ΗΥ-457: Εισαγωγή στα Συστήματα Ασφάλειας Πληροφοριών

Τμήμα Επιστήμης Υπολογιστών

Εαρινό Εξάμηνο 2024

Άσκηση 2

“Υλοποίηση Πλατφόρμας Προστασίας Απέναντι σε Ransomware”

Φροντιστήριο: 21/03/2024
Προθεσμία: 21/04/2024

Εισαγωγή

Τις τελευταίες εβδομάδες, έχουν προκύψει πολλές αναφορές σχετικά με το έργο της KozaliBear, μιας διαβόητης ομάδας κυβερνοεγκληματιών που είναι γνωστοί για την εισβολή σε εταιρικά δίκτυα και περιβάλλοντα cloud οργανισμών. Αυτή η ομάδα είναι γνωστό ότι χρησιμοποιεί ιούς για να μολύνει τους σταθμούς εργασίας ανυποψίαστων υπαλλήλων, να αποκτά μη-εξουσιοδοτημένη πρόσβαση σε απόρρητα δεδομένα και ακόμη και να κλέβει απόρρητες πληροφορίες. Αυτό που κάνει αυτή την ομάδα εξαιρετικά επικίνδυνη είναι η χρήση ransomware. Τα ransomware είναι ένας τύπος κακόβουλου λογισμικού που κλειδώνει τα προσωπικά αρχεία του χρήστη έως ότου πληρωθούν τα λύτρα [1]. Τα αρχεία είναι κρυπτογραφημένα και ο μόνος τρόπος για να λάβει το θύμα το κλειδί αποκρυπτογράφησης είναι εάν πληρώσει ένα συγκεκριμένο χρηματικό ποσό. Η ομάδα KozaliBear έχει καταφέρει να εκβιάσει εκατομμύρια χρησιμοποιώντας ransomware. Τέλος, συνήθως ζητούν χρήματα με τη μορφή κρυπτονομισμάτων (π.χ. Bitcoin) [2] αφού δεν είναι εύκολο (αλλά όχι αδύνατο) να εντοπιστούν.

Είστε μηχανικός κυβερνοασφάλειας που εργάζεται για έναν κορυφαίο κατασκευαστή ηλεκτρονικών ειδών αυτοκινήτου. Χθες το βράδυ, ο προϊστάμενός σας ενημερώθηκε ότι ένας από τους ανταγωνιστές σας δέχθηκε επίθεση από το KozaliBear και ένα ransomware εγκαταστάθηκε σε όλα τα συστήματά τους. Ο προϊστάμενός σας ανησυχεί ότι μπορεί να είστε ο επόμενος στόχος ή ότι ορισμένοι από τους σταθμούς εργασίας σας μπορεί να έχουν ήδη μολυνθεί. Οι ανταγωνιστές σας μοιράστηκαν ευγενικά μαζί σας, Indicators of Compromise (IoC) [3]. Αυτά είναι ψηφιακά στοιχεία που υποδηλώνουν ότι ένα σύστημα μπορεί να έχει παραβιαστεί. Σας έχει ανατεθεί να βελτιώσετε την υποδομή ασφαλείας της εταιρείας σας. Απαιτείται να διασφαλίσετε ότι τα συστήματά σας δεν έχουν παραβιαστεί και να δημιουργήσετε ένα ασφαλές περιβάλλον, όπου μπορούν να αποθηκευτούν με ασφάλεια εμπιστευτικά αρχεία. Τέλος, θα θέλατε να υποστηρίξετε άλλους μηχανικούς κυβερνοασφάλειας και να τους βοηθήσετε να εντοπίσουν τη συγκεκριμένη επίθεση.

Indicators of Compromise

Οι ειδικοί ασφαλείας που μελέτησαν αυτήν την επίθεση έχουν εντοπίσει τα ακόλουθα χαρακτηριστικά:

- Ο ιός χρησιμοποιεί μια κακόβουλη βιβλιοθήκη για να κρυπτογραφηθεί και να παραμείνει κρυφός. Οι εισβολείς θεώρησαν ότι θα ήταν καλή ιδέα να υλοποιήσουν τον κώδικα κρυπτογράφησης ως κοινόχρηστη βιβλιοθήκη προκειμένου να τον χρησιμοποιήσουν σε άλλες εφαρμογές κακόβουλου λογισμικού τους. Οι ειδικοί έχουν εντοπίσει δύο διαφορετικές εκδόσεις αυτής της βιβλιοθήκης. Η πρώτη έκδοση έχει κατακερματισμό MD5 [4] `85578cd4404c6d586cd0ae1b36c98aca` ενώ η δεύτερη έκδοση έχει SHA256 [5] `d56d67f2c43411d966525b3250bfaa1a85db34bf371468df1b6a9882fee78849`

- Το ransomware που ανέπτυξε το KozaliBear ζητά Bitcoin για να ξεκλειδώσει τα αρχεία του θύματος. Οι εισβολείς χρησιμοποιούν το Bitcoin επειδή πιστεύουν ότι είναι ανώνυμο. Τα θύματα θα πρέπει να πληρώσουν τα χρήματα στο ακόλουθο πορτοφόλι: bc1qa5wkgaew2dkv56kfvj49j0av5nml45x9ek9hz6

- Ειδικοί ασφαλείας ερεύνησαν διεξοδικά την επίθεση και ανακάλυψαν ότι οι εισβολείς χρησιμοποιούν έναν παλιό ιό που όταν συνδέεται με άλλα προγράμματα περιέχει την υπογραφή 98 1d 00 00 ec 33 ff ff fb 06 00 00 00 46 0e 10

- Το ransomware που χρησιμοποιούν οι εισβολείς κρυπτογραφεί τοπικά αρχεία χρησιμοποιώντας τη διαδικασία:
    a. Διαβάζει πρώτα ολόκληρο το περιεχόμενο ενός αρχείου: example.txt
    b. Στη συνέχεια δημιουργεί ένα νέο αρχείο με βάση το αρχικό όνομα αρχείου και προσαρτά το .locked στο νέο όνομα του αρχείου: example.txt.locked
    c. Γράφει το κρυπτογραφημένο περιεχόμενο στο νέο αρχείο.
    d. Διαγράφει το αρχικό αρχείο.

- Άλλοι μηχανικοί κυβερνοασφάλειας παρατήρησαν αυξημένη κίνηση δικτύου όταν τα συστήματά τους μολύνθηκαν από το KozaliBear. Χρησιμοποιώντας reverse engineering πάνω σε αυτή τη κίνηση από διάφορους μολυσμένους σταθμούς εργασίας, διαπίστωσαν ότι το ransomware προσπαθεί να κατεβάσει κακόβουλο λογισμικό από δημοφιλείς διαδικτυακές πλατφόρμες διανομής κακόβουλου λογισμικού για να μολύνει περαιτέρω τον σταθμό εργασίας του θύματος.

1. Σάρωση για μολυσμένα αρχεία

Η πρώτη σας εργασία είναι να δημιουργήσετε μια εφαρμογή που σαρώνει έναν σταθμό εργασίας και αναζητά μολυσμένα αρχεία. Για να ανακαλύψετε μολυσμένα αρχεία θα χρησιμοποιήσετε τα Indicators of Compromise της προηγούμενης ενότητας. Η εφαρμογή θα λειτουργεί μόνο ως σύστημα ειδοποιήσεων. Δεν θα διαγράψει τυχόν μολυσμένα αρχεία που ανακαλύφθηκαν. Θα ειδοποιήσει μόνο τους διαχειριστές ότι υπάρχουν μολυσμένα αρχεία. Για αυτήν την εργασία πρέπει να χρησιμοποιήσετε τη γλώσσα C. Η εφαρμογή σας θα λάβει ως είσοδο έναν φάκελο τον οποίο στη συνέχεια θα σαρώσει για μολυσμένα αρχεία.

Στοιχεία Υλοποίησης

Για αυτήν την εργασία, επιτρέπεται να χρησιμοποιείτε μόνο την standard βιβλιοθήκη της C, καθώς και τη βιβλιοθήκη OpenSSL για τον υπολογισμό των τιμών κατακερματισμού των αρχείων. Περισσότερες πληροφορίες για τη βιβλιοθήκη OpenSSL μπορείτε να βρείτε στα man pages [6]. Είναι σημαντικό η υλοποίησή σας να εκτελεί μια αναδρομική διαδρομή ξεκινώντας από τον δεδομένο κατάλογο και να αναζητά μολυσμένα αρχεία σε όλους τους υποκαταλόγους του δέντρου αρχείων. Για παράδειγμα, εάν ο χρήστης θέλει να σαρώσει το /secret/data, τότε το σύστημά σας θα πρέπει να βρει όλα τα μολυσμένα αρχεία στον συγκεκριμένο κατάλογο (π.χ. /secret/data/malicious.out) καθώς και σε οποιουσδήποτε υποκαταλόγους (π.χ. /secret/data/super /duper/secret/evil.dat). Για κάθε αρχείο που ανακαλύπτετε στο δέντρο καταλόγου, πρέπει να υπολογίσετε τις τιμές κατακερματισμού MD5 και SHA256 και να ελέγξετε αν ταιριάζουν με κάποιον από τους κατακερματισμούς των γνωστών κακόβουλων βιβλιοθηκών. Επιπλέον, πρέπει να διαβάσετε το περιεχόμενο κάθε αρχείου και να αναζητήσετε είτε την υπογραφή του γνωστού ιού είτε την αναφερόμενη διεύθυνση Bitcoin. Οποιοδήποτε αρχείο βρεθεί ότι περιέχει αυτές τις δύο τιμές μπορεί να θεωρηθεί μολυσμένο. Όταν έχετε επεξεργαστεί με επιτυχία όλα τα αρχεία, η εφαρμογή σας θα πρέπει να παρέχει μια μικρή αναφορά με πληροφορίες σχετικά με τα αρχεία που επεξεργάστηκε, καθώς και τυχόν μολυσμένα αρχεία που ανακαλύφθηκαν.

```
$ antivirus scan /home/ceo/Downloads

[INFO] [9046] [14-Mar-24 13:53:43] Application Started
[INFO] [9046] [14-Mar-24 13:53:43] Scanning directory /home/ceo/Downloads
[INFO] [9046] [14-Mar-24 13:53:45] Found 3125 files
[INFO] [9046] [14-Mar-24 13:53:45] Searching…
[INFO] [9046] [14-Mar-24 13:53:55] Operation finished
[INFO] [9046] [14-Mar-24 13:53:55] Processed 3125 files. Found 7 infected

/home/ceo/Downloads/boot/grub/x86_64-efi/reiserfs.mod:REPORTED_VIRUS
/home/ceo/Downloads/malicious.so:REPORTED_MD5_HASH
/home/ceo/Downloads/snap/bare/current/dev/display.py:REPORTED_BITCOIN
/home/ceo/Downloads/snap/bare/current/dev/get.py:REPORTED_BITCOIN
/home/ceo/Downloads/snap/bare/current/dev/main.py:REPORTED_BITCOIN
/home/ceo/Downloads/kozi/v2/mal.so:REPORTED_SHA256_HASH
/home/ceo/Downloads/opt/google/chrome/libEGL.so:REPORTED_VIRUS
```

2. Ανίχνευση πιθανής επιβλαβούς κίνησης δικτύου

Η δεύτερη εργασία σας είναι να υλοποιήσετε μια εφαρμογή που σαρώνει όλα τα αρχεία μέσα σε ένα φάκελο και προσπαθεί να ανακαλύψει εάν αυτά τα αρχεία ενδέχεται να δημιουργήσουν επιβλαβή κίνηση δικτύου. Το σύστημά σας θα λειτουργεί σε επίπεδο domain και θα ειδοποιεί τους διαχειριστές ότι ένα αρχείο είναι δυνητικά επιβλαβές αν επιχειρήσει να αλληλεπιδράσει με έναν κακόβουλο domain. Επιπλέον, το σύστημά σας θα είναι προληπτικό. Θα προσπαθήσει να εντοπίσει τέτοια αρχεία πριν εκτελεστούν και προτού δημιουργηθεί οποιαδήποτε κίνηση δικτύου. Και πάλι, δεν πρέπει να διαγράψετε ή να βάλετε σε καραντίνα τυχόν αρχεία που ανακαλύφθηκαν. Επιτρέπεται να χρησιμοποιείτε μόνο τη γλώσσα προγραμματισμού C. Η εφαρμογή σας θα λάβει ως είσοδο έναν κατάλογο τον οποίο στη συνέχεια θα σαρώσει.

Στοιχεία Υλοποίησης

Η εφαρμογή σας θα πρέπει να βρει όλα τα αρχεία σε ένα δέντρο καταλόγου, να εξαγάγει όλα τα domains που βρίσκονται σε κείμενο μέσα σε αυτά τα αρχεία και να ελέγξει εάν είναι κακόβουλα ή όχι. Όπως και πριν, θα πρέπει να εκτελέσετε μια αναδρομική διαδρομή ξεκινώντας από τον δεδομένο φάκελο και να αναζητήσετε αρχεία στους υποφακέλους. Για να εξαγάγετε domains από το περιεχόμενο ενός αρχείου θα πρέπει να χρησιμοποιήσετε regular expressions. Είστε ελεύθεροι να χρησιμοποιήσετε οποιαδήποτε regular expression θεωρείτε κατάλληλη, αλλά θα πρέπει να είστε σε θέση να την εξηγήσετε.

Για να ταξινομήσετε τομείς σε κακόβουλους ή καλοήθεις, θα πρέπει να χρησιμοποιήσετε το φίλτρο περιεχομένου της Cloudflare [7]. Για κάθε domain που ανακαλύπτετε, θα στέλνετε requests στα endpoints του Cloudflare για να ανακαλύψετε εάν το domain είναι κακόβουλο ή όχι. Για να το πετύχετε αυτό, πρέπει να χρησιμοποιήσετε την βιβλιοθήκη libcurl. Μπορείτε να βρείτε περισσότερες πληροφορίες σχετικά με τη βιβλιοθήκη URL στη σελίδα της [8].

Όταν έχετε επεξεργαστεί με επιτυχία όλα τα αρχεία, η εφαρμογή σας θα πρέπει να παρέχει μια μικρή αναφορά με πληροφορίες σχετικά με τα domains που εντόπισε, τα αρχεία που τα περιείχαν, καθώς και μια απόφαση εάν το domain είναι κακόβουλο ή όχι. Η εφαρμογή σας θα πρέπει να επεξεργάζεται όλα τα αρχεία που ανακαλύπτει, ανεξάρτητα από το αν είναι δυαδικά ή αρχεία κειμένου ή εάν είναι εκτελέσιμα ή όχι. Ωστόσο, στην τελική αναφορά σας μπορείτε προαιρετικά να καθορίσετε ποια αρχεία ήταν εκτελέσιμα.

```
$ antivirus inspect /home/ceo/

[INFO] [9046] [14-Mar-24 13:53:43] Application Started
[INFO] [9046] [14-Mar-24 13:53:43] Scanning directory /home/ceo/
[INFO] [9046] [14-Mar-24 13:53:45] Found 18312 files
[INFO] [9046] [14-Mar-24 13:53:45] Searching…
[INFO] [9046] [14-Mar-24 13:53:55] Operation finished
[INFO] [9046] [14-Mar-24 13:53:55] Processed 18312 files.

| FILE    | PATH                  | DOMAIN        | EXECUTABLE | RESULT  |
=============================================================================
| foo.exe | /home/ceo/docs/secret | www.google.com    | True   | Safe    |
| bar.txt | /home/ceo/hy457grade/ | alphaxiom.com     | False  | Malware |
| libd.so | /home/ceo/Desktop/    | https://bbc.com   | False  | Safe    |
| wget.sh | /home/ceo/aws/plugin  | biawwer.com       | True   | Malware |
```

3. Διασφάλιση πολύτιμων αρχείων

Το τρίτο καθήκον σας είναι να δημιουργήσετε έναν ασφαλή θύλακα όπου οι χρήστες μπορούν να τοποθετήσουν τα πιο πολύτιμα δεδομένα τους. Αυτό θα υλοποιηθεί με τη μορφή ενός ασφαλούς φακέλους ο οποίος θα παρακολουθείται συνεχώς και θα προστατεύεται από ransomware. Για άλλη μια φορά, επιτρέπεται να χρησιμοποιήσετε μόνο τη γλώσσα προγραμματισμού C. Η εφαρμογή σας θα λάβει ως είσοδο έναν κατάλογο τον οποίο θα παρακολουθεί μέχρι να τερματίσει.

Στοιχεία Υλοποίησης

Η εφαρμογή σας θα δημιουργήσει έναν ασφαλή θύλακα παρακολουθώντας τα συμβάντα του συστήματος αρχείων στον φάκελο που καθόρισε ο χρήστης. Θα πρέπει να παρακολουθείτε και να εκτυπώνετε όλα αυτά τα συμβάντα σε πραγματικό χρόνο. Κάθε φορά που λαμβάνει χώρα ένα νέο συμβάν συστήματος αρχείων, θα πρέπει να αξιολογείτε εάν αυτό το συμβάν (μαζί με προηγούμενα συμβάντα) υποδεικνύει την παρουσία ενός ransomware που προσπαθεί να επιτεθεί στον ασφαλή θύλακα. Όταν εντοπίσετε μια τέτοια συμπεριφορά, απαιτείται μόνο να εκτυπώσετε ένα μήνυμα ως ειδοποίηση προς τους διαχειριστές. Δεν χρειάζεται να πραγματοποιηθεί καμία άλλη ενέργεια.

Για την παρακολούθηση συμβάντων συστήματος αρχείων, πρέπει να χρησιμοποιήσετε το inotify API [9] και επικεντρωθείτε στη λειτουργικότητα των καταλόγων. Δεν πρέπει να παρακολουθείτε συγκεκριμένα αρχεία, καθώς δεν γνωρίζουμε εκ των προτέρων ποια αρχεία θα τοποθετήσουν οι διευθυντές στον φάκελο. Σε αυτήν την εργασία, δεν απαιτείται να χειρίζεστε υποκαταλόγους. Μπορείτε να υποθέσετε ότι όλα τα αρχεία τοποθετούνται απευθείας στον κύριο κατάλογο που ο χρήστης καθόρισε κατά την εκκίνηση της εφαρμογής.

```
$ antivirus monitor /root/vault/

[INFO] [9046] [14-Mar-24 13:53:43] Application Started
[INFO] [9046] [14-Mar-24 13:53:43] Monitoring directory /root/vault/
[INFO] [9046] [14-Mar-24 13:53:43] Waiting for events...
File 'info.txt' was created
File 'info.txt' was opened
File 'info.txt' that was not opened for writing was closed
File 'passwords.txt' was opened
File 'passwords.txt' was accessed
File '.tmpSjxiska.dat' was deleted from watched directory
File 'passwords.txt.locked' was created
File 'passwords.txt.locked' was modified
File 'passwords.txt.locked' that was opened for writing was closed
File 'passwords.txt' was deleted from watched directory
[WARN] Ransomware attack detected on file passwords.txt
File '.tmpSIfwiunew.dat' was created
File 'studentGrades.csv' was opened
```

4. Προστασία από μη εξουσιοδοτημένη πρόσβαση

Το τελευταίο σας καθήκον είναι να διασφαλίσετε ότι όλα τα έγγραφα που έχουν τοποθετηθεί στον ασφαλή θύλακα του προηγούμενου βήματος, δεν θα πέσουν σε λάθος χέρια. Υπάρχει πιθανότητα ένας από τους σταθμούς εργασίας να έχει μολυνθεί ή κάποιος από τους ενδιαφερόμενους να γίνει κακόβουλος και να προσπαθήσει να κλέψει ένα από αυτά τα αρχεία. Οι διευθυντές θέλουν να διασφαλίσουν ότι όλα τα αρχεία στον κατάλογο είναι κρυπτογραφημένα και ότι κανένα άτομο δεν μπορεί να έχει πρόσβαση στα αρχεία μόνο του. Ο προϊστάμενός σας σας ανέθεσε να εφαρμόσετε μια λύση χρησιμοποιώντας το σύστημα του Shamir [10]. Το κομμάτι της κρυπτογράφησης θα υλοποιηθεί από διαφορετικό συνάδελφο. Δεν χρειάζεται να ανησυχείτε για αυτό.

Στοιχεία Υλοποίησης

Η κοινή χρήση μυστικού λειτουργεί με το διαχωρισμό ενός μυστικού σε μικρότερα κομμάτια και στη συνέχεια τη διανομή αυτών των κομματιών μεταξύ μιας ομάδας ή ενός δικτύου. Κάθε μεμονωμένο μερίδιο είναι άχρηστο από μόνο του, αλλά όταν όλες τα κομμάτια είναι μαζί, ανασυνθέτουν το αρχικό μυστικό. Το αρχικό μυστικό στην περίπτωσή μας είναι το κλειδί που αποκρυπτογραφεί τα αρχεία. Το να απαιτούμε όλα τα μέρη να ανακατασκευάζουν το αρχικό μυστικό κάθε φορά που θέλουμε να αποκτήσουμε πρόσβαση σε ένα αρχείο, φαίνεται μη πρακτικό και αναποτελεσματικό. Αντίθετα, πρέπει να καθοριστεί ένα όριο ελάχιστων μετοχών για να αποφευχθεί η απρόβλεπτη συμπεριφορά των διευθυντών.

Για αυτήν την εργασία απαιτείται να εφαρμόσετε έναν μυστικό μηχανισμό κοινής χρήσης που θα αποκρυπτογραφεί τα αρχεία όταν είναι παρόντα τουλάχιστον τρία μέλη του διοικητικού συμβουλίου της εταιρίας. Υπάρχουν δέκα μέλη στο συμβούλιο των ενδιαφερομένων. Μόνο εάν υπάρχουν τρία (ή περισσότερα) από αυτά, είναι δυνατή η πρόσβαση σε ένα αρχείο. Διαφορετικά, ο ασφαλής θύλακας παραμένει σφραγισμένος. Για να μοιραστείτε τον κωδικό πρόσβασης μεταξύ των μελών του συμβουλίου, πρέπει να εφαρμόσετε μια μέθοδο κοινής χρήσης μυστικού που βασίζεται σε πολυώνυμα. Πιο συγκεκριμένα θα γράψετε ένα πρόγραμμα C που:

1. Κατασκευάζει ένα πολυώνυμο 2ου βαθμού f(x) = a2·x2 + a1·x + a0 όπου ου a0 είναι ο κωδικός πρόσβασης και a1 , a2 είναι αριθμοί που δημιουργούνται τυχαία. Σημειώστε ότι εάν το μυστικό πρέπει να ανακατασκευαστεί από k οντότητες, ο βαθμός πολυωνύμου πρέπει να είναι k-1.

2. Δίνει σε κάθε μέλος του συμβουλίου ένα ζεύγος (x_n , $f(x_n)$). Το πρώτο μέλος του συμβουλίου θα έπαιρνε το f(1) το δεύτερο το f(2), το τρίτο το f(3) και το τελευταίο θα έπαιρνε το f(10). Σημειώστε ότι η f(0) έχει ως αποτέλεσμα $f(x) = a_2 \cdot 0 + a \cdot 0 + a_0 \Leftrightarrow f(0) = a_0$. Ως εκ τούτου, το f(0) είναι ο μυστικός κωδικός πρόσβασης που χωρίζεται σε κομμάτια και δεν πρέπει να κοινοποιείται.

3. Είναι σε θέση να ανακατασκευάσει το αρχικό κλειδί κρυπτογράφησης εάν παρέχονται τουλάχιστον 3 μέρη του μυστικού ως είσοδο. Σημειώστε ότι αυτά τα 3 μέρη μπορεί να είναι οποιαδήποτε από τα αρχικά 10 και όχι απαραίτητα διαδοχικά μέλη του συμβουλίου.

4. Παρέχει δύο τρόπους λειτουργίας. Ένα τρόπο που χωρίζει το κλειδί κρυπτογράφησης και δημιουργεί τα 10 μέρη και ένα που κατασκευάζει το κλειδί με τουλάχιστον 3 μέρη.

```
$ antivirus slice 156

[INFO] [9046] [14-Mar-24 13:53:43] Application Started
[INFO] [9046] [14-Mar-24 13:53:43] Generating shares for key ‘156’

(1, 313)
(2, 760)
(3, 1497)
(4, 2524)
(5, 3841)
(6, 5448)
(7, 7345)
(8, 9532)
(9, 12009)
(10, 14776)
```

```
$ antivirus unlock (1, 313) (4, 2524) (9, 12009)

[INFO] [9046] [14-Mar-24 13:53:43] Application Started
[INFO] [9046] [14-Mar-24 13:53:43] Received 3 different shares
[INFO] [9046] [14-Mar-24 13:53:44] Computed that a=145 and b=12
[INFO] [9046] [14-Mar-24 13:53:44] Encryption key is: 156
```

5. Διάδοση Ευρημάτων (BONUS + 1)

Είναι σημαντικό να δημοσιοποιήσετε τα ευρήματά σας για να βοηθήσετε άλλους μηχανικούς κυβερνοασφάλειας και να μειώσετε τη δυσκολία για άλλους ερευνητές να κατανοήσουν τη συγκεκριμένη επίθεση. Αυτό θα διευκολύνει τον εντοπισμό και τη διακοπή αυτής της επίθεσης όσο το δυνατόν νωρίτερα, ακόμη και τη δημιουργία άμυνων εναντίον της. Ένας αποτελεσματικός τρόπος εντοπισμού και ταξινόμησης κακόβουλου λογισμικού είναι η χρήση των κανόνων YARA [11]. Ένας κανόνας YARA είναι μια περιγραφή ενός κακόβουλου λογισμικού με βάση τα μοτίβα του. Γράψτε έναν κανόνα YARA για να περιγράψετε την επίθεση KozaliBear που περιγράφεται στην Εισαγωγή. Επιπλέον, χρησιμοποιήστε το Arya εργαλείο [12] για τη δημιουργία ψευδο-κακόβουλων αρχείων που ταιριάζουν με τον κανόνα YARA σας για να δοκιμάσετε την υλοποίηση της εργασίας σας. Μπορείτε να υποβάλετε τον κανόνα YARA μαζί με την εντολή που χρησιμοποιήσατε για το εργαλείο Arya στο αρχείο README της εργασίας σας.

Σημειώσεις

1. Η εφαρμογή σας θα πρέπει να παράγει ένα μόνο εκτελέσιμο αρχείο που θα παρέχει διάφορες λειτουργίες. Κάθε μέρος αυτής της εργασίας θα πρέπει να υλοποιηθεί ως ενότητα της ίδιας εφαρμογής. Ο χρήστης της εφαρμογής σας θα καθορίσει ποια ενότητα θα εκτελεστεί χρησιμοποιώντας το κατάλληλο ρήμα (π.χ. Scan, secure, unlock) ως όρισμα CLI.

2. Αυτή δεν είναι ομαδική εργασία. Κάθε φοιτητής πρέπει να υποβάλει τη δική του υλοποίηση και δεν επιτρέπεται να συνεργαστείτε μεταξύ σας. Εάν αποφασίσετε να χρησιμοποιήσετε υπηρεσίες φιλοξενίας ή συστήματα ελέγχου εκδόσεων (π.χ. Git), μην ξεχάσετε να επισημάνετε το repository σας ως ιδιωτικό.

3. Εκτελέστε όλα τα ερωτήματα αυτής της άσκησης χρησιμοποιώντας τη γλώσσα προγραμματισμού C. Είστε ελεύθεροι να αναπτύξετε και να δοκιμάσετε την εφαρμογή σας στην προσωπική σας συσκευή, ωστόσο, η τελική έκδοση θα πρέπει να λειτουργεί σε σταθμούς εργασίας του CSD.

4. Έχει δημιουργηθεί ένας κατάλογος με διάφορα αρχεία για να δοκιμάσετε την εφαρμογή σας. Μπορείτε να κατεβάσετε τα αρχεία δοκιμής από την ιστοσελίδα του μαθήματος. Λάβετε υπόψη ότι αυτές οι δοκιμές είναι απλώς ενδεικτικές και ότι θα πρέπει να δημιουργήσετε τα δικά σας αρχεία δοκιμών για να βεβαιωθείτε ότι η εφαρμογή σας είναι σωστή.

5. Μπορείτε να χρησιμοποιήσετε τη λίστα αλληλογραφίας του μαθήματος για τυχόν ερωτήσεις που σχετίζονται με αυτήν την εργασία. Σας παρακαλούμε να παρέχετε ένα σαφές θέμα κατά την αποστολή ενός email.

6. Μην στέλνετε προσωπικά μηνύματα στους βοηθούς του μαθήματος. Άλλοι φοιτητές μπορεί να έχουν την ίδια ερώτηση.

7. 7. Μην στέλνετε αποσπάσματα κώδικα ή αρχεία της εφαρμογής σας στη λίστα αλληλογραφίας. Εάν το κάνετε, η εργασία σας δεν θα γίνει δεκτή και δεν θα βαθμολογηθείτε.

8. Για αυτήν την εργασία πρέπει να παρέχετε ένα Makefile. Το makefile πρέπει να περιέχει τουλάχιστον τρεις κανόνες. Έναν που δημιουργεί το πρόγραμμά σας, έναν που το εκτελεί χρησιμοποιώντας τα δικά σας δοκιμαστικά αρχεία και έναν που διαγράφει αρχεία κατασκευής (π.χ. αρχεία αντικειμένων). Καθαρίστε τους καταλόγους πριν υποβάλετε την εργασία σας.

9. Θα πρέπει να παρέχετε τα δικά σας δοκιμαστικά αρχεία για να αποδείξετε ότι η εφαρμογή σας είναι σωστή. Αυτό ισχύει για όλες τις εργασίες και μπορεί να περιλαμβάνει τη δημιουργία απλών εκτελέσιμων αρχείων.

10. Θα πρέπει να παρέχετε ένα σύντομο αρχείο README που να περιγράφει ποια μέρη της εργασίας έχετε υλοποιήσει, τι εφαρμόσατε διαφορετικά και οτιδήποτε άλλο θεωρείτε σημαντικό. Κρατήστε αυτό το αρχείο σχετικά σύντομο.

11. Ακολουθήστε τα βήματα που περιγράφονται παραπάνω και εφαρμόστε την εργασία σταδιακά. Αυτό θα είναι ιδιαίτερα χρήσιμο για εσάς. Μπορείτε να αναπτύξετε μικρά δομικά στοιχεία και στη συνέχεια να τα ενσωματώσετε στην τελική εφαρμογή. Αυτό θα σας βοηθήσει επίσης με τον εντοπισμό και την επίλυση σφαλμάτων.

12. Ο κωδικός που υποβάλατε θα ελεγχθεί για λογοκλοπή χρησιμοποιώντας κατάλληλο λογισμικό.

13. Μπορείτε να υποβάλετε την ανάθεση εκτελώντας την εντολή turnin assignment_2@hy457 directory_name όπου directory_name είναι ο κατάλογος που περιέχει τον πηγαίο κώδικα.

References

[1] https://en.wikipedia.org/wiki/Ransomware
[2] https://en.wikipedia.org/wiki/Cryptocurrency
[3] https://en.wikipedia.org/wiki/Indicator_of_compromise
[4] https://en.wikipedia.org/wiki/MD5
[5] https://en.wikipedia.org/wiki/SHA-2
[6] https://www.openssl.org/docs/manmaster/
[7] https://blog.cloudflare.com/introducing-1-1-1-1-for-families
[8] https://curl.se/libcurl/
[9] https://man7.org/linux/man-pages/man7/inotify.7.html
[10] https://en.wikipedia.org/wiki/Shamir%27s_secret_sharing
[11] https://yara.readthedocs.io/en/stable/index.html
[12] https://githu b.com/claroty/arya
[13] https://csd.uoc.gr/~hy457/resources/assignments/hy457_assignment_2_test_filesystem.zip