**ΑΣΦΑΛΕΙΑ ΣΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ**

**ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ ΕΡΓΑΣΤΗΡΙΟΥ**

**Ονοματεπώνυμο:** Φίλιππος Γεωργάντζος

**Τμήμα:** AΣΦ02(Δευτέρα 11:00-13:00)

**Ημερομηνία :** 11/6/20

**Εξάμηνο:** 10o

ΘΕΜA: Βιβλιογραφική έρευνα και αναλυτική παρουσίαση ευπάθειας ασφάλειας και σχετικών περιστατικών ψηφιακών επιθέσεων.

ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΓΑΣΙΑΣ

Κοινή Αδυναμία: CWE 121 (Stack-based Buffer Overflow)

Κοινές ευπάθειες: CVE 2019-5618

CVE 2019-5619

CVE 2019-5621



**ΠΕΡΙΕΧΟΜΕΝΑ**

**Σελίδα 3:** Εισαγωγή

**Σελίδα 4:** Περιγραφή των CWE και CVE

**Σελίδα 5:** Σύνδεση με 1η εργαστηριακή άσκηση (Buffer Overflow Attack)

**Σελίδα 6:** Τεχνική περιγραφή με παραδείγματα

**Σελίδα 9:** Περιστατικά αδυναμίας (CWE-121)

**Σελίδα 10:** Συνέπειες της αδυναμίας CWE-121 (Stack-based Buffer Overflow)

**Σελίδα 11:** Πιθανά αντίμετρα της αδυναμίας

**Σελίδα 13:** Βιβλιογραφία

**ΕΙΣΑΓΩΓΗ**

Το **Common Weakness Enumeration (CWE)** είναι ένα σύστημα κατηγοριών για αδυναμίες και ευπάθειες λογισμικού. Υποστηρίζεται από ένα κοινοτικό έργο με στόχο την κατανόηση των ελαττωμάτων στο λογισμικό και τη δημιουργία αυτοματοποιημένων εργαλείων που μπορούν να χρησιμοποιηθούν για τον εντοπισμό, τη διόρθωση και την πρόληψη αυτών των ελαττωμάτων. Όλα αυτά τα ελαττώματα αν δεν αντιμετωπιστούν με την εμφάνιση τους τότε μπορεί να δημιουργήσουν σοβαρά προβλήματα σε δίκτυα,συστήματα ή υλικά που δεν περιέχουν ισχυρή ασφάλεια και είναι ευάλωτα σε επιθέσεις. Ο κύριος σκοπός λοιπόν του CWE είναι να εξαλείψει τα τρωτά σημεία στην πηγή κάποιου συστήματος και τα πιο κοινά λάθη πριν από την παράδοση υλικού και λογισμικού.Γενικά, βοηθάει στην αποτροπή των ειδών ευπάθειας και αδυναμιών ασφαλείας που θέτουν εταιρίες σε κίνδυνο έκθεσης των δεδομένων και μολύνουν τις βιομηχανίες υλικού και λογισμικού. Για όλους αυτούς τους λόγους δημιουργήθηκε και αναπτύχθηκε από ερευνητές ασφάλειας η λίστα των CWE και διατηρείται επί του παρόντος από τον οργανισμό MITRE με την υποστήριξη του Εθνικού Τμήματος Ασφάλειας στον Κυβερνοχώρο (DHS).

Όσον αφορά τo **Common Vulnerabilities and Exposures (CVE)** κυκλοφόρησε το 1999 όταν τα περισσότερα εργαλεία ασφάλειας στον κυβερνοχώρο χρησιμοποίησαν τις δικές τους βάσεις δεδομένων με τα δικά τους ονόματα για ευπάθειες ασφαλείας. Εκείνη την εποχή υπήρχε σημαντική διακύμανση μεταξύ των προϊόντων και κανένας τρόπος να καθοριστεί πότε οι διαφορετικές βάσεις δεδομένων αναφερόταν στο ίδιο πρόβλημα. Οι συνέπειες ήταν πιθανά κενά στην κάλυψη ασφάλειας και καμία αποτελεσματική διαλειτουργικότητα μεταξύ των διαφορετικών βάσεων δεδομένων και εργαλείων. Τo CVE λοιπόν, που διατηρείται από τον οργανισμό MITRE, βοηθά τα προιόντα και τα συστήματα που είναι συμβατά να έχουν βελτιωμένη ασφάλεια και μεγαλύτερη κάλυψη.

Σε αυτήν την εργασία θα γίνει αναλυτική περιγραφή της κοινής αδυναμίας(CWE) και των κοινών ευπαθειών και εκθέσεων (CVE’s) που έχω επιλέξει μετά από σχετική έρευνα. Mε κριτήρια λοιπόν τις διαθέσιμες κοινές αδυναμίες, των πηγών που υπάρχουν και την σύνδεση αυτής της αδυναμίας με τη θεματολογία των εργαστηριακών ασκήσεων που έχουμε υλοποιήσει, έχω επιλέξει το CWE-121 που έχει ως θέμα το Stack-based Buffer Overflow και συνδέεται με την 1η εργαστηριακή άσκηση που έχει πραγματοποιηθεί (Buffer Overflow Attack). Tα σχετιζόμενα CVE’s που θα αναλύσουμε στην συνέχεια είναι τα παρακάτω:

* CVE 2019-5618
* CVE 2019-5619
* CVE 2019-5621

**ΠΕΡΙΓΡΑΦΗ ΤΩΝ CWE ΚΑΙ CVE**

CWE-121 (Stack-based Buffer Overflow)

Ως Buffer Overflow ορίζεται μια κατάσταση υπερχείλησης στοίβας όπου ένα πρόγραμμα ενώ γράφει δεδομένα σε ένα buffer, υπερβαίνει το όριο του και αντικαθιστά τις κοντινές θέσεις μνήμης. Τα buffer χρησιμοποιούνται ως θέσεις μνήμης για την αποθήκευση των δεδομένων. Όταν βασίζεται σε στοίβα τότε το buffer που αντικαθιστάται εκχωρείται στην στοίβα. Oι υπερχειλήσεις buffer χρησιμοποιούνται συχνά για την εκτέλεση αυθαίρετου κώδικα, οποίος δεν εμπίπτει τόσο στο πεδίο εφαρμογής της απεριόριστης πολιτικής ασφάλειας ενός προγράμματος. Η εκμετάλλευση της συμπεριφοράς της υπερχείλισης buffer είναι ένα πολύ γνωστό πλεονέκτημα ασφαλείας . Αυτή η αδυναμία που αναλύουμε εμφανίζεται κατά τις φάσεις της Αρχιτεκτονικής, του Σχεδιασμού και της Υλοποίησης.

CVE 2019-5618

Η μια από τις ευπάθειες με τις οποίες θα ασχοληθούμε είναι η CVE-2019-5618 και κυκλοφόρησε στις 29/4/2020. Η συγκεκριμένη ευπάθεια έχει ταξινομηθεί ως κρίσιμη και έχει βρεθεί στο A-PDF WAV σε MP3 1.0.0. Είναι συσχετιζόμενη με την αδυναμία CWE-121. Αυτή η ευπάθεια επηρεάζει κάποια άγνωστη λειτουργικότητα. Ο χειρισμός με μια άγνωστη είσοδο οδηγεί σε ευπάθεια στη διαφθορά μνήμης (βάσει στοίβας). Η επίθεση μπορεί να ξεκινήσει από απόσταση. Δεν απαιτείται μορφή ελέγχου ταυτότητας για μια επιτυχημένη εκμετάλλευση.

CVE 2019-5619

Πρόκειται για άλλη μια κρίσιμη ευπάθεια που σχετίζεται με την αδυναμία CWE-121, κυκλοφόρησε και αυτή στο ίδιο διάστημα και έχει βρεθεί στο AASync 2.2.1.0 . Έχει την ίδια λειτουργικότητα με το παραπάνω CVE.

CVE 2019-5621

Πρόκειται για άλλη μια κρίσιμη ευπάθεια που σχετίζεται με την αδυναμία CWE-121, κυκλοφόρησε και αυτή στο ίδιο διάστημα και έχει βρεθεί στο ABBS Software Audio Media Player 3.1 ( Multimedia Player Software ). Έχει την ίδια λειτουργικότητα με τα παραπάνω CVE. Η επιρροή από αυτήν την ευπάθεια είναι ένα άγνωστο μπλοκ κώδικα.

**ΣΥΝΔΕΣΗ ΜΕ 1Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ (Buffer Overflow Attack)**

Το CWE-121 που έχω επιλέξει να αναλύσω, έχει σύνδεση με την θεματολογία της 1ης εργαστηριακής άσκησης που είχαμε υλοποιήσει αυτό το εξάμηνο με τίτλο “Buffer Overflow Attack”. Όπως και στην συγκεκριμένη ενότητα έτσι και στο CWE-121, αυτή η κατάσταση βασίζεται σε στοίβα οπότε όλα τα δεδομένα αποθηκεύονται εκεί και κάθε φορά που εκτελείται μια συνάρτηση, διατίθεται χώρος στην στοίβα για να εκτελεστεί η λειτουργία. Οι υπερχειλίσεις buffer γενικά οδηγούν σε διακοπές. Επίσης, μπορεί συχνά να επιτρέψουν την εκτέλεση αυθαίρετου-κακόβουλου κώδικα που επιτρέπει την εκτέλεση περισσότερων εντολών για να πάρει πρόσβαση στο σύστημα. Οι υπερχειλίσεις buffer που βασίζονται σε στοίβα μπορούν να προκαλέσουν αντικατάσταση δεικτών στοίβας ή αντικατάσταση διευθύνσεων επιστροφής. Οι γλώσσες προγραμματισμού που συσχετίζονται συνήθως με υπερχείλιση buffer περιλαμβάνουν C και C ++ , οι οποίες δεν παρέχουν ενσωματωμένη προστασία έναντι πρόσβασης ή αντικατάστασης δεδομένων σε οποιοδήποτε μέρος της μνήμης και δεν ελέγχουν αυτόματα ότι τα δεδομένα που είναι γραμμένα σε έναν πίνακα βρίσκονται εντός τα όρια αυτού του πίνακα.



**ΤΕΧΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΠΑΡΑΔΕΙΓΜΑΤΑ**

Μια υπερχείλιση buffer (Buffer Overflow) πραγματοποιείται κάθε φορά που τα δεδομένα που γράφονται σε μια περιοχή μνήμης buffer αλλοιώνουν τις τιμές δεδομένων σε διευθύνσεις μνήμης που είναι κοντά στο buffer προορισμού λόγω ανεπαρκούς ελέγχου.Αυτή η περίπτωση συμβαίνει όταν αντιγράφονται δεδομένα από ένα buffer σε άλλο χωρίς πρώτα να γίνεται έλεγχος αν ταιριάζουν τα δεδομένα στο buffer προορισμού. Γενικά σε μια στοίβα εκτέλεσης υπάρχουν πολλά δεδομένα που έχουν την δυνατότητα να οδηγήσουν σε εκτέλεση ενός κακόβουλου κώδικα. Η Return address είναι η πιο εμφανής ένδειξη και μόλις ολοκληρωθεί η τρέχουσα λειτουργία τότε η εκτέλεση θα συνεχιστεί από αυτήν την διεύθυνση μνήμης. Βρίσκοντας λοιπόν την απόσταση μεταξύ αυτής της διεύθυνσης επιστροφής και της βάσης του buffer, καθώς και την διεύθυνση τοποθέτησης του shellcode μπορούμε να δημιουργήσουμε ένα badfile που θα περιέχει τον κακόβουλο κώδικα για να πραγματοποιηθεί η επίθεση. Ο εισβολέας μπορεί να αντικαταστήσει λοιπόν την διεύθυνση επιστροφής με κάποια άλλη διεύθυνση μνήμης που έχει πρόσβαση για εγγραφή στην οποία τοποθετείται ο αυθαίρετος κώδικας ή εναλλακτικά μπορεί να χρησιμοποιήσει την διεύθυνση μιας σημαντικής κλήσης μιας συνάρτησης.

Η αδυναμία CWE-121 περιέχει διάφορους τρόπους εισαγωγής και αυτές γίνονται στην φάση της Αρχιτεκτονικής και Σχεδιασμού και στην φάση της Εκτέλεσης. Ενώ η πιθανότητα εκμετάλλευσης της είναι υψηλή. Δυο αδυναμίες και κατηγορίες υψηλού επιπέδου που δίνουν πληροφορίες για παρόμοια αντικείμενα που υπάρχουν σε υψηλότερα επίπεδα αφαίρεσης είναι το CWE-787 τύπου Β με όνομα «εκτός ορίων εγγραφή» και το CWE-788 με όνομα «Πρόσβαση στη θέση μνήμης μετά το τέλος του buffer».

ΠΑΡΑΔΕΙΓΜΑΤΑ

Ένα παράδειγμα υπερχείλισης buffer σε γλώσσα C που βασίζεται σε στοίβα είναι το παρακάτω :

**#define BUFSIZE 256**

int main (int argc, char \*\* argv) {

char buf [BUFSIZE];

strcpy (buf, argv [1]);

}

Σε αυτό το παράδειγμα έχουμε δημιουργήσει έναν πίνακα χαρακτήρων buf στον οποίον γίνεται με την συνάρτηση strcpy αντιγραφή και αποθήκευση της συμβολοσειράς argv [1]. Επίσης, δεν είναι σίγουρο ότι η συμβολοσειρά που βρίσκεται στην argv [1] δεν θα υπερβεί αυτό το μέγεθος και θα προκαλέσει υπερχείλιση.

Ένα δεύτερο παράδειγμα πάλι σε γλώσσα C είναι το παρακάτω όπου έχοντας ως δεδομένο μια διεύθυνση ΙP ενός χρήστη, επαληθεύεται η εγκυρότητα της και αφού βρεθεί μετά το όνομα του κεντρικού υπολογιστή τότε αυτό αντιγράφεται σε ένα buffer:

void host\_lookup(char \*user\_supplied\_addr){

struct hostent \*hp;  
in\_addr\_t \*addr;  
char hostname[64];  
in\_addr\_t inet\_addr(const char \*cp);  
  
*/\*routine that ensures user\_supplied\_addr is in the right format for conversion \*/*  
  
validate\_addr\_form(user\_supplied\_addr);  
addr = inet\_addr(user\_supplied\_addr);  
hp = gethostbyaddr( addr, sizeof(struct in\_addr), AF\_INET);  
strcpy(hostname, hp->h\_name);

}

Η συνάρτηση host\_lookup εκχώρει μια διεύθυνση μνήμης buffer 64 byte για την αποθήκευση του ονόματος του κεντρικού υπολογιστή, χωρίς να είναι σίγουρο ότι δεν θα είναι μεγαλύτερο από αυτά τα byte. Στην συνέχεια έχουμε μια ρουτίνα που εξασφαλίζει ότι η διεύθυνση του χρήστη (user\_supplied\_adrr ) είναι στην κατάλληλη μορφή για μετατροπή. Έπειτα εκχωρείται στην μεταβλητή hp η διεύθυνση του address του host με οικογένεια διευθύνσεων AF\_INET για να προσδιοριστεί ο τύπος υποδοχής της διευθυνσιοδότησης για την εφαρμογή της επικοινωνίας. Τέλος γίνεται, με την βοήθεια της μεταβλητής hp, αντιγραφή του ονόματος του κεντρικού υπολογιστή στον πίνακα hostname. Ένα κάποιος εισβολέας καθορίσει αυτή την διεύθυνση, που καταλήγει σε πολύ μεγάλο όνομα κεντρικού υπολογιστή, τότε ενδέχεται να αντικαταστήσουμε πολύ ευαίσθητα δεδομένα.

Σημειώστε ότι αυτό το παράδειγμα περιέχει επίσης μια μη ελεγμένη τιμή επιστροφής ( [CWE-252](https://cwe.mitre.org/data/definitions/252.html) ) που μπορεί να οδηγήσει σε διαφορά αναφοράς δείκτη NULL ( [CWE-476](https://cwe.mitre.org/data/definitions/476.html) ).

Τέλος έχουμε ένα παράδειγμα πάλι γραμμένο σε γλώσσα C που δημιουργείται υπερχείλιση στοίβας (Stack Overflow) με off-by-one error:

**#include <stdio.h>**

**#define N 8**

int main(int argc, char\*\* argv) {

int sum = 0;

int numbers[N]; // fill as { 1, 2, 3, 4, 5 }

for (int i=0; i < N; i++)

numbers[i] = i+1;

for (int i = 0; i <= N; i++)

sum += numbers[i];

printf("Sum=%d\n", sum);

return 0;

}

Η αναμενόμενη συμπεριφορά είναι απροσδιόριστη. Ανάλογα προγράμματα γραμμένα σε γλώσσες ασφαλείς στη μνήμη θα δημιουργήσει μια εξαίρεση χρόνου εκτέλεσης του προγράμματος που σηματοδοτεί την μη έγκυρη πρόσβαση σε πίνακα.

Υπάρχει υπερχείλιση στοίβας στην πρόσβαση στον αριθμό, δεδομένου ότι είναι τοπικές μεταβλητές που κατανέμονται στη στοίβα.

**ΠΕΡΙΣΤΑΤΙΚΑ ΑΔΥΝΑΜΙΑΣ (CWE-121)**

Παρακάτω θα αναλύσουμε κάποιες τις τελευταίες ευπάθειες της αδυναμίας CWE-121:

*Απομακρυσμένη εκτέλεση κώδικα στον κόμβο Advantech WebAccess*

Το συγκεκριμένο περισταστικό διαδραματίστηκε στο ευάλωτο λογισμικό WebAccess Scada Node 8.4.4 με υψηλή δριμύτητα και σοβαρότητα. Ως αναγνωριστικό CVE (κοινή ευπάθεια) υπήρχε το CVE-2020-12019. Αυτή η υπερχείλιση buffer βάσει στοίβας έγινε χωρις αξιοποίηση διαθεσιμότητας και η επιτυχής εκμετάλλευση αυτής της ευπάθειας μπορεί να οδηγήσει σε πλήρη συμβιβασμό του ευάλωτου συστήματος.

*Πολλαπλές ευπάθειες στο Huawei E6878-370*

Το συγκεκριμένο περισταστικό διαδραματίστηκε στο ευάλωτο λογισμικό WebAccess Huawei E6878-370 με χαμηλή δριμύτητα και σοβαρότητα. Ως αναγνωριστικό CVE (κοινή ευπάθεια) υπήρχε το CVE-2020-1832. Αυτή η υπερχείλιση buffer βάσει στοίβας έγινε χωρις αξιοποίηση διαθεσιμότητας και η επιτυχής εκμετάλλευση αυτής της ευπάθειας μπορεί να οδηγήσει σε πλήρη συμβιβασμό του ευάλωτου συστήματος. Ύστερα η Huawei κυκλοφόρησε ενημερώσεις λογισμικού για να διορθώσει αυτήν την ευπάθεια.

*Απομακρυσμένη εκτέλεση κώδικα στο Adobe Character Animator*

Το συγκεκριμένο περισταστικό διαδραματίστηκε στο ευάλωτο λογισμικό Character Animator 2020 με υψηλή και κρίσιμη δριμύτητα και σοβαρότητα. Ως αναγνωριστικό CVE (κοινή ευπάθεια) υπήρχε το CVE-2020-9586. Αυτή η υπερχείλιση buffer βάσει στοίβας έγινε χωρις αξιοποίηση διαθεσιμότητας και η επιτυχής εκμετάλλευση αυτής της ευπάθειας μπορεί να οδηγήσει σε πλήρη συμβιβασμό του ευάλωτου συστήματος. Ύστερα η Adobe κυκλοφόρησε μια ενημέρωση για το Adobe Character Animator για Windows και macOS. Αυτή η ενημέρωση επιλύει μια ευπάθεια υπερχείλισης buffer βάσει στοίβας που θα μπορούσε να οδηγήσει σε απομακρυσμένη εκτέλεση κώδικα.

*Πολλαπλές ευπάθειες στο Microsoft Media Foundation*

Το συγκεκριμένο περισταστικό διαδραματίστηκε στο ευάλωτο λογισμικό Windows και στον server του με υψηλή δριμύτητα και σοβαρότητα. Ως αναγνωριστικά CVE (κοινές ευπάθειες) υπήρχαν τα CVE-2020-1028,CVE-2020-1126,CVE-2020-1150,CVE-2020-1136. Αυτή η υπερχείλιση buffer βάσει στοίβας έγινε χωρις αξιοποίηση διαθεσιμότητας και η επιτυχής εκμετάλλευση αυτής της ευπάθειας μπορεί να οδηγήσει σε πλήρη συμβιβασμό του ευάλωτου συστήματος. Οι ευάλωτες εκδόσεις λογισμικού ήταν τα παράθυρα: 10 1607, 10 1709, 10 1803, 10 1809, 10 1903, 10 1909 και τα Windows servers: 2016, 2019, 2019 1803, 2019 1903, 2019 1909. Η ενημερωμένη έκδοση ασφαλείας αντιμετωπίζει την ευπάθεια διορθώνοντας τον τρόπο με τον οποίο το Windows Media Foundation χειρίζεται αντικείμενα στη μνήμη.

**Συνέπειες της αδυναμίας CWE-121 (Stack-based Buffer Overflow)**

Γενικά όπως έχουμε αναφέρει και παραπάνω, οι υπερχειλίσεις buffer oδηγούν σε διακοπές. Όταν οι συνέπειες της υπερχείλισης είναι αυθαίρετη εκτέλεση κώδικα εξ αποστάσεως, τότε αυτό χρησιμοποιείται για την καταστροφή οποιασδήποτε υπηρεσίας ασφάλειας.

Υπάρχουν πολλές και διαφορετικές συνέπειες που δημιουργούνται με αυτήν την αδυναμία. Ανάλογο το πεδίο της εφαρμογής (περιοχή ασφαλείας της εφαρμογής) λοιπόν της επίθεσης, υπάρχουν και διαφορετικές επιπτώσεις που έχουν φυσικά αρνητικό αποτέλεσμα, όταν ο εκάστοτε εισβολέας εκμεταλλευτεί αυτή την αδυναμία. Υπάρχει πάντως μεγάλη πιθανότητα εκμετάλλευσης μιας αδυναμίας για την επίτευξη ενός συγκεκριμένου αντίκτυπου, αλλά χαμηλή πιθανότητα να αξιοποιηθεί για να επιτευχθεί διαφορετικός αντίκτυπος.

Όταν η περιοχή της εφαρμογής που εμφανίζεται η αδυναμία είναι η διαθεσιμότητα, τότε το τεχνικό αποτέλεσμα που προκύπτει είναι ένα σφάλμα και υπάρχει η ανάγκη για επανεκκίνηση ή και έξοδο. Επίσης,καταναλώνονται οι πόροι στην μνήμη και στο CPU. Κάποιες επιθέσεις που οδηγούν σε έλλειψη διαθεσιμότητας είναι αρκετά ισχυρές, όπως η τοποθέτηση του προγράμματος σε έναν άπειρο βρόχο.

Όταν η περιοχή της εφαρμογής που εμφανίζεται η αδυναμία είναι ο έλεγχος πρόσβασης, τότε το τεχνικό αποτέλεσμα που προκύπτει είναι η εκτέλεση μη εξουσιοδοτημένου εντολών ή κώδικα. Επίσης, παρακάμπτωνται οι μηχανισμοί προστασίας. Κάποιες επιθέσεις που οδηγούν σε έλλειψη διαθεσιμότητας είναι αρκετά ισχυρές, όπως η τοποθέτηση του προγράμματος σε έναν άπειρο βρόχο. Έτσι υπάρχει συχνά η εκτέλεση αυθαίρετου κώδικα, ο οποίος συνήθως δεν εμπίπτει στο πεδίο εφαρμογής της σιωπηρής πολιτικής ασφάλειας ενός προγράμματος.

Όταν η περιοχή της εφαρμογής που εμφανίζεται η αδυναμία είναι οπουδήποτε αλλού, τότε το τεχνικό αποτέλεσμα που προκύπτει είναι το ίδιο με τα υπόλοιπα. Πάλι η συνέπεια είναι αυθαίρετη εκτέλεση κώδικα και συχνά χρησιμοποιείται για την ανατροπή οποιασδήποτε άλλης υπηρεσίας ασφαλείας.

Πάνω κάτω οι ίδιες συνέπεις εμφανίστηκαν και όταν διαδραματίστηκαν τα παραπάνω σχετιζόμενα περιστατικά της αδυναμίας CWE-121 σε ευάλωτα λογισμικά.

**ΠΙΘΑΝΑ AΝΤΙΜΕΤΡΑ ΤΗΣ ΑΔΥΝΑΜΙΑΣ**

Για να αντιμετωπιστεί αυτή η αδυναμία και οι ευπάθειες σε ένα σύστημα τότε υπάρχουν κάποια αντίμετρα-μετριασμοί για διάφορες προσεγγίσεις ώστε να επιτευχθεί η ασφάλεια ενός συστήματος ή υπηρεσίας και των δεδομένων του και να μην είναι δυνατή πλέον η εκτέλεση αυθαίρετου κώδικα, η αντικατάσταση δεικτών στοίβας-διευθύνσεων επιστροφής που διευκολύνει την πρόσβαση στο σύστημα από εισβολείς.

Τα αντίμετρα αυτά ενεργοποιούνται σε διάφορες φάσεις κάθε φορά. Προγραμματιστικά υπάρχει η δυνατότητα χρήσης ασφαλέστερων συναρτήσεων στον κώδικα όπως είναι οι strncat(), strncpy() κ.α. Επίσης μπορούν να χρησιμοποιηθούν ασφαλέστερες δυναμικές βιβλιοθήκες σύνδεσης για να ελεγχθεί καλύτερα το μήκος των δεδομένων πριν την αντιγραφή.

Στην φάση των απαιτήσεων λοιπόν, μπορεί να χρησιμοποιηθεί ένας μεταγλωττιστής ή μια γλώσσα που να εκτελούν αυτόματο έλεγχο ορίων των δεδομένων.

Στο σημείο της σχεδίασης και της αρχιτεκτονικής μπορεί να χρησιμοποιηθεί μια δυναμική βιβλιοθήκη αφαίρεσης. Αυτό το αντίμετρο δεν αποτελεί μια ολοκληρωμένη λύση για την επίθεση.

Στο επόμενο στάδιο της κατασκευής υπάρχουν μηχανισμοί εκτέλεσης και μεταγλώττισης που περιέχουν μηχανισμό προστασίας και εξαλείφουν τις υπερχειλίσεις buffer. Κάποιοι από αυτούς τους μηχανισμούς είναι το StackGuard, το ProPolice κ.α. Και αυτό απαιτεί αυτόματο έλεγχο ορίων για να αποτελεί ολοκληρωμένη λύση.

Στην φάση της υλοποίησης μια λύση είναι φύσικα η εφαρμογή ορίων ελέγχου στην είσοδο, η χρήση ατελείωτης στοίβας, καθώς και η μη χρησιμοποίηση επικίνδυνων λειτουργιών. Χρειάζονται ασφαλέστερες λειτουργίες για να γίνεται έλεγχος για σφάλματα ορίου.

Τέλος, όσον αφορά το λειτουργικό σύστημα μπορεί να εφαρμοστεί προληπτική λειτουργικότητα με το ASLR (Address Space Layout Randomization), με το οποίο κάθε φορά που φορτώνεται ο κώδικας στη μνήμη, η διεύθυνση της στοίβας αλλάζει και είναι δύσκολο να την μαντέψει κάποιος.



**ΒΙΒΛΙΟΓΡΑΦΙΑ**

* <https://cve.mitre.org/about/>
* <https://cwe.mitre.org/about/index.html>
* <https://en.wikipedia.org/wiki/Buffer_overflow>
* <https://cwe.mitre.org/data/definitions/121.html>
* <https://www.cybersecurity-help.cz/vdb/cwe/121/>
* <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-5618>
* <https://vuldb.com/?id.154576>
* <https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=cwe-121>
* <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-5619>
* <https://vuldb.com/?id.154577>
* <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-5621>
* <https://vuldb.com/?id.154579>
* <https://www.dcc.fc.up.pt/~edrdo/aulas/qses/lectures/qses-05-buffer-overflows.pdf>
* <https://www.cybersecurity-help.cz/vdb/SB2020061028>
* <https://www.cybersecurity-help.cz/vdb/SB2020052022>
* <https://www.cybersecurity-help.cz/vdb/SB2020052019>
* <https://www.huawei.com/en/psirt/security-advisories/huawei-sa-20200527-01-stack-en>
* <https://helpx.adobe.com/security/products/character_animator/apsb20-25.html>
* <https://www.cybersecurity-help.cz/vdb/SB2020051228>
* <https://fortiguard.com/encyclopedia/endpoint-vuln/62881>