**Περιγραφή των challenges**

**Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών**

**Μάθημα: Ασφάλεια και Προστασία Υπολογιστικών Συστημάτων**

**Καθηγητής: Αθανάσιος Αυγερινός**

**Εξάμηνο: 8ο**

**Sshkey**

* **Δημιουργία ssh key pair**

Δημιούργησα ένα ζεύγος δημόσιου και ιδιωτικού κλειδιού SSH στον τοπικό μου υπολογιστή με την εντολή «**ssh-keygen -t rsa -b 4096**», η οποία δημιουργεί ένα ζεύγος κλειδιών τύπου RSA με μήκος 4096 bits. Το **ιδιωτικό κλειδί** αποθηκεύτηκε στο **~\.ssh\hackintro\_key**, ενώ το **δημόσιο** στο **~\.ssh\hackintro\_key.pub**.

* **Σύνδεση μέσω SSH**

Στο αρχείο «**~/.ssh/authorized\_keys**», αποθήκευσα το δημόσιο κλειδί. Συνδέθηκα μέσω ssh, με «**ssh -i ~\.ssh\hackintro\_key** [**athanasiospapoulias1979@shell.hackintro25.di.uoa.gr»**](mailto:athanasiospapoulias1979@shell.hackintro25.di.uoa.gr), όπου μου εμφανίστηκε το σωστό flag: **secure\_your\_shells\_with\_crypto**.

* **Ιδιαιτερότητα του challenge**

Η συγκεκριμένη άσκηση δεν απαιτεί username/password για σύνδεση, αλλά μας φέρνει σε επαφή με πιο ασφαλείς μεθόδους ταυτοποίησης, όπως το SSH. Επιπλέον, βλέπουμε στην πράξη πως υλοποιείται η κρυπτογραφία δημοσίου κλειδιού (ασύμμετρη κρυπτογράφηση), κατά την οποία μπορούμε να μοιραζόμαστε το δημόσιο κλειδί, ενώ κρατάμε μυστικό το ιδιωτικό και η σύνδεση, μας επιτρέπεται μόνο όταν διαθέτουμε το σωστό ιδιωτικό κλειδί.

**Hipstore**

* Το πρόγραμμα Hipstore δημιουργεί, αγοράζει και πουλάει Pokémon.
* Δοκίμασα να δημιουργήσω Pokémon με διαφορετικές τιμές, οι οποίες αποθηκεύονται στη μνήμη. Παρατήρησα ότι όταν εισήγαγα ως τιμή έναν μεγάλο αριθμό (100000000), το πρόγραμμα ναι μεν αποθηκεύει την τιμή στη μνήμη, αλλά δεν τη διαχειρίζεται σωστά.
* Ενώ προσπάθησα να πουλήσω Pokémon με πολύ μεγάλη τιμή, το πρόγραμμα δεν ανταποκρίνεται σωστά. Συγκεκριμένα, το πρόγραμμα δεν το αναγνώρισε, πιθανόν λόγω της μεγάλης τιμής που έχει.
* Επομένως, σε αυτό το σημείο, κατάλαβα ότι κατά τη δημιουργία Pokémon με υπερβολικά μεγάλες τιμές, το πρόγραμμα δεν χειρίζεται σωστά τη μνήμη και μπορεί να «διαρρεύσει» δεδομένα λόγω αυτής της υπερχείλισης. Αυτό μπορεί να οδηγήσει σε **heap overflow**, κάτι που μπορεί να μας επιτρέψει να περάσουμε μέσω αυτής της μνήμης σε περιοχές της μνήμης που είναι πιο δύσκολο να προσπελαστούν κι έτσι να καταφέρουμε να αποκτήσουμε πρόσβαση στο flag.