλεια στους υπολογιστές και στο Διαδίκτυο είναι ένα θέμα για το οποίο τον τελευταίο καιρό ακούμε όλο και περισσότερα. Μάλιστα, συχνά οι συζητήσεις για την ασφάλεια χαρακτηρίζονται από ένα πνεύμα υστερίας και πανικού, καθώς δεν είναι λίγοι εκείνοι που θεωρούν ότι οι επιτήδειοι ανακαλύπτουν ευπάθειες (vulnerabilities) πολύ πιο γρήγορα απ' ό,τι οι προγραμματιστές και οι ειδικοί στην ασφάλεια καταφέρνουν να τις "μπαλώσουν". Σε αυτό εδώ το άρθρο, θα προσπαθήσουμε να δούμε τα πράγματα πιο ψύχραιμα, αναλύοντας τις κυριότερες ευπάθειες που μπορούν να απειλήσουν τις Web εφαρμογές και παρουσιάζοντας δύο open source εργαλεία που μπορούν να χρησιμοποιηθούν για να τις εντοπίσουν. Ολα αυτά από την οπτική γωνία της ομάδας εργασίας του OWASP (Open Web Application Security Project), μίας διεθνούς οργάνωσης για την ασφάλεια των εφαρμογών στο Internet, που τα τελευταία χρόνια δραστηριοποιείται και στη χώρα μας.

Μία λανθασμένη άποψη που επικρατεί στο χώρο της ασφάλειας είναι ότι αφενός αφορά μόνο σε λίγους ειδικούς, αφετέρου είναι κάτι με το οποίο πρέπει κάποιος να ασχοληθεί μία φορά μόνο: λίγες μετατροπές στον κώδικα και το πρόγραμμα είναι ασφαλές για πάντα. Δυστυχώς, όμως, τα πράγματα δεν είναι ακριβώς έτσι. Κατ' αρχάς, η ασφάλεια είναι υπόθεση όλων: αναλυτές, προγραμματιστές, οργανισμοί, αλλά και απλοί χρήστες πρέπει να ενημερώνονται έτσι ώστε καθένας από την πλευρά του να προφυλάσσεται. Ειδικά οι προγραμματιστές σήμερα καλούνται να είναι συνέχεια σε εγρήγορση, καθώς οι εφαρμογές τους, ειδικά αυτές που αναπτύσσονται για Web περιβάλλον, είναι διαρκώς εκτεθειμένες σε κάθε είδους επίθεση.

Επιπλέον, δεν αρκεί να "ασφαλίσει" κάποιος τον κώδικά του την ώρα που τον γράφει και μετά να μείνει για πάντα ήσυχος. Οι ευπάθειες λογισμικού διαρκώς μεταβάλλονται και αυτά που συζητάμε τώρα, σε έναν χρόνο θα έχουν αλλάξει. Νέα προβλήματα θα έχουν προκύψει και ο κώδικας που μέχρι χτες ήταν ασφαλής, θα πρέπει να αλλαχθεί, για να μπορεί είναι ανθεκτικός απέναντι στις νέες επιθέσεις. Ετσι, η δημιουργία ασφαλούς κώδικα αναφέρεται σε όλα τα στάδια της ανάπτυξης: από το σχεδιασμό μέχρι και κατά τη διάρκεια της λειτουργίας.

Στην κατεύθυνση αυτή, το OWASP έχει αναπτύξει το "OWASP Top10", ένα κείμενο που παρουσιάζει τις 10 κυριότερες ευπάθειες των διαδικτυακών εφαρμογών. Στόχος του κειμένου αυτού, που ανανεώνεται ανά τακτά χρονικά διαστήματα, είναι να ενημερώνει και να αφυπνίζει τους προγραμματιστές κυρίως, έτσι ώστε οι εφαρμογές που παράγουν να είναι ασφαλείς στην πράξη. Μάλιστα, το κείμενο αυτό έχει υιοθετηθεί από οργανισμούς και εταιρείες, όπως η Visa, ως απαραίτητη προϋπόθεση για εφαρμογές που έχουν να κάνουν με ηλεκτρονικές συναλλαγές στο Διαδίκτυο.

Τα δέκα χειρότερα!

Ας δούμε, όμως, ποιες είναι, σύμφωνα με το OWASP, οι 10 κυριότερες ευπάθειες από τις οποίες κινδυνεύουν οι εφαρμογές που "τρέχουν" σήμερα στο Διαδίκτυο:

■ **Cross Site Scripting (XSS):** Προβλήματα ασφαλείας με XSS προκύπτουν όταν μία εφαρμογή λαμβάνει δεδομένα που έχει εισάγει κάποιος χρήστης από τον browser, χωρίς το περιεχόμενό τους να έχει ελεγχθεί ή κωδικοποιηθεί. Οι XSS επιθέσεις επιτρέπουν στον επιτιθέμενο να εκτελέσει scripts στον browser του θύματος. Με αυτά τα scripts, ο επιτιθέμενος μπορεί να πάρει τον έλεγχο ανοιχτών sessions του θύματος, να μεταβάλει το περιεχόμενο ιστοσελίδων (γνωστό ως deface), να μεταδώσει worms κ.λπ.

■ **Injection Flaws:** Τα injection flaws (επίθεση με εμβόλιμο κώδικα), ιδίως τα SQL injections, αποτελούν τις πιο συχνές αιτίες για επιθέσεις σε διαδικτυακές εφαρμογές. Η επίθεση με εμβόλιμο κώδικα επιτυγχάνεται όταν τα δεδομένα που εισάγονται από το χρήστη, αποστέλλονται σε κάποιον μεταγλωττιστή ως τμήμα κάποιας εντολής ή ερωτήματος (query). Τα δεδομένα του επιτιθέμενου ξεγελούν το μεταγλωττιστή, ο οποίος εκτελεί εντολές ή ακόμη προχωράει σε αλλαγή των δεδομένων της βάσης.

■ **Malicious File Execution:** Κώδικας τρωτός σε remote file inclusion (RFI) επιτρέπει στον επιτιθέμενο να συμπεριλάβει κακόβουλο κώδικα και δεδομένα που καταλήγουν σε ισχυρές και καταστροφικές επιθέσεις, μέχρι και πλήρη παράδοση του

ΤΙ ΕΙΝΑΙ ΤΟ OWASP

Το OWASP (Open Web Application Security Project - http://www.owasp.org) αποτελεί μία πρωτοβουλία που αποσκοπεί στον εντοπισμό και στην καταπολέμηση των τρωτών σημείων του λογισμικού τέτοιων εφαρμογών. Ως ένας μη κερδοσκοπικός οργανισμός, συμμερίζεται τις ιδέες του ελεύθερου/ανοικτού λογισμικού, παρέχοντας δωρεάν αλλά επαγγελματικής ποιότητας έγγραφα, εργαλεία και πρότυπα.

Παράλληλα, ενισχύει τη διοργάνωση συνεδρίων και τοπικών ομάδων εργασίας (local chapters), τη δημοσίευση άρθρων και συγγραμμάτων, καθώς και την ανταλλαγή απόψεων μέσα από forums και mailing lists. Το OWASP αριθμεί μέλη σε όλο τον πλανήτη, συμπεριλαμβανομένων μεγάλων οργανισμών και εταιρειών του χώρου όπως οι VISA, Deloitte, Unisys, Foundstone και άλλες.

ελέγχου του server. Οι επιθέσεις τύπου malicious file execution είναι συχνές σε PHP, XML και σε κάθε framework που δέχεται ονόματα αρχείων ή αρχεία από τους χρήστες.

■ **Insecure Direct Object Reference:** Επιθέσεις τύπου Insecure Direct Object Reference προκύπτουν όταν ο προγραμματιστής αφήνει εκτεθειμένη κάποια αναφορά σε αντικείμενο εσωτερικής εμβέλειας και εφαρμογής, όπως κάποιο αρχείο, ένας κατάλογος, μία εγγραφή σε βάση δεδομένων ή κάποιο κλειδί όπως ένα URL. Ο επιτιθέμενος μπορεί να πάρει τον έλεγχο αυτών των αναφορών και να έχει πρόσβαση σε άλλα αντικείμενα χωρίς εξουσιοδότηση.

■ **Cross Site Request Forgery (CSRF):** Με επίθεση τύπου CSRF, ο επιτιθέμενος εκμεταλλεύεται τον browser του θύματος για να στείλει κάποιο pre-authenticated request σε μία ευάλωτη δικτυακή εφαρμογή, το οποίο μετά υποχρεώνει τον browser να προβεί σε ενέργειες, σύμφωνα με τις εντολές του επιτιθέμενου. Οι επιδράσεις των CSRF επιθέσεων είναι ανάλογες των δυνατοτήτων των εφαρμογών εναντίον των οποίων πραγματοποιούνται οι επιθέσεις.

■ **Broken Authentication and Session Management:** Τα στοιχεία των λογαριασμών σε εφαρμογές και οι ανοιχτές συνδέσεις (session tokens) συχνά δεν προστατεύονται όπως θα έπρεπε. Οι επιτιθέμενοι αποκτούν κωδικούς, κλειδιά ή authentication tokens, με τελικό στόχο να "κλέψουν την ταυτότητα" άλλων χρηστών (identity thefts).

■ **Insecure Cryptographic Storage:** Σπάνια οι διαδικτυακές εφαρμογές χρησιμοποιούν σωστά τις λειτουργίες κρυπτογράφησης, ούτως ώστε να προστατεύσουν τα δεδομένα και τα στοιχεία των χρηστών (credentials). Ο επιτιθέμενος στην περίπτωση αυτή χρησιμοποιεί δεδομένα που δεν προστατεύονται αρκετά για να πραγματοποιήσει "κλοπή ταυτότητας" και άλλα εγκλήματα, όπως απάτες μέσω πιστωτικών καρτών.

■ **Insecure Communications:** Οι εφαρμογές συχνά αποτυγχάνουν να κρυπτογραφήσουν την κίνηση που πραγματοποιείται στο Διαδίκτυο, όποτε χρειάζεται να προστατευτούν ευαίσθητες συνδέσεις και επικοινωνίες.

■ **Failure to Restrict URL Access:** Συχνά, οι εφαρμογές προστατεύουν ευαίσθητες λειτουργίες τους, αποκρύπτοντας τα URLs από μη εξουσιοδοτημένους χρήστες. Ο επιτιθέμενος μπορεί να εκμεταλλευτεί αυτή την αδυναμία για να πάρει πρόσβαση στα σημεία αυτά και να προχωρήσει σε λειτουργίες για τις οποίες δεν έχει εξουσιοδοτηθεί.

Προφανώς, οι ευπάθειες αυτές δεν είναι τα μόνα σημεία από τα οποία μπορεί να κινδυνεύει ένα Web site ή μία εφαρμογή στο Διαδίκτυο. Αποτελούν, όμως, τα σημαντικότερα τρωτά σημεία που πρέπει να γνωρίζουν και να λαμβάνουν υπόψη τους όσοι αναπτύσσουν λογισμικό τέτοιου είδους. Και στη συγκεκριμένη περίπτωση δεν αρκεί μόνο η πρόληψη και η ενημέρωση. Συχνά πρέπει εκ των υστέρων, σε μία εφαρμογή που ήδη τρέχει, να εντοπιστούν ευπάθειες που δεν αντιμετωπίστηκαν σωστά ή που εμφανίστηκαν εκ των υστέρων. Στην κατεύθυνση αυτή, το OWASP αναπτύσσει δύο εργαλεία λογισμικού ανοιχτού κώδικα, που βοηθούν τον αναλυτή στον εντοπισμό αυτών των ευπαθειών. Πρόκειται για το OWASP Web Scarab, το οποίο αυτό τον καιρό βρίσκεται στο στάδιο της ανανέωσης, και το WVS (από τα αρχικά Web Vulnerability Scanner), που αποτελεί μία πρωτοβουλία της ελληνικής ομάδας εργασίας.

Το WebScarab **(εικόνα 1)** είναι ένα πλαίσιο εργασίας που μπορεί να χρησιμοποιηθεί για την ανάλυση εφαρμογών που χρησιμοποιούν τα πρωτόκολλα HTTP και HTTPS για επικοινωνία. Εχει γραφτεί σε γλώσσα Java, έτσι ώστε να μπορεί να χρησιμοποιηθεί σε πολλές διαφορετικές πλατφόρμες. Ουσιαστικά, λειτουργεί ως ενδιάμεσος proxy, που παρατηρεί τα μηνύματα που μεταδίδονται από τον αναλυτή προς τον server. Ο χειριστής του WebScarab μπορεί να καθορίσει ακριβώς τη μορφή των αιτήσεων που θα σταλούν προς τον server και στη συνέχεια να παρατηρήσει τις απαντήσεις που αυτός δίνει ή και να τις τροποποιήσει έτσι ώστε να δει πώς θα ανταποκριθεί ο browser, αλλά και το λειτουργικό σύστημα γενικότερα. Πέρα από αυτή τη δυνατότητα, υπάρχει μία πληθώρα από plug-ins που έχουν αναπτυχθεί και δίνουν πολλές άλλες δυνατότητες στο συγκεκριμένο εργαλείο. Αυτό τον καιρό, δημιουργείται μία νέα έκδοση του προγράμματος, με στόχο να γίνει πιο φιλικό προς το χρήστη. Το WebScarab είναι διαθέσιμο δωρεάν από τη διεύθυνση http://www.owasp.org/index.php/OWASP_WebScarab_Project.

WVS

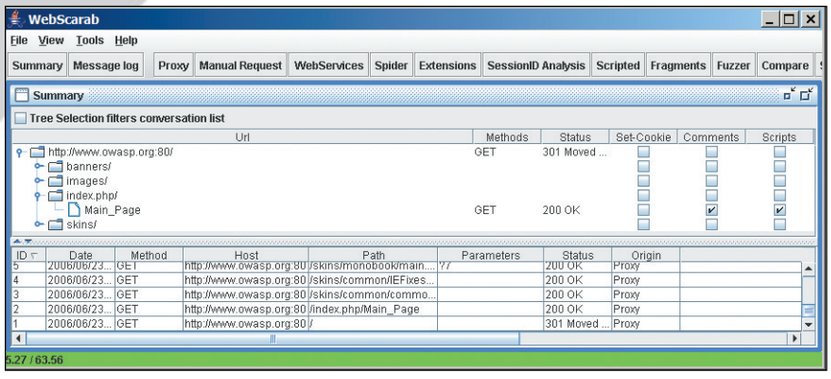
Το WVS (Web Vulnerability Scanner) αποτελεί πρωτοβουλία μίας ομάδας φοιτητών του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών, κάτω από την ομπρέλα της ελληνικής ομάδας εργασίας του OWASP. Πρόκειται για ένα open source εργαλείο, το οποίο κυκλοφορεί υπό την άδεια χρήσης GPL και απευθύνεται στον αναλυτή ασφαλείας, στον penetration tester ή ακόμη και στον Web developer που έχει ολοκληρώσει μία ιστοσελίδα και θέλει να την ελέγξει για προβλήματα ασφαλείας **(εικόνα 2).** Δεν είναι σε καμία περίπτωση ένα κακόβουλο εργαλείο και αυτό φαίνεται τόσο από τον τρόπο λειτουργίας όσο και από τη φιλοσοφία του.

Το εργαλείο ενσωματώνει το δικό του μηχανισμό επικοινωνίας με τον server και ως επί το πλείστον προσπαθεί να εντοπίσει την ύπαρξη ευπαθών καταστάσεων, χρησιμοποιώντας μία βάση δεδομένων με ευπάθειες και μία ευρεία γκάμα άλλων μεθόδων, όπως θα εξηγήσουμε παρακάτω. Πρέπει να σημειωθεί ότι ο WVS ελέγχει μόνο την ιστοσελίδα και όχι τον server που την υποστηρίζει. Συνεπώς, λόγω αυτής της φιλοσοφίας δεν κρίνεται απαραίτητη η υλοποίηση κάποιας τεχνικής παράκαμψης των IDS/IPS (δείτε πλαίσιο "Ορολογία"), τουλάχιστον σε αυτό το στάδιο.

Ο WVS οργανώνεται σε τρία επίπεδα και έχει σχεδιαστεί ώστε να λειτουργεί με έναν εξαιρετικά απλό και αξιόπιστο τρόπο, ώστε να επιτευχθεί η μέγιστη δυνατή ταχύτητα και κατ' επέκταση ακρίβεια **(δείτε σχήμα 1).** Στο πρώτο επίπεδο βρίσκεται η βάση δεδομένων SQlite, που αποτελεί το θεμέλιο λίθο της λειτουργίας, καθώς εκεί είναι αποθηκευμένες οι περισσότερες από τις ευπάθειες για τις οποίες θα ψάξει ο scanner. Επίσης, εδώ βρίσκεται και το API για την ανάκτηση των δεδομένων της βάσης και την ανανέωσή της με νέες ευπάθειες. Στο δεύτερο επίπεδο βρίσκεται το API για την επικοινωνία με τον server, το οποίο χρησιμοποιεί τις κατάλληλες μεθόδους επικοινωνίας (GET, POST, HEAD) ανάλογα με την εκάστοτε ευπάθεια. Οι απαντήσεις που επιστρέφονται από τις αιτήσεις στον server, τροφοδοτούν το τρίτο και τελευταίο επίπεδο, το οποίο διαχειρίζεται τα δύο προαναφερθέντα και είναι υπεύθυνο για την παρουσίαση των αποτελεσμάτων στο χρήστη ανάλογα με την επιλογή του (κονσόλα με plain text, γραφικό περιβάλλον και μελ-

Η ΕΛΛΗΝΙΚΗ ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ ΤΟΥ OWASP

Η ελληνική ομάδα εργασίας του OWASP (http://www.owasp.gr) δημιουργήθηκε το 2005, με κύριο στόχο την ενημέρωση και την αφύπνιση της ελληνικής κοινότητας αναφορικά με τους κινδύνους ασφαλείας στις διαδικτυακές εφαρμογές. Αφορμή για τη δημιουργία της αποτέλεσαν ουσιαστικά τα ολοένα αυξανόμενα περιστατικά ασφαλείας στο Διαδίκτυο, όπως τα κρούσματα phishing σε ελληνικές τράπεζες. Σήμερα, η ελληνική ομάδα του OWASP δραστηριοποιείται σε προγράμματα ελεύθερου/ανοικτού λογισμικού, καθώς και σε μεταφράσεις κειμένων του OWASP στα Ελληνικά, προωθώντας την ιδέα του σε τοπικό επίπεδο. Παράλληλα, μέσα από τη mailing list της ενημερώνει και προκαλεί συζητήσεις σχετικά με επίκαιρα θέματα ασφαλείας στο Διαδίκτυο, ενώ εκδίδει και μηνιαίο newsletter. Επιπλέον, διοργανώνει συναντήσεις και συμμετέχει σε συνέδρια, με στόχο κυρίως την ενημέρωση και την ευαισθητοποίηση γύρω από τα θέματα ασφαλείας.



Εικόνα 1: Το OWASP WebScarab λειτουργεί ως proxy, ελέγχοντας τα μηνύματα που ανταλλάσσει ο browser με τον server.

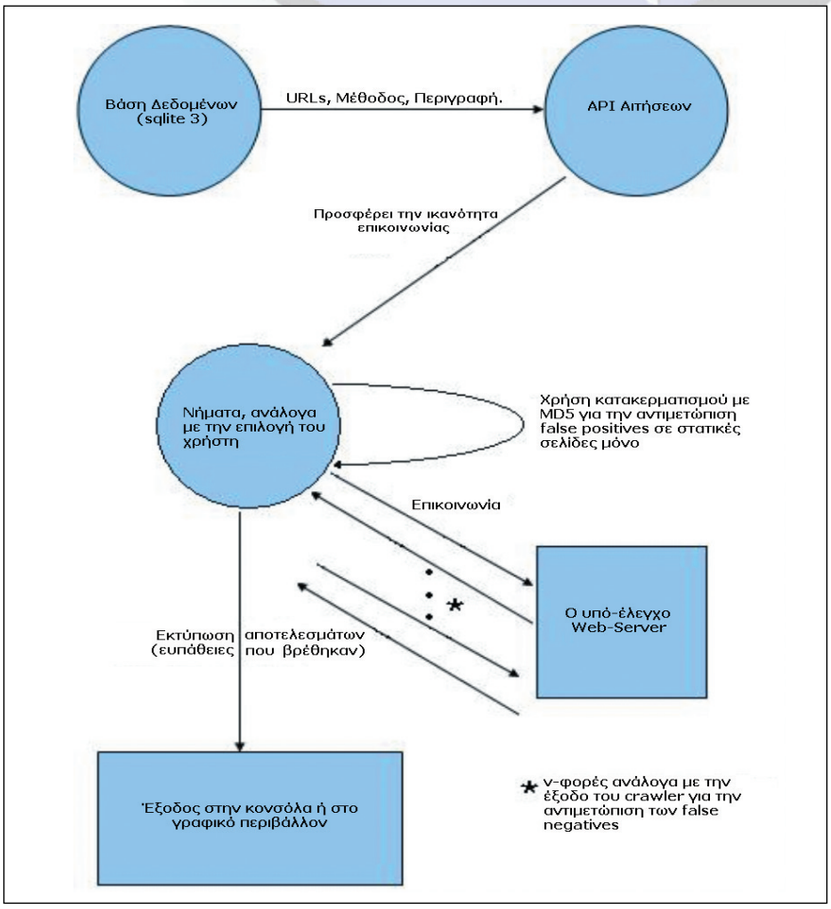
TOP 10

Οι 10 κυριότερες ευπάθειες
Αυτές είναι οι κυριότερες αιτίες ή απαρχές επιθέσεων στα Web sites...
1. Cross Site Scripting (XSS)
2. Injection Flaws
3. Malicious File Execution
4. Insecure Direct Object Reference
5. Cross Site Request Forgery (CSRF)
6. Information Leakage and Improper Error Handling
7. Broken Authentication and Session Management
8. Insecure Cryptographic Storage
9. Insecure Communications
10. Failure to Restrict URL Access

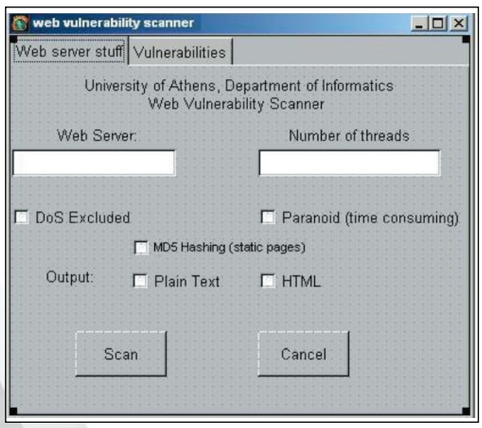
λοντικά html και odt). Θα πρέπει να σημειωθεί ότι το τελευταίο επίπεδο είναι multi-threaded (posix threads), με τον αριθμό των threads να καθορίζεται από το χρήστη μέχρι το όριο των 50. Ο λόγος για τον οποίο η επιλογή του αριθμού των threads επαφίεται στο χρήστη, είναι διότι υποθέτουμε ότι εκείνος γνωρίζει τις συνθήκες κάτω από τις οποίες γίνεται ο έλεγχος και είναι σε θέση να κρίνει τα περιθώρια επιλογής. Για παράδειγμα, αν ελέγχει τη δική του ιστοσελίδα από το εσωτερικό δίκτυο, μπορεί να ενεργοποιήσει το μέγιστο αριθμό, ενώ αν ελέγχει μία πολύ σημαντική ιστοσελίδα στο Internet, τότε είναι πιθανό να επιλέξει τον ελάχιστο αριθμό, ώστε να αποφευχθεί υπερφόρτωση του δικτύου.

Ενα σημαντικό πρόβλημα των εργαλείων αυτής της κατηγορίας είναι τα λεγόμενα false positives και false negatives (δείτε πλαίσιο "Ορολογία"). Στο WVS έγινε προσπάθεια να ελαχιστοποιηθούν με τη χρήση εναλλακτικών μεθόδων ελέγχου και πιστοποίησης ευπαθειών. Αυτές οι μέθοδοι, σε συνδυασμό με τις ευπάθειες της βάσης δεδομένων, καθιστούν το τεστ πολύ πιο αποτελεσματικό και ακριβές. Για την αποφυγή των false negatives, υλοποιήθηκε το λεγόμενο paranoid scanning, το οποίο ακολουθεί την εξής διαδικασία: Αρχικά προσπαθεί να εντοπίσει με τη μέγιστη δυνατή ακρίβεια τη δομή της προς έλεγχο ιστοσελίδας, δηλαδή, όλους τους καταλόγους, και στη συνέχεια πραγματοποιεί όλους τους δυνατούς ελέγχους για καθέναν από αυτούς. Αυτό μας βεβαιώνει ότι θα εντοπιστούν ευπάθειες πακέτων λογισμικού, τα οποία έχουν εγκατασταθεί σε διαφορετικό κατάλογο από αυτόν που προτείνει ο κατασκευαστής. Για την αποφυγή των false positives, έχει προς το παρόν υλοποιηθεί ένα σύστημα που χρησιμοποιεί md5 hashing, το οποίο εγγυάται την ελαχιστοποίησή τους σε ελέγχους πάνω σε στατικές σελίδες.

Ενα εξίσου σημαντικό ζήτημα με τα παραπάνω είναι και η



Σχήμα 1: Η δομή του WVS, χάρη στην οποία λειτουργεί γρήγορα και αποτελεσματικά.



Εικόνα 2: Το WVS δίνει τη δυνατότητα να ελεγχθεί ένα site για γνωστά προβλήματα ασφαλείας, ελαχιστοποιώντας τις περιπτώσεις false positives και false negatives.

δυνατότητα για portability του WVS σε πολλά λειτουργικά συστήματα. Γι' αυτό το λόγο, οι τεχνολογίες που χρησιμοποιήθηκαν, είναι η βάση δεδομένων SQlite, η βιβλιοθήκη libcurl για την επικοινωνία με τον server, καθώς και το POSIX πρότυπο για τα threads. Τα δύο πρώτα είναι portable σε όλα τα λειτουργικά συστήματα, ενώ τα POSIX threads δεν είναι σίγουρο πως μπορούν να λειτουργήσουν σε περιβάλλον MS Windows, κάτι που αποτελεί αντικείμενο έρευνας που γίνεται αυτή τη στιγμή.

Επιπλέον, η έρευνα προσανατολίζεται στην ελαχιστοποίηση των false positives για ιστοσελίδες δυναμικού περιεχομένου, στην ολοκλήρωση μεθοδολογιών για χρήση των προτύπων ελέγχου για SQL injections και cross site scripting και, τέλος, στην ολοκλήρωση των μεθόδων fuzzing στις μεταβλητές της ιστοσελίδας που έχουν ανακαλυφθεί από τμήμα της διαδικασίας paranoid scanning που αναφέρθηκε παραπάνω.

Επίλογος

Οπως αναφέραμε και στην αρχή, η ασφάλεια των διαδικτυακών εφαρμογών είναι ένα σύνθετο ζήτημα που δεν μπορεί να αντιμετωπιστεί μεμονωμένα και μετά να ξεχαστεί. Απαιτεί συνεχή ενημέρωση και εγρήγορση, έτσι ώστε να αντιμετωπίζονται εγκαίρως οι νέες απειλές που συχνά εμφανίζονται. Η ελληνική ομάδα εργασίας του OWASP καταβάλλει συνεχή προσπάθεια για την προώθηση στην Ελλάδα της φιλοσοφίας του τόσο μέσα από την ενημέρωση σχετικά με θέματα ασφαλείας όσο και με την παροχή λογισμικού υψηλής ποιότητας σε προγραμματιστές και αναλυτές ασφαλείας. Σύντομα, ολοκληρώνεται η μετάφραση της δεύτερης έκδοσης OWASP Top10 (2007), ενώ στα άμεσα σχέδια είναι και εξελληνισμός του WebScarab. ■

ΟΡΟΛΟΓΙΑ

IDS/IPS: Intrusion Detection System/Intrusion Prevention System. Λογισμικό/υλικό που εντοπίζει(IDS)/αποτρέπει(IPS) και καταγράφει ακατάλληλη, εσφαλμένη ή μη φυσιολογική δραστηριότητα.
False Positive: Προκύπτει όταν ένα σύστημα απαντά θετικά στην ύπαρξη μίας ευπάθειας, ενώ στην πραγματικότητα δεν υπάρχει.
False Negative: Προκύπτει όταν ένα σύστημα απαντά αρνητικά στην ύπαρξη μίας ευπάθειας, ενώ στην πραγματικότητα υπάρχει.
Fuzzing: Είναι μία τεχνική ελέγχου λογισμικού που παρέχει τυχαία δεδομένα ("fuzz") ως είσοδο σε κάποιο πρόγραμμα.