Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Δ.Π.Μ.Σ. Επιστήμης Δεδομένων και Μηχανικής Μάθησης

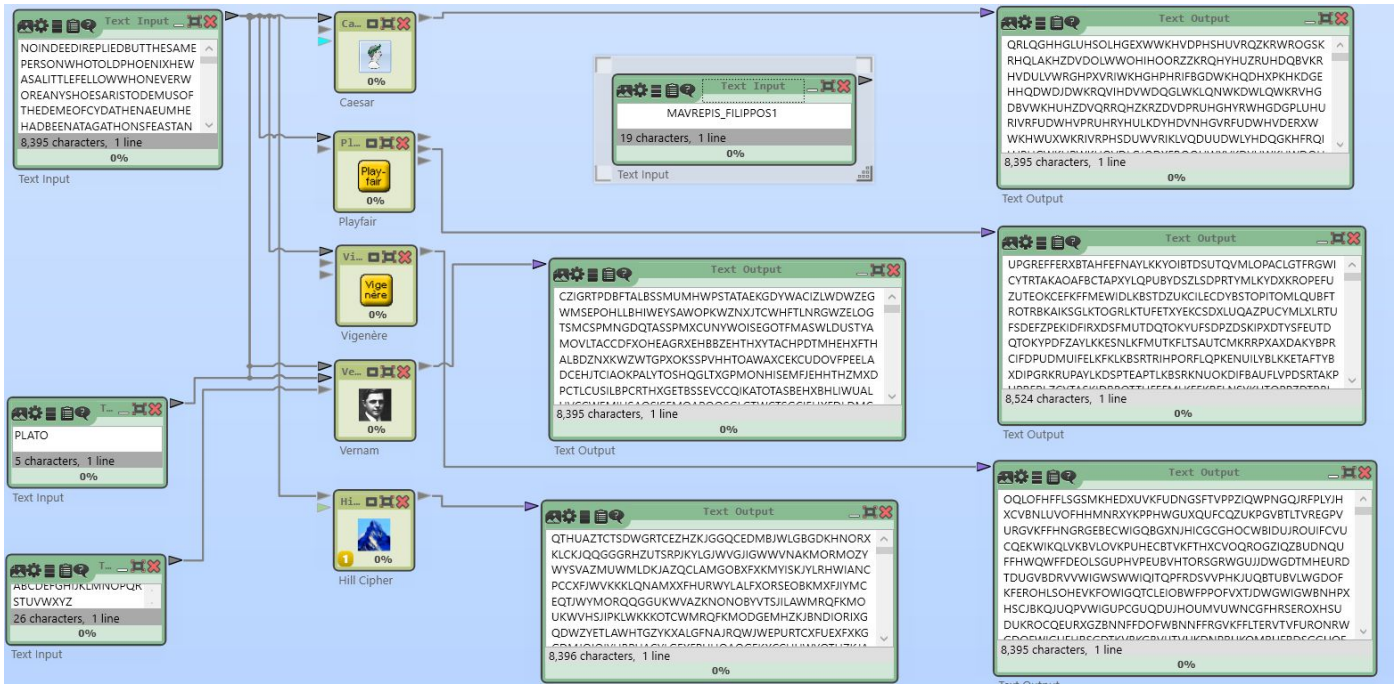
Ασφάλεια Δεδομένων και Ιδιωτικότητα

Εργαστηριακή Εργασία

[redacted]

Σαν κείμενο για το κομμάτι της ενασχόλησης με το Cryptool θα χρησιμοποιηθεί μέρος του έργου της "Πολιτείας" του Πλάτωνα.

Στην παρακάτω εικόνα φαίνονται τα ζητούμενα του ερωτήματος 1. Λόγω του μεγάλου μεγέθους της ενδείκνυται η χρήση zoom για την καλύτερη κατανόηση των αποτελεσμάτων.



Αρχικά προκειμένου να λειτουργήσουν όλοι οι αλγόριθμοι σωστά κάνουμε προεπεξεργασία στο κείμενο. Συγκεκριμένα αφαιρούνται τα κενά και τα σημεία στίξης.

Οι αλγόριθμοι που φαίνονται είναι οι Caesar, Playfair, Vigenere, Vernam, Hill

Caesar

Η μέθοδος πήρε το όνομά της από τον Ιούλιο Καίσαρα, ο οποίος την χρησιμοποιούσε στην προσωπική του αλληλογραφία. Ο Κώδικας του Καίσαρα είναι μία από τις απλούστερες και πιο γνωστές τεχνικές κωδικοποίησης στην κρυπτογραφία. Είναι κώδικας αντικατάστασης στον οποίο κάθε γράμμα του κειμένου αντικαθίσταται από κάποιο άλλο γράμμα με σταθερή απόσταση κάθε φορά στο αλφάβητο.
Για παράδειγμα, με μετατόπιση 3, το Α θα αντικαθίστατο από το Δ, το Β από το Ε, και ούτω καθεξής.

Το βήμα κωδικοποίησης που εκτελείται από τον κώδικα του Καίσαρα συχνά ενσωματώνεται ως τμήμα ενός πιο πολύπλοκου πλαισίου, όπως ο κώδικας Vigenère, και έχει ακόμη σύγχρονη εφαρμογή στο σύστημα ROT13. Όπως με όλους τους μονοαλφαβητικούς κώδικες αντικατάστασης, ο κώδικας του Καίσαρα "σπάει" εύκολα και στη σύγχρονη εφαρμογή του δεν παρέχει ουσιαστικά κάποια ασφάλεια επικοινωνίας.

Ο κώδικας του Καίσαρα μπορεί εύκολα να σπάσει ακόμα και με ciphertext-only scenario. Μπορούν να ληφθούν υπόψη δύο περιπτώσεις.

α) Ο επιτιθέμενος γνωρίζει (ή υποθέτει) ότι έχει χρησιμοποιηθεί κάποιου είδους κώδικας απλής αντικατάστασης, αλλά όχι ότι πρόκειται για τον κώδικα του Καίσαρα συγκεκριμένα.

β) Ο επιτιθέμενος γνωρίζει ότι πρόκειται για κώδικα του Καίσαρα, αλλά δεν γνωρίζει την τιμή της μετατόπισης.

Στην πρώτη περίπτωση ο κώδικας μπορεί να σπάσει χρησιμοποιώντας τις ίδιες τεχνικές όπως και σε ένα γενικό απλό κώδικα αντικατάστασης, όπως η ανάλυση συχνότητας ή οι λέξεις μοτίβα. Κατα την διάρκεια της λύσης, είναι πιθανό ότι ο επιτιθέμενος θα διαπιστώσει σύντομα την κανονικότητα στη λύση και θα συμπεράνει ότι χρησιμοποιείται ο κώδικας του Καίσαρα.

Playfair

Ο αλγόριθμος κρυπτογράφησης Playfair ήταν ο πρώτος πρακτικός αλγόριθμος που κρυπτογραφεί ζεύγη γραμμάτων έναντι μεμονωμένων όπως συνηθιζόταν. Το κλειδί είναι μια λέξη π.χ 'monarchy' η οποία χρησιμοποιείται για την παραγωγή του παρακάτω τετραγωνικού πίνακα.

m o n a r
c h y b d
e f g i k
l p q s t
u v w x z

Κάθε ακολουθία 25 γραμμάτων μπορεί να αποτελέσει κλειδί αρκεί να έχει όλα τα γράμματα μια φορά και να μην έχει επαναλήψεις. Αξίζει να σημειωθεί πως δεν υπάρχει 'j' στον παραπάνω πίνακα καθώς είναι συνδυασμένο με το 'i'.

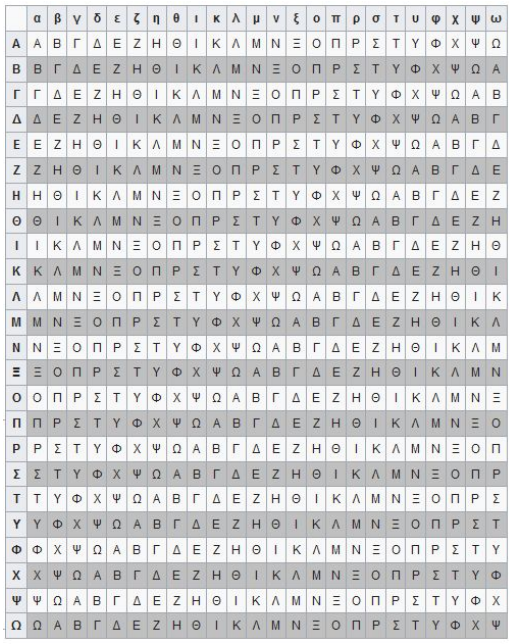
1) Στο κείμενο που δίνεται ως plaintext αντικαθίστανται τα διπλά γράμματα ως εξής "hammer"->"haxmer".
2) Συμπληρώνουμε με x ώστε το plaintext να έχει ζυγό αριθμό γραμμάτων.
3) Διαχωρίζουμε το plaintext σε ζευγη γραμμάτων.
4) Με την χρήση του πίνακα παραπάνω
    a) Αν τα γράμματα είναι σε διαφορετική στήλη και γραμμή αντικαθιστούμε το ζεύγος με τα γράμματα της ίδιας γραμμής αλλα της στήλης του άλλου γράμματος 'ha' -> 'bo', 'es' -> 'il'.
    b) Αν είναι στην ίδια γραμμή τα αντικαθιστούμε με τα δεξιά τους γράμματα 'ma' -> 'or', 'lp' -> 'pq'.
    c) Αν είναι στην ίδια στήλη τα αντικαθιστούμε με τα αμέσως από κάτω τους 'rk' -> 'dt', 'pv' -> 'vo'.

Ο Playfair "σπάει" πολύ πιο δύσκολο καθώς η συχνοτική ανάλυση που χρησιμοποιείται σε αλγορίθμους απλής αντικατάστασης δεν λειτουργεί. Φυσικά, μπορεί να γίνει συχνοτική ανάλυση αλλα αυτή πρέπει να εκπονηθεί επι των 25*25=625 πιθανών διγραμμάτων έναντι των 25 γραμμάτων. Επομένως η ανάλυση χρειάζεται πολύ περισσότερο κρυπτοκείμενο για να λειτουργήσει.

Vigenère

Ο αλγόριθμος κρυπτογράφησης Vigenère είναι μία μέθοδος κρυπτογράφησης σε αλφαβητικό κείμενο στο οποίο εφαρμόζονται διαφορετικοί αλγόριθμοι κρυπτογράφησης Καίσαρα με βάση τη θέση των γραμμάτων μιας λέξης ή φράσης κλειδί. Για την κρυπτογράφηση, ένας πίνακας του αλφάβητου μπορεί να χρησιμοποιηθεί, ως πίνακας αντικατάστασης (*tabula Recta*, *Vigenère square*, *ή Vigenère table*). Αποτελείται από το αλφάβητο, που αναγράφεται σε διαφορετικές γραμμές (ή στήλες) τόσες φορές όσες και τα γράμματα του αλφαβήτου και κάθε αλφάβητο μετατοπίζεται κυκλικά σε σχέση με το προηγούμενο αλφάβητο, ώστε να υπάρχουν όλοι οι πιθανοί αλγόριθμοι κρυπτογράφησης του Καίσαρα.

Κατά τη διαδικασία κρυπτογράφησης, χρησιμοποιείται διαφορετικό αλφάβητο σε κάθε ένα από τα γράμματα. Το αλφάβητο που χρησιμοποιείται σε κάθε γράμμα εξαρτάται από μια επαναλαμβανόμενη λέξη-κλειδί.



	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω
Α	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω
Β	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α
Γ	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β
Δ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ
Ε	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ
Ζ	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε
Η	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ
Θ	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η
Ι	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ
Κ	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι
Λ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ
Μ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ
Ν	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ
Ξ	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν
Ο	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ
Π	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
Ρ	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π
Σ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ
Τ	Τ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ
Υ	Υ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ
Φ	Φ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ
Χ	Χ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ
Ψ	Ψ	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ
Ω	Ω	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ

Ασφάλεια

Η ασφάλεια στον αλγόριθμο Vigenère εξαρτάται από το μήκος του κλειδιού, όχι μόνο για να αποφευχθεί επίθεση εξαντλητικής αναζήτησης αλλά και για την απόκρυψη της κατανομής συχνοτήτων των γραμμάτων. Το πρώτο βήμα για να ανακαλυφθεί το κλειδί είναι να βρεθεί το μήκος του. Αν βρεθεί αυτό, τότε το πρόβλημα κρυπτανάλυσης μπορεί να αναχθεί σε πολλαπλές μονοαλφαβητικές αντικαταστάσεις. Μία μέθοδος κρυπτανάλυσης για τον προσδιορισμό του μήκους του κλειδιού είναι η μέθοδος του Kasiski. Η μέθοδος αυτή ελέγχει για συμβολοσειρές που εμφανίζονται πολλές φορές μέσα στο κείμενο. Αυτές οι συμβολοσειρές πρέπει να είναι πάνω από 3 χαρακτήρες για να αυξηθεί το ποσοστό επιτυχίας της μεθόδου. Ύστερα, χρησιμοποιείται η απόσταση μεταξύ των διαδοχικών εμφανίσεων των συμβολοσειρών, η οποία έχει μεγάλη πιθανότητα να είναι πολλαπλάσιο του μήκους του κλειδιού. Κάνοντας αυτό για τις διάφορες συχνά εμφανιζόμενες συμβολοσειρές, μπορούμε να εξάγουμε το μήκος του κλειδιού απλοποιώντας το πρόβλημα.

Για να γίνει πιο κατανοητή εις βάθος η συγκεκριμένη μέθοδος αξίζει να σημειωθούν τα εξής. Αν το κλειδί Κ είναι μήκους 3 τότε υπάρχουν μόνο 3 διαφορετικοί τρόποι να τοποθετηθεί πάνω από μια αρχική λέξη του αρχικού κειμένου. Επομένως, μια λέξη του αρχικού κειμένου που εμφανίζεται 3 φορές θα πρέπει να έχει κρυπτογραφηθεί τουλάχιστον δύο φορές με την ίδια αντιστοίχιση.

Αρχικό μήνυμα (μ) = tobeornottobethatisthe

Κλειδί (δ) = runrunrunrunrunrunrun

Κρυπτογραφημένο (ψ) = **kiov**ieeig**kiov**nurnvjnuv

Vernam

Ο αλγόριθμος vernam είναι μια συμμετρική κρυπτογράφηση ροής στην οποία το απλό κείμενο (plaintext) συνδυάζεται με την κλειδοροή (keystream) που έχει το ίδιο μήκος χρησιμοποιώντας την πύλη XOR. Σε περίπτωση που η κλειδοροή είναι πραγματικά τυχαία και χρησιμοποιείται μόνο μια φορά τότε ο αλγόριθμος αυτός δεν μπορεί να σπάσει (αποτελεί one-time pad). Αν όμως η ίδια κλειδοροή χρησιμοποιηθεί σε δύο μηνύματα (γνωστά) σε κάποιον τότε έχουμε το εξής.

Ciphertext1 XOR Keystream = Plaintext1
Ciphertext2 XOR Keystream = Plaintext2

Ciphertext1 XOR Ciphertext2 = Plaintext1 XOR Plaintext2

Αναιρείται δηλαδή η επίδραση της κλειδοροής εκμηδενίζοντας την ασφάλεια του αλγορίθμου.

Hill Cipher

Για την κρυπτογράφηση με τον Hill πρέπει να επιλεγεί ένα κλειδί το οποίο θα είναι ένας τετραγωνικός πίνακας που έχει αντίστροφο mod 26. Για να ισχύει αυτό πρέπει η ορίζουσα του πίνακα να είναι σχετικά πρώτη με το 26.

π.χ.

$$M_2 = \begin{bmatrix} 2 & 3 \\ 5 & 3 \end{bmatrix}, M_3 = \begin{bmatrix} 25 & 2 & 11 \\ 19 & 5 & 12 \\ 21 & 22 & 6 \end{bmatrix}$$

det(M2) = -9 όμως (-9 mod 26)=17 και το 17 είναι σχετικά πρώτος με το 26.

det(M3) = -2131 όμως (2131 mod 26)=1 και το 1 είναι σχετικά πρώτος με το 26.

Εάν ένας πίνακας 2χ2 επιλεγεί ως κλειδί τότε το plaintext πρέπει να γίνει συμπληρωθεί (συνήθως με X) ώστε να είναι πολλαπλάσιο του 2. Αντιστοίχως για έναν πίνακα NχN το κείμενο συμπληρώνεται για να είναι πολλαπλάσιο του N.

Για N=2, “CAREFUL” θα γινόταν “CAREFULX”
Για N=2, “SPORTS” θα γινόταν “SPORTS”

Για N=3, “CAREFUL” θα γινόταν “CAREFULXX”

Όταν το κείμενο έχει το κατάλληλο μέγεθος, χρησιμοποιούμε γράμματα σε blocks μεγέθους N και τα μετατρέπουμε σε ενα διάνυσμα στήλης το οποίο εν συνεχεία πολλαπλασιάζεται με τον πίνακα-κλειδί.

Ασφάλεια

Η σημαντική παρατήρηση είναι πως κάθε γραμμή του πίνακα-κλειδί χρησιμοποιείται για την κρυπτογράφηση ενός γράμματος ανεξαρτήτως από τον υπόλοιπο πίνακα.

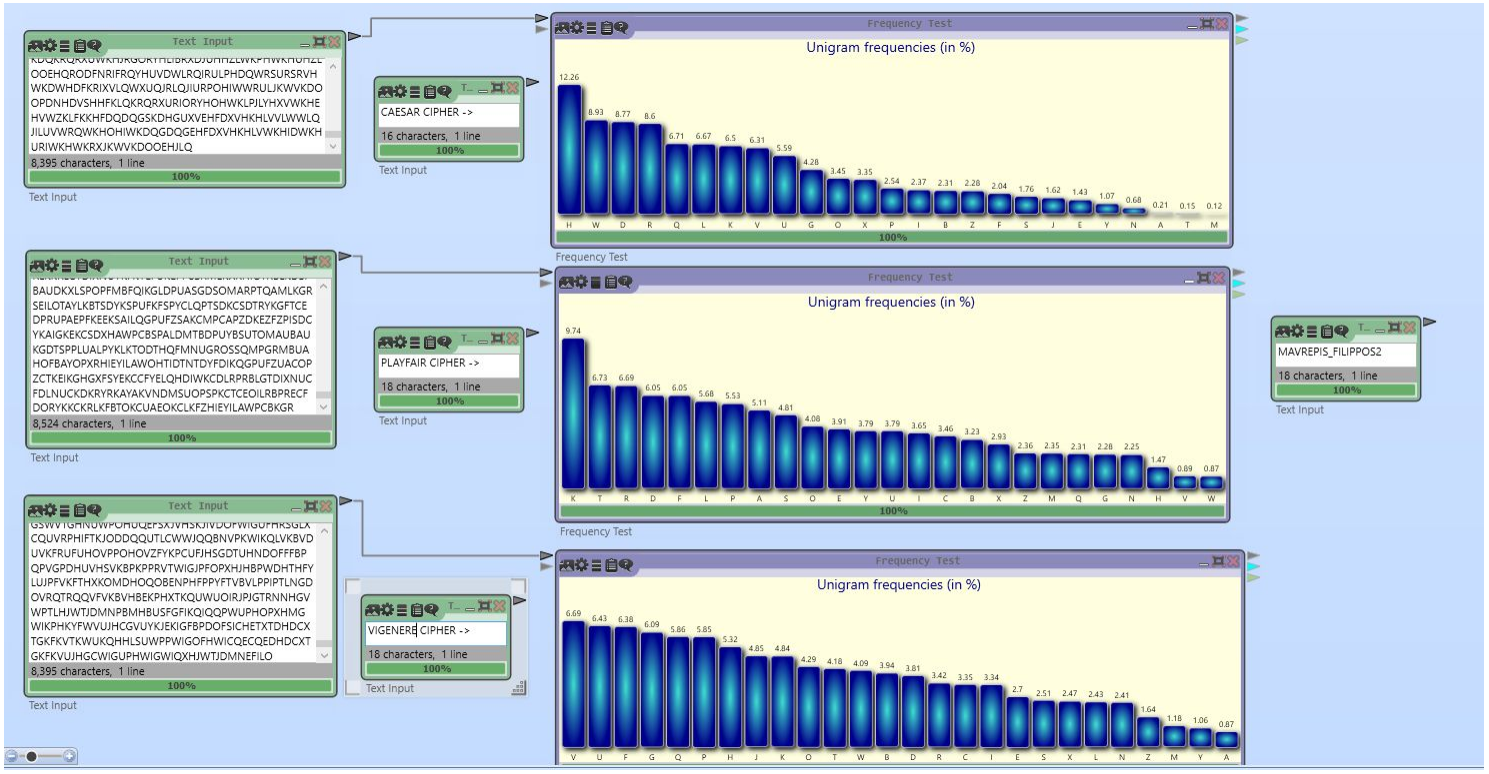
Για έναν πίνακα μεγέθους N υπάρχουν 26^NχN μοναδικά κλειδιά (θα είναι λιγότερα καθώς δεν έχουν όλοι οι πίνακες αντίστροφο). Αν όμως ελεγχθεί η κάθε γραμμή ξεχωριστά υπάρχει μόνο 26^N κλειδιά που πρέπει να ελεγχθούν.

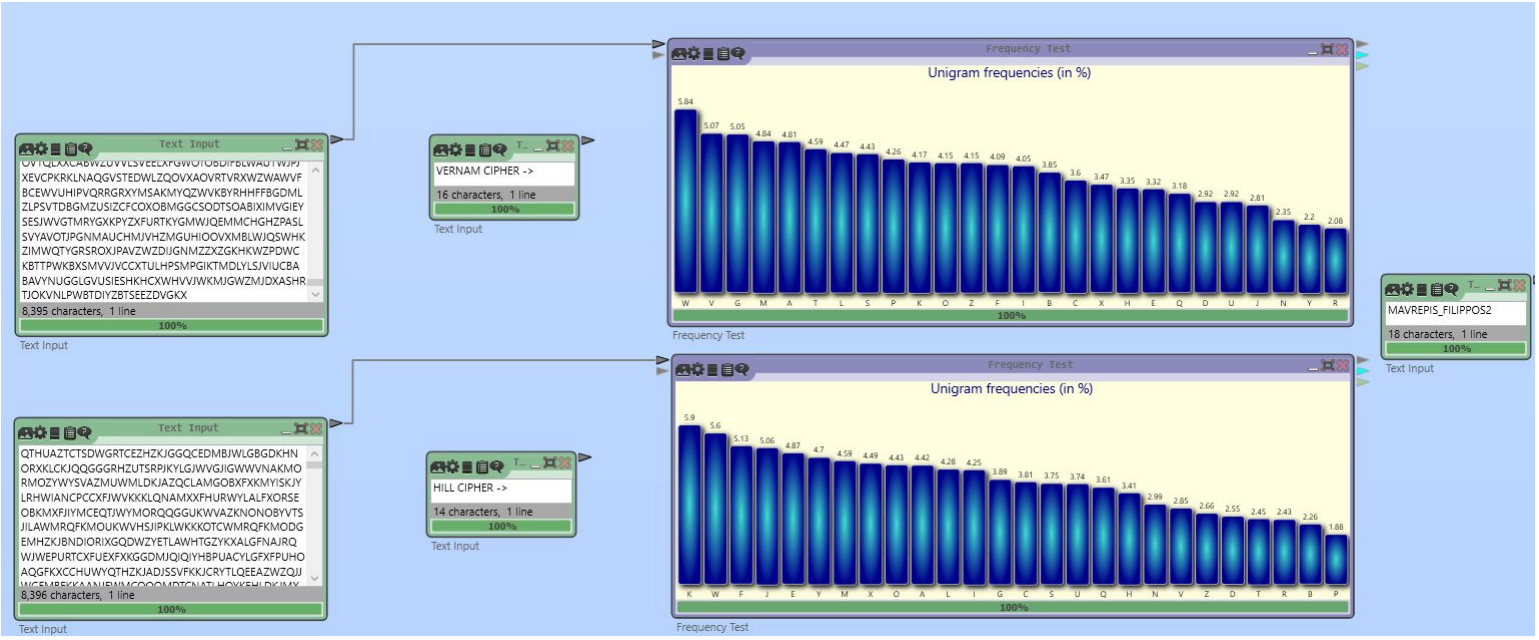
Το πρώτο βήμα είναι εύρεση του μεγέθους N του πίνακα (το διαιρέτης του μήκους του κειμένου). Μετά απαιτείται εξαντλητική αναζήτηση και έλεγχος καλής προσαρμογής Chi-squared. Γενικότερα αυτός ο αλγόριθμος είναι ισχυρός απέναντι σε επιθέσεις ciphertext-only αλλά μπορεί να σπάσει εύκολα με known-plaintext επιθέσεις.

Ανάλυση Συχνοτήτων

Στις παρακάτω εικόνες φαίνονται οι συχνοτικές αναλύσεις για τα κρυπτοκείμενα που παράγονται από τους αντίστοιχους αλγορίθμους. Η επιθυμητή εικόνα για ένα τέτοιο γράφημα από την μεριά του ατόμου που κρυπτογραφεί το μήνυμα θα ήταν κάτι που θα προσομοιώνει την κανονική κατανομή καθώς έτσι ο επιτιθέμενος δεν μπορεί να εξάγει (απευθείας) γλωσσικές συσχετίσεις.

Γενικότερα όλοι οι αλγόριθμοι που παρουσιάζονται παρακάτω φέρουν μια φθίνουσα πορεία η οποία όμως δεν ακολουθεί (αισθητά) την zipf κατανομή που τείνουν να ακολουθούν οι γλώσσες και τα γλωσσικά μοντέλα. Αυτό που αλλάζει στις διάφορες συχνοτικές αναλύσεις είναι τα ποσοστιαία μεγέθη των γραμμάτων. Για παράδειγμα βλέπουμε ότι στον αλγόριθμο του καίσαρα το πρώτο γράμμα έχει συχνότητα 12.36% ενώ το τελευταίο 0.12%. Αντιστοίχως στον venram βλέπουμε ότι το πρώτο γράμμα έχει 5.84% και το τελευταίο 2.06% φέρνοντας πιο κοντά το μέγιστο και το ελάχιστο ποσοστό εμφάνισης κάποιου γράμματος.

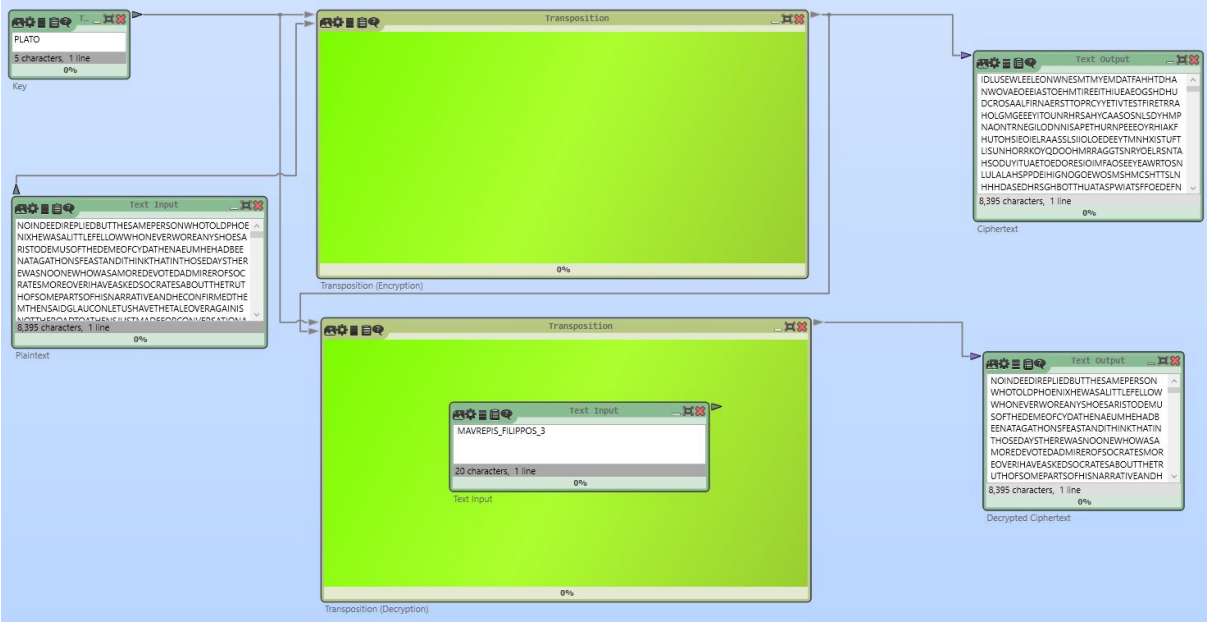




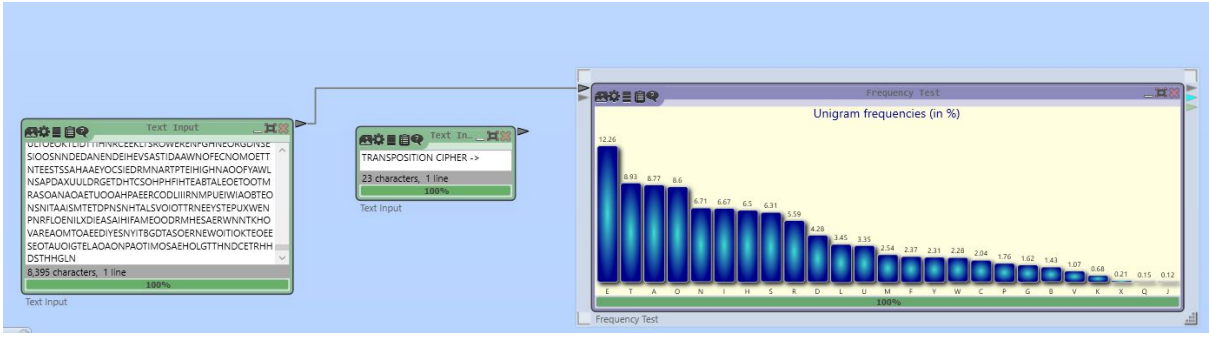
Transposition

Στην κρυπτογραφία ο transposition αλγόριθμος εκτελεί μεταθέσεις με καθορισμένο τρόπο στο αρχικό κείμενο για να παραχθεί το κρυπτοκείμενο. Ύστερα για

την αποκρυπτογράφηση του χρησιμοποιείται η αντίστροφη συνάρτηση μετάθεσης προκειμένου να επαναπροκύψει το αρχικό κείμενο.



Όπως καταλαβαίνουμε και βλέπουμε παρακάτω οι συχνότητες των γραμμάτων δεν μεταβάλλονται καθόλου καθώς αυτά αλλάζουν μόνο θέσεις. Επομένως ο συγκεκριμένος αλγόριθμος (όταν το μήκος του κρυπτοκειμένου είναι μεγάλο) είναι επιρρεπής σε επιθέσεις συχνοτικής ανάλυσης.

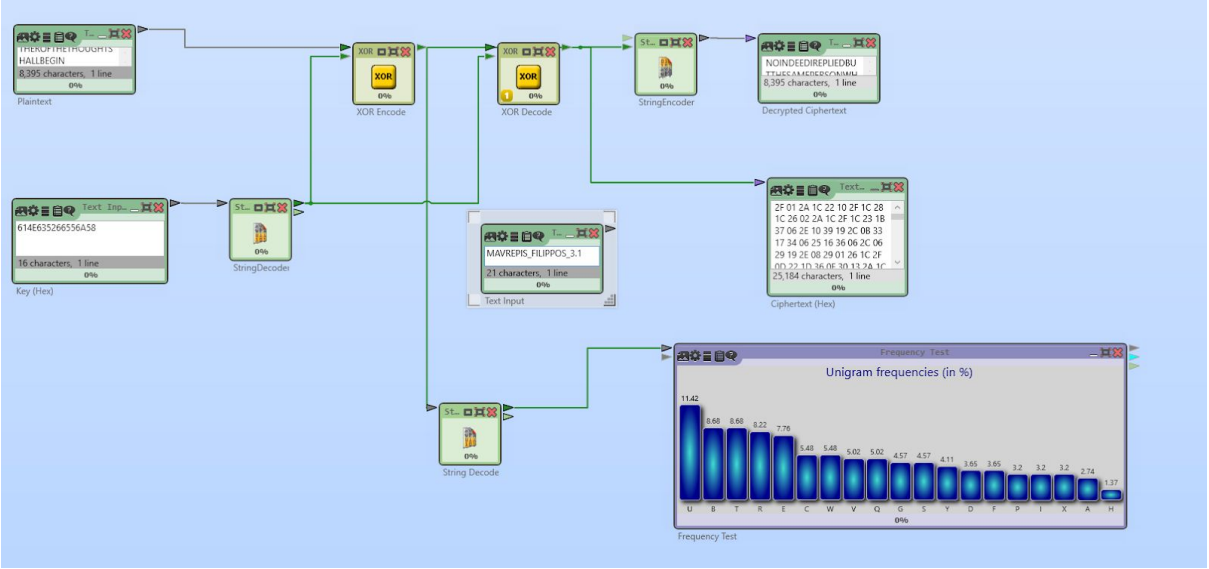


3. Συγχρονοι αλγοριθμοι κρυπτογραφησης

XOR

Παρακάτω φαίνεται η διάταξη του XOR για την κρυπτογράφηση/ αποκρυπτογράφηση καθώς και για την συχνοτική ανάλυση. Σημειώνεται πως προκειμένου να μπορεί να εκτελεστεί συχνοτική ανάλυση στην έξοδο του αλγορίθμου που όπως φαίνεται είναι δεκαεξαδική πρέπει πρώτα αυτή να μετατραπεί μέσω του block “string decode”.

Ο αλγόριθμος XOR χρειάζεται κλειδί μήκους ίσο ή μεγαλύτερο από το κείμενο το οποίο κρυπτογραφεί. Για αυτόν το λόγο το cryptool σε περίπτωση που χορηγηθεί κλειδί μικρότερου μήκους το επεκτείνει ώστε να μπορεί να εκτελεστεί η κρυπτογράφηση η οποία δεν είναι άλλη παρά το XOR του αρχικού κειμένου και του κλειδιού.



DES

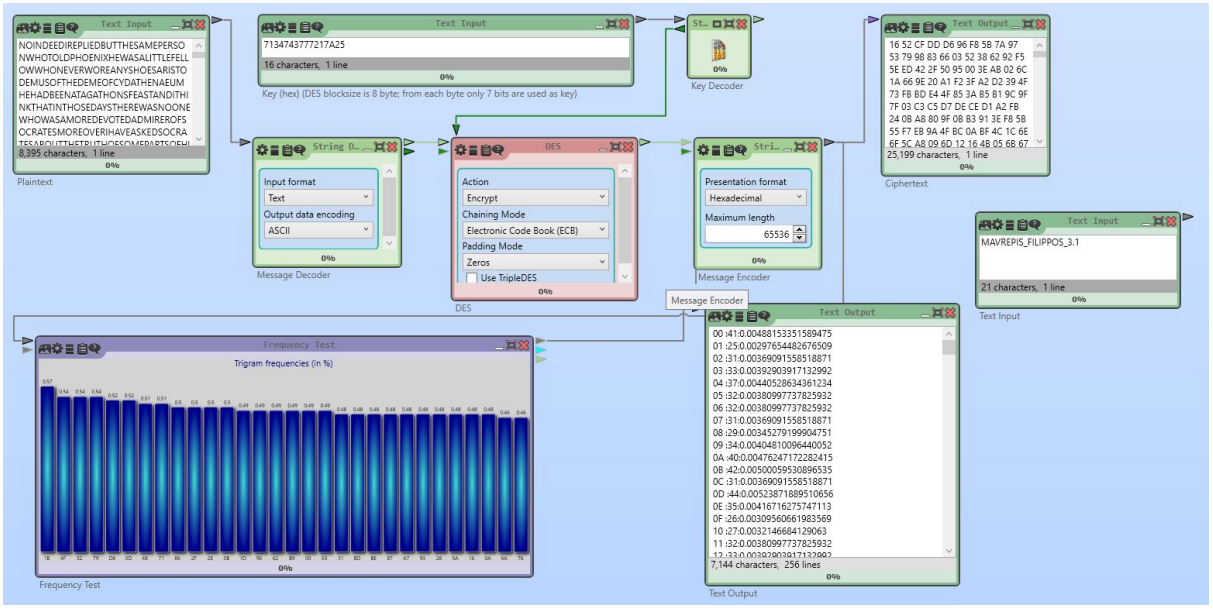
Ο αλγόριθμος DES είναι ένας συμμετρικός αλγόριθμος με σχετικά μικρό κλειδί (64 bit). Στηρίζεται στην δομή τύπου Feistel την οποία χρησιμοποιεί 16 φορές κατά την κρυπτογράφηση. Ο DES είναι block cipher αλγόριθμος, αυτό σημαίνει πως παίρνει μια σειρά από bit απλού κειμένου (σταθερού μήκους) και τα κρυπτογραφεί (μέσω κάποιων ενεργειών). Ο DES χρησιμοποιεί ένα ακόμα κλειδί το οποίο παράγεται από το πρώτο προκειμένου η αποκρυπτογράφηση να μπορεί να πραγματοποιηθει μονο απο εκείνους που γνωρίζουν το συγκεκριμένο κλειδί που χρησιμοποιήθηκε για την κρυπτογράφηση.

Το κλειδί που φαίνεται να αποτελείται από 64-bit πρακτικά είναι μόνο 56-bit καθώς τα υπόλοιπα 8-bit χρησιμοποιούνται μόνο για parity check. Γενικότερα για να αυξήσουμε

την ασφάλεια του DES και οποιουδήποτε αλγορίθμου κρυπτογράφηση block πρέπει να προστεθεί ο τρόπος λειτουργίας (mode of operation).

Επιγραμματικά φαίνονται οι εξής:

	Type	Initialization Vector	Error Propagation?
Electronic Code Book (ECB)	Block	No	No
Cipher Block Chaining (CBC)	Block	Yes	Yes
Cipher Feedback (CFB)	Stream	Yes	Yes
Output Feedback (OFB)	Stream	Yes	No
Counter Mode (CTR)	Stream	Yes	No



Triple-DES

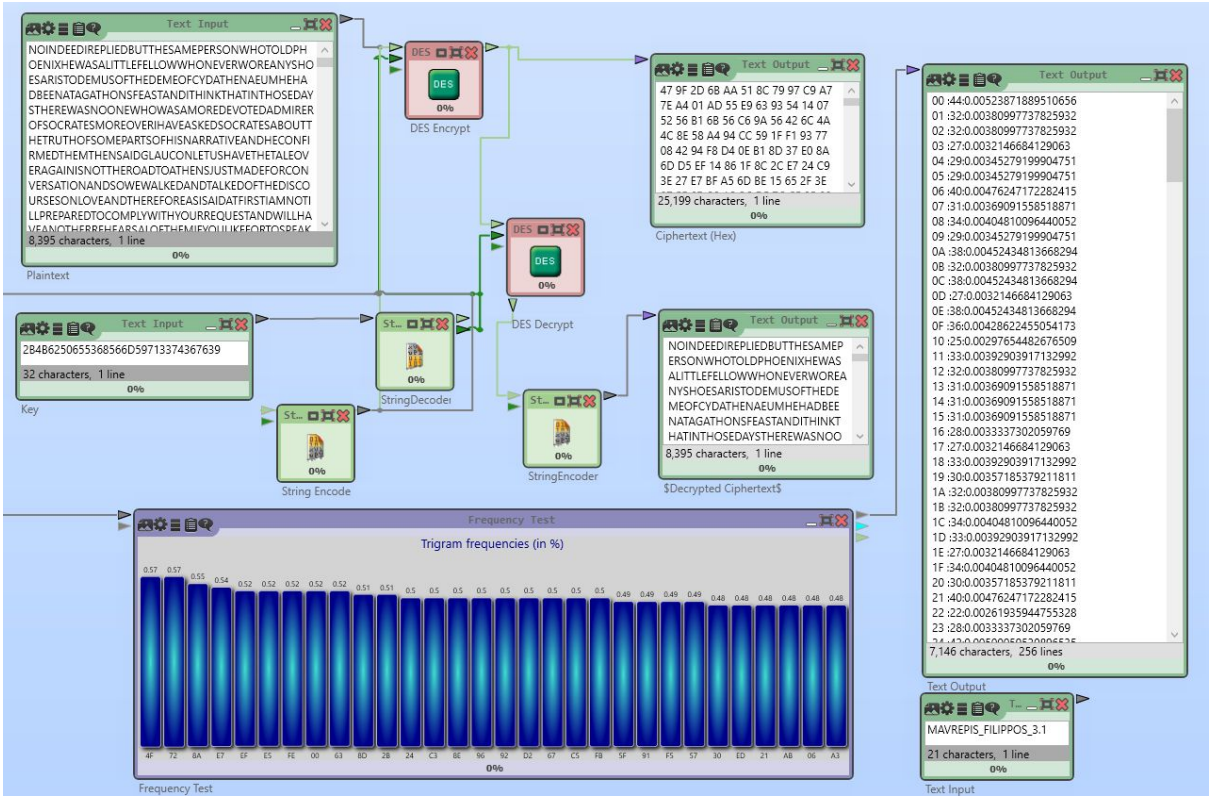
Δεδομένου του γεγονότος ότι το κλειδί του DES θεωρείται πολύ μικρό πλέον λόγω των μοντέρνων τεχνικών κρυπτανάλυσης και της ύπαρξης υπερυπολογιστών ο αλγόριθμος 3-DES εφαρμόζει τρεις φορές τον DES σε κάθε block. Επομένως ο triple-des έχει 3 κλειδιά K1,K2,K3 (των οποίων το ωφέλιμο μήκος είναι 56-bit) και το κρυπτοκείμενο προκύπτει ως εξής:

$$\text{ciphertext} = E_{K3}(D_{K2}(E_{K1}(\text{plaintext}))).$$

Κρυπτογράφηση με το K1 αποκρυπτογράφηση με το K2 και κρυπτογράφηση με το K3. Αντιστοίχως η αποκρυπτογράφηση προκύπτει ανάλογα.

$$\text{plaintext} = D_{K1}(E_{K2}(D_{K3}(\text{ciphertext}))).$$

Εφόσον πλέον υπάρχουν 3 κλειδιά προκύπτει το ερώτηση για την επιλογή τους. Η πρώτη πιθανή επιλογή είναι τα K1,2,3 να είναι διαφορετικά μεταξύ τους. Αυτή είναι η ισχυρότερη κρυπτογράφηση με 168-bit. Η επόμενη επιλογή είναι το K1=K3, προσφέροντας ένα κλειδί μικρότερου μήκους 112-bit . Φυσικά υπάρχει και η τρίτη επιλογή που είναι ότι K1=K2=K3 το οποίο είναι προφανές ότι είναι ισοδύναμο με τον απλό DES.



Σίγουρα αποτελεί μια πιο ασφαλή έκδοση του DES καθώς έχει καταστήσει την εξαντλητική αναζήτηση πολύ πιο δύσκολη (αυξάνοντας το κλειδί). Εν γένει ο αλγόριθμος του DES είναι ασφαλής επομένως το μόνο πρακτικό πρόβλημα του Triple-DES την εποχή που δημιουργήθηκε ήταν η έλλειψη αποδοτικής υλοποίησης σε λογισμικό.

AES

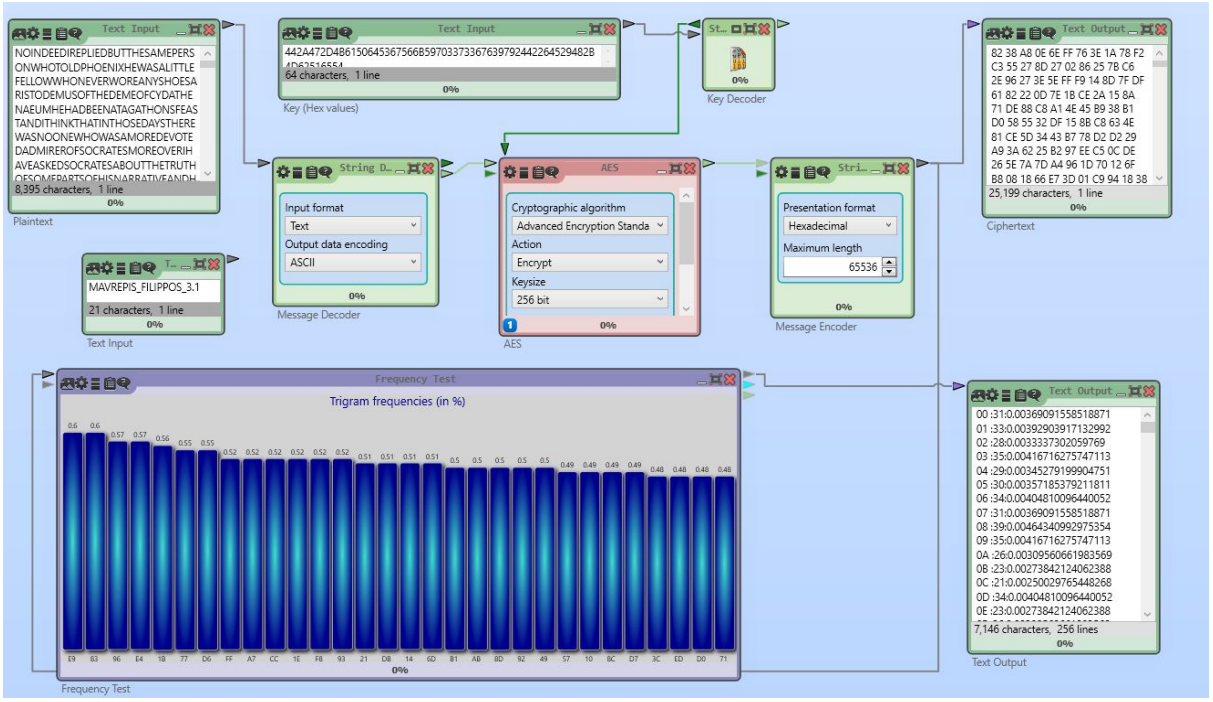
Ο αλγόριθμος AES (Advanced Encryption Standard) αποτελεί έναν block cipher αλγόριθμο που στηρίζεται σε μεταθέσεις και αντικαταστάσεις. Αντιθέτως με τον DES δεν χρησιμοποιεί Feistel και έχει μήκος κλειδιού 128,192 ή 256 bits επιδρώντας πάντα σε 128 bit blocks. Οι υπολογισμοί συνήθως εκτελούνται σε εναν 4χ4 πίνακα από bytes που ονομάζεται state. Αναλόγως το μέγεθος του κλειδιού γίνονται και αντίστοιχοι γύροι του αλγορίθμου.

Μια σύντομη περιγραφή του αλγορίθμου είναι η εξής:

1. Δημιουργία του κλειδιού του γύρου
2. Κάθε byte του state γίνεται bitwise xor με ένα byte του κλειδιού

3. Επανάληψη γύρων
    a. Αντικατάσταση bytes (με μη γραμμικό τρόπο)
    b. Αντιμετάθεση γραμμών
    c. Ανάμειξη στηλών
    d. βήμα 2
4. Τελικός γύρος
    a. Αντικατάσταση bytes
    b. Αντιμετάθεση γραμμών
    c. βήμα 2

Ο συγκεκριμένος αλγόριθμος είναι πολύ ασφαλής και η χρήση συχνοτικής ανάλυσης δεν βοηθάει επαρκώς στην επίλυση της κρυπτογράφησης. Όπως φαίνεται και παρακάτω τα γράμματα του κρυπτοκειμένου ακολουθούν σχεδόν ομοιόμορφη κατανομή (μεταφέρουν δηλαδή την μικρότερη δυνατή πληροφορία).



RSA

Ο RSA απαιτεί το δημόσιο εκθετικό κρυπτογράφησης e, το δημόσιο modulus **N**, και το ιδιωτικό εκθετικό αποκρυπτογράφησης d και την παραγοντοποίηση $N = pq$, όπου

$$ed = k\varphi(\mathrm{N}) + 1 \qquad (1)$$

Αν m είναι το μήνυμα που θέλω να μεταδώσω τότε το κρυπτογράφημα είναι

$$c = m^e \bmod N \qquad (2)$$

Για την αποκρυπτογράφηση εφόσον ληφθεί το c τότε έχουμε

$$m = c^d \bmod n \equiv m^{ed} \bmod N \qquad (3)$$

Όμως ξέρουμε ότι $ed \equiv 1 \ (mod\ p-1)$ και $ed \equiv 1 \ (mod\ q-1)$ οπότε από Fermat

$$m^{ed} \equiv m^1 \equiv m \bmod p-1 \quad (4) m^{ed} \equiv m^1 \equiv m \bmod q-1 \quad (5)$$

Επίσης οι αριθμοί p,q είναι πρώτοι μεταξύ τους επομένως μέσω του Κινεζικού Θεωρήματος Υπολοίπων έχουμε

$$m^{ed} \equiv m \bmod N \qquad (6)$$

Για να βρούμε το κλειδί κρυπτογράφησης του RSA αυτό που χρειαζόμαστε στην πραγματικότητα είναι το $\varphi(N)$.

$$\varphi(N) = (p-1)(q-1) \ (7) \qquad \varphi(N) = pq - (p+q) + 1 \approx N \quad (8)$$

Το τελευταίο μέρος της δεύτερης σχέσης προκύπτει καθώς οι αριθμοί p, q είναι πολύ μεγάλοι αριθμοί επομένως η τάξη μεγέθους του γινομένου είναι πολύ μεγαλύτερη της τάξης μεγέθους της πρόσθεσης τους.

Ακόμα από την σχέση (1) :

$$ed - k\varphi(N) = 1 \qquad (10) \frac{e}{\varphi(N)} - \frac{k}{d} = \frac{1}{d\varphi(N)} \ (11) \text{απο } (8) \quad \frac{e}{N} \approx \frac{k}{d} \qquad (12)$$

Η τελευταία σχέση προκύπτει καθώς η ποσότητα $\frac{1}{d\varphi(N)}$ πλησιάζει το μηδέν καθώς $d\varphi(N)$ είναι ένας πολύ μεγάλος αριθμός.

Επομένως από την σχέση (12) προκύπτει ότι αν μπορώ να προσεγγίσω την ποσότητα $\frac{e}{N}$ με κάποιον ρητό αριθμό ($\frac{k}{d}$), τότε υπάρχει πιθανότητα ο παρονομαστής αυτού να είναι το d

(που χρειάζομαι για την αποκρυπτογράφηση)

Εύρεση του $\frac{e}{N}$

Μπορούμε να χρησιμοποιήσουμε τα συνεχή κλάσματα για να βρούμε ένα σύνολο κλασμάτων (τα συγκλίνοντα) $\frac{k}{d}$ που προσεγγίζουν το $\frac{e}{N}$.

Έστω ότι βρίσκουμε $\frac{k}{d} \approx \frac{e}{N}$, τότε παρατηρούμε τα εξής:

- Εφόσον $ed \equiv 1 \ mod\ \varphi(N)$ και το $\varphi(N)$ είναι ζυγός αριθμός, τότε το d πρέπει να είναι μονός αριθμός. Επομένως αν το d είναι ζυγός, συνεχίζουμε στο επόμενο συγκλίνων κλάσμα. (Παρατήρηση 1)

- Εφόσον το φ(N) πρέπει να είναι ακέραιος αριθμός, θα ελέγξουμε αν το $\frac{ed-1}{k}$ είναι ακέραιος αριθμός. Αν δεν είναι συνεχίζουμε στο επόμενο συγκλίνων κλάσμα. (Παρατήρηση 2)

Εάν το d του παρονομαστή ικανοποιεί αυτές τις δυο συνθήκες τότε θα μπορούσαμε δυνητικά να ελέγξουμε αν λειτουργεί ως εκθετικό αποκρυπτογράφησης αλλά θα κάνουμε κάτι πιο απλό που βασίζεται σε τετραγωνικές εξισώσεις.

Τετραγωνικές εξισώσεις

Ας υποθέσουμε τους πρώτους αριθμούς p, q των οποίων το γινόμενο είναι το N. Τότε από την σχέση (8) έχουμε ότι $p+q = N - \varphi(\mathrm{N}) + 1$ (13).

Θεωρούμε την εξίσωση $(x-p)(x-q) = 0$ με ριζες p, q (τους πρώτους παράγοντες του N).

Τότε έχουμε

$$(x-p)(x-q) = 0 \quad (14)$$

$$x^2 - (p+q)x + pq = 0 \quad (15) \qquad x^2 - (N - \varphi(\mathrm{N}) + 1)x + N = 0 \quad (16)$$

Αν έχουμε τις σωστές τιμές για τα e, d, k μπορούμε να λύσουμε θεωρητικά την εξίσωση (16). Αν η τιμή του $\varphi(\mathrm{N})$ είναι σωστή, τότε οι ρίζες της εξίσωσης αυτής θα είναι ακέραιοι αριθμοί και οι παράγοντες του N.

Παράδειγμα

Έστω κρυπτογράφηση RSA με N = 64741 και δημόσιο εκθετικό e = 42667. Να βρεθεί το εκθετικό αποκρυπτογράφησης d.

Βήμα 1ο: Προσπαθούμε να προσεγγίσουμε το $\frac{e}{N}$ με το $\frac{k}{d}$, όπου $\frac{e}{N} = \frac{42667}{64741}$. Θα χρησιμοποιήσουμε τον αλγόριθμο του Ευκλείδη για να βρούμε τα διαδοχικά συγκλίνοντα κλάσματα του $\frac{e}{N}$.

Βήμα 2ο: Χρήση αλγορίθμου Ευκλείδη.

- 42667 div 64741 = 0, υπόλοιπο 42667 αρά $\frac{k}{d} \approx \frac{0}{1}$ k=0, d=1 το οποίο προφανώς δεν μπορεί να ισχύει οπότε συνεχίζουμε στην επόμενη συγκλίνουσα.
- 64741 div 42667 = 1, υπόλοιπο 22074 αρά $\frac{k}{d} \approx 0 + \frac{1}{1}$, αρά k=1, d=1 το οποίο πάλι προφανώς δεν είναι λειτουργεί επομένως προχωράμε παρακάτω.

- 42667 div 22074 = 1, υπόλοιπο 20593 αρά $\frac{k}{d} \approx 0 + \frac{1}{1+\frac{1}{1}} = \frac{1}{2}$, αρά k= 1, d=2

  όμως από Παρατήρηση 1 ξέρουμε ότι το d πρέπει να είναι μονός αριθμός προκειμένου να είναι σωστό, επομένως συνεχίζουμε.

- 22074 div 20593 = 1, υπόλοιπο 1481 αρά $\frac{k}{d} \approx 0 + \frac{1}{1+\frac{1}{1+\frac{1}{1}}} = \frac{2}{3}$, αρά k=2 και d =3

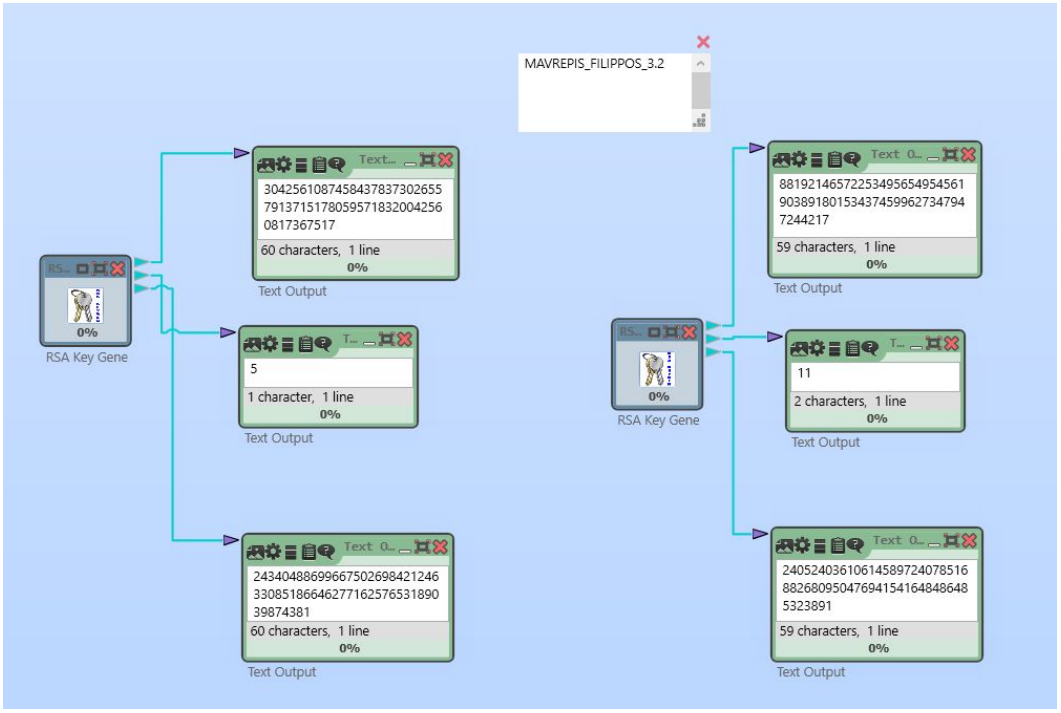
  το d είναι μονός αριθμός αρά επιβεβαιώνει την πρώτη παρατήρηση και το $\varphi(N) = \frac{ed-1}{k}$ πρέπει να είναι ακέραιος αριθμός. Μετά από αντικατάσταση το φ(N) = 64000, αρά επιβεβαιώνεται και η δεύτερη παρατήρηση. Επομένως τώρα μπορούμε να ελέγξουμε αν η αντίστοιχη εξίσωση (16) έχει ακέραιες λύσεις.

$$x^2 - (742)x + 64741 = 0$$

Η εξίσωση αυτή έχει λύσεις ακέραιες χ_1=641, χ_2=101.

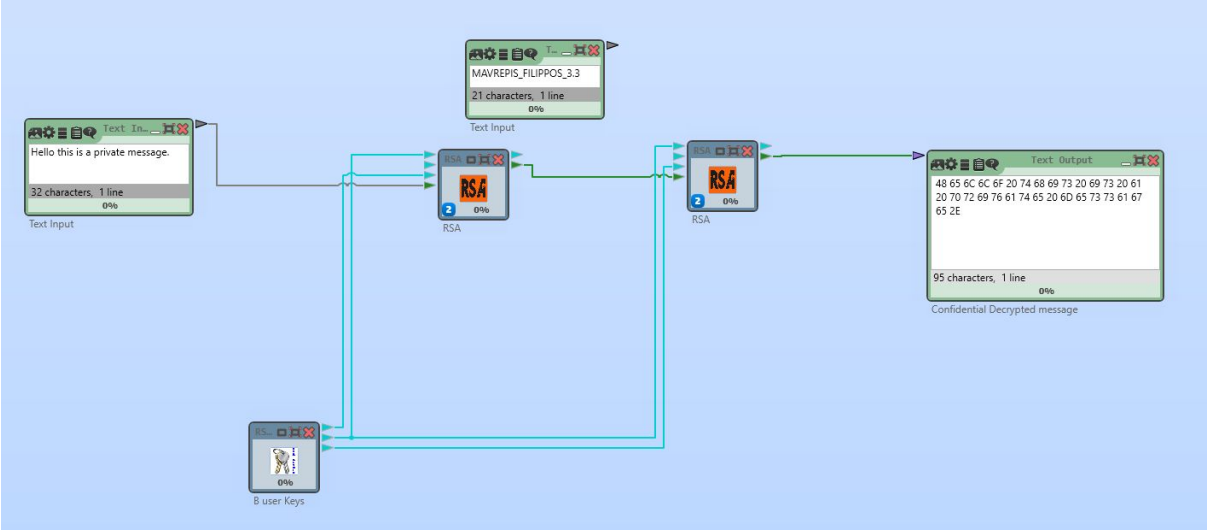
Οι λύσεις τις εξισώσεις 641 * 101 = 64741 επομένως έχουμε μια παραγοντοποίηση του N και το ιδιωτικό εκθετικό είναι το d = 3

Αντιστοίχως μέσω του Cryptool κατασκευάζονται οι τριπλέτες N, e, d που χρειάζονται για την παρασκευή του κλειδιού του καθενός χρήστη.



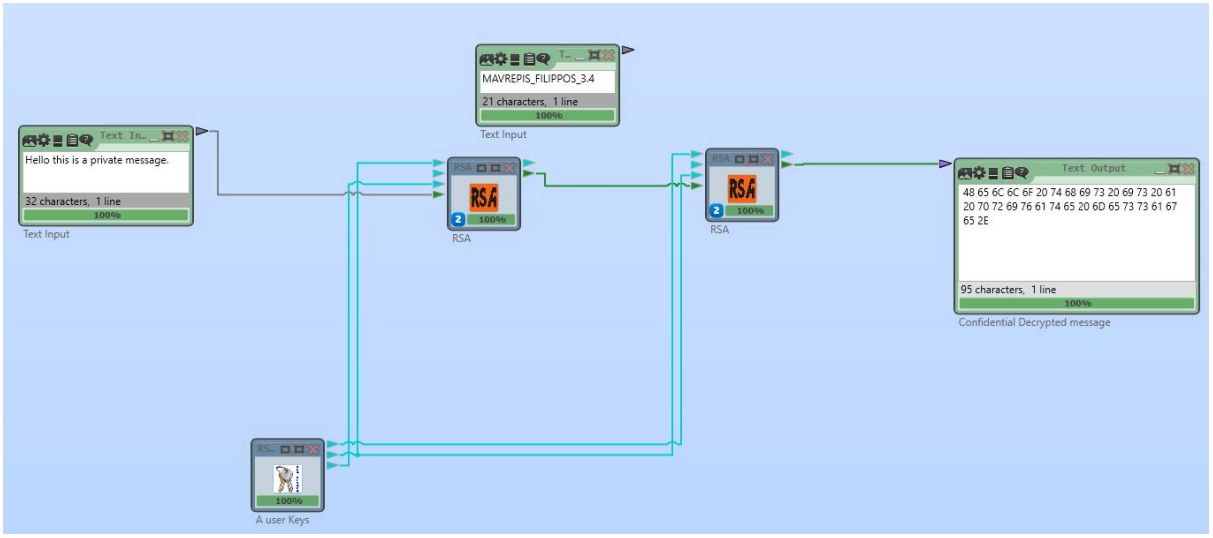
Confidentiality

Μέσω της έννοιας της εμπιστευτικότητας εξασφαλίζουμε ότι το μήνυμα που αποστέλλεται μέσω ενός καναλιού επικοινωνίας θα μπορεί να το διαβάσει μόνο αυτός που στον οποίο απευθύνεται το μήνυμα. Δηλαδή ακόμα και αν το κανάλι επικοινωνίας έχει εκτεθεί, η πληροφορία που μεταφέρεται είναι άχρηστη. Για τον λόγο αυτό ο Α κρυπτογραφεί το μήνυμα που στέλνει χρησιμοποιώντας το δημόσιο κλειδί του χρήστη Β. Με αυτόν τον τρόπο μόνο ο Β μέσω του μυστικού ιδιωτικού του κλειδιού μπορεί να το αποκρυπτογραφήσει.



Authentication

Στην έννοια της αυθεντικοποίησης ο χρήστης Α θέλει ο παραλήπτης να γνωρίζει ότι το μήνυμα προήλθε από εκείνον. Προκειμένου να γίνει αυτό, θα χρησιμοποιήσει μια πληροφορία που μόνο αυτός κατέχει, το ιδιωτικό του κλειδί. Με αυτό κρυπτογραφεί το μήνυμα ώστε ο Β να μπορεί να το αποκρυπτογραφήσει μόνο με το δημόσιο κλειδί του Α. Έτσι είναι σίγουρος ότι το μήνυμα προέρχεται από αυτόν. Σημαντική παρατήρηση αποτελεί πως με αυτόν τον τρόπο δεν διασφαλίζεται η ιδιωτικότητα του μηνύματος καθώς το δημόσιο κλειδί του Α που χρησιμοποιείται για την αποκρυπτογράφηση του μηνύματος αποτελεί πληροφορία γνωστή στον οποιονδήποτε.



SNORT

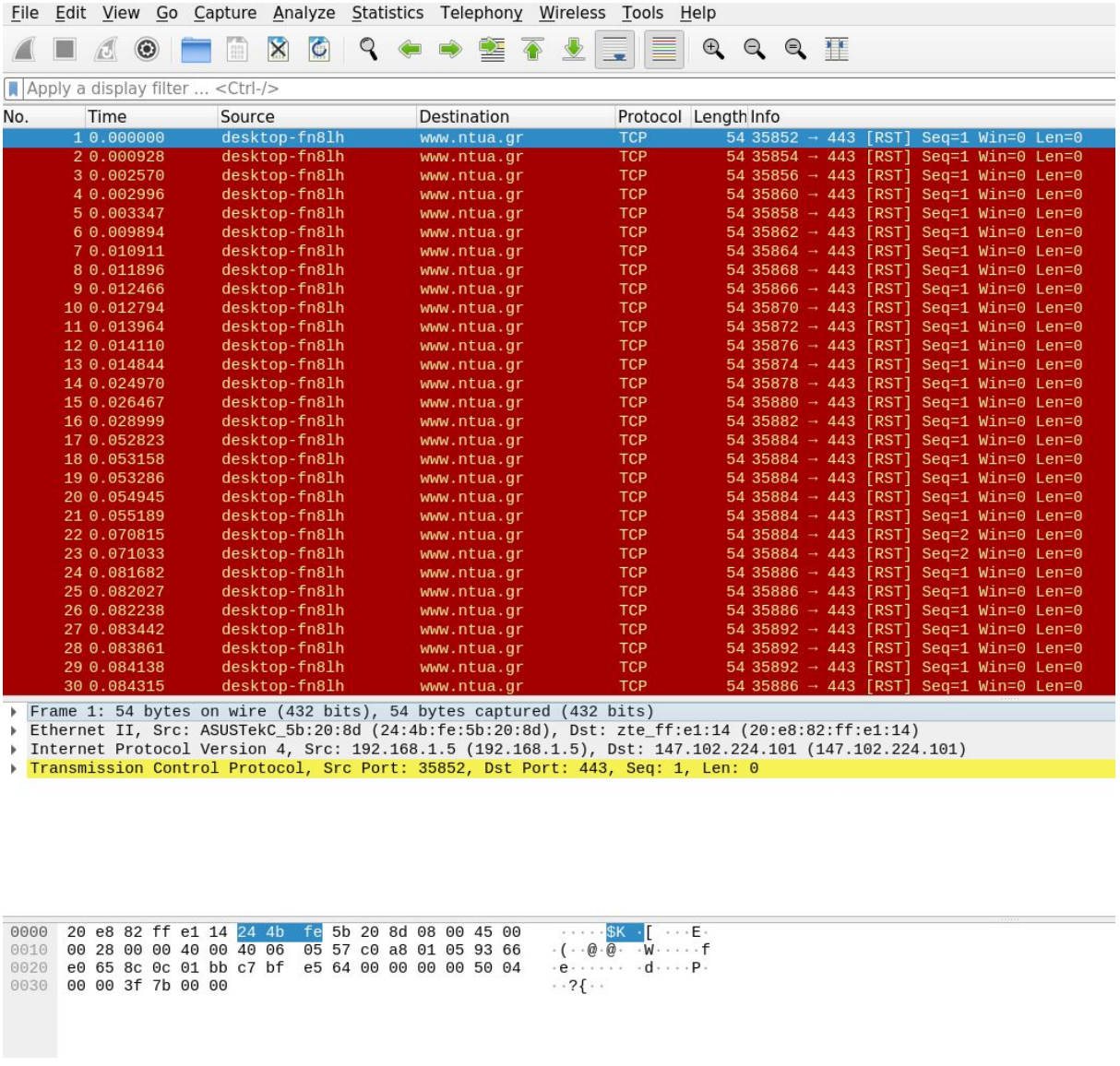
Η διεπαφή στην οποία υπάρχει κίνηση είναι η enp4s0 επομένως η εντολή καταγραφής είναι snort -v -i enp4s0. Εκτελώντας ping προς έναν οποιοδήποτε στόχο λαμβάνουμε τα εξής:

```
===============================================================================
Run time for packet processing was 20.483724 seconds
Snort processed 62 packets.
Snort ran for 0 days 0 hours 0 minutes 20 seconds
   Pkts/sec:           3
===============================================================================
Memory usage summary:
  Total non-mmapped bytes (arena):       786432
  Bytes in mapped regions (hblkhd):      22941696
  Total allocated space (uordblks):      684176
  Total free space (fordblks):           102256
  Topmost releasable block (keepcost):   100560
===============================================================================
Packet I/O Totals:
   Received:           65
   Analyzed:           62 ( 95.385%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            3 (  4.615%)
   Injected:            0
```

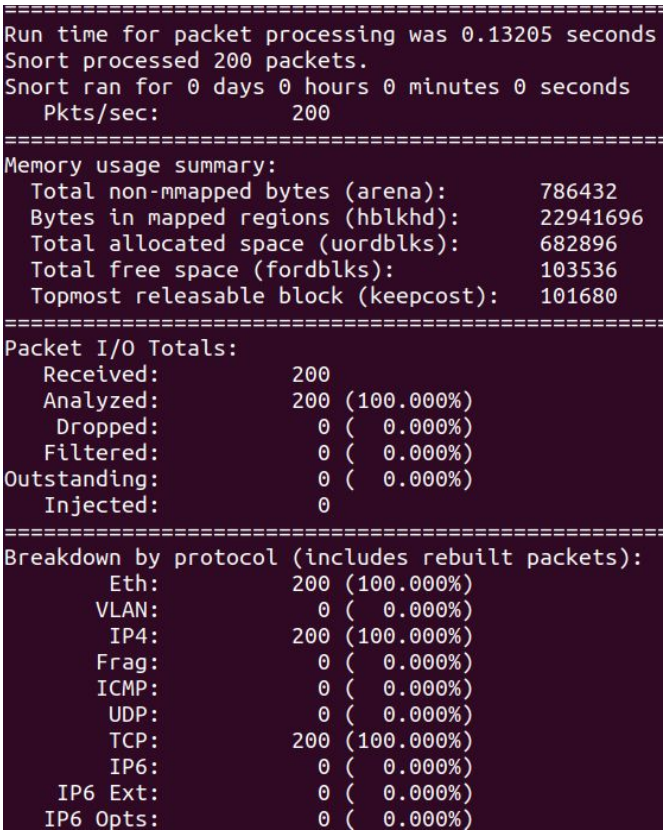
```
Breakdown by protocol (includes rebuilt packets):
        Eth:           62 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:           48 ( 77.419%)
       Frag:            0 (  0.000%)
       ICMP:           28 ( 45.161%)
        UDP:            6 (  9.677%)
        TCP:           14 ( 22.581%)
        IP6:            0 (  0.000%)
    IP6 Ext:            0 (  0.000%)
   IP6 Opts:            0 (  0.000%)
      Frag6:            0 (  0.000%)
      ICMP6:            0 (  0.000%)
       UDP6:            0 (  0.000%)
       TCP6:            0 (  0.000%)
     Teredo:            0 (  0.000%)
    ICMP-IP:            0 (  0.000%)
    IP4/IP4:            0 (  0.000%)
    IP4/IP6:            0 (  0.000%)
    IP6/IP4:            0 (  0.000%)
    IP6/IP6:            0 (  0.000%)
        GRE:            0 (  0.000%)
    GRE Eth:            0 (  0.000%)
   GRE VLAN:            0 (  0.000%)
    GRE IP4:            0 (  0.000%)
    GRE IP6:            0 (  0.000%)
GRE IP6 Ext:            0 (  0.000%)
   GRE PPTP:            0 (  0.000%)
    GRE ARP:            0 (  0.000%)
    GRE IPX:            0 (  0.000%)
   GRE Loop:            0 (  0.000%)
       MPLS:            0 (  0.000%)
        ARP:           14 ( 22.581%)
        IPX:            0 (  0.000%)
   Eth Loop:            0 (  0.000%)
   Eth Disc:            0 (  0.000%)
   IP4 Disc:            0 (  0.000%)
   IP6 Disc:            0 (  0.000%)
   TCP Disc:            0 (  0.000%)
   UDP Disc:            0 (  0.000%)
  ICMP Disc:            0 (  0.000%)
All Discard:            0 (  0.000%)
      Other:            0 (  0.000%)
Bad Chk Sum:           10 ( 16.129%)
    Bad TTL:            0 (  0.000%)
     S5 G 1:            0 (  0.000%)
     S5 G 2:            0 (  0.000%)
```

Εκτελώντας την εντολή με την επιπλέον παράμετρο -l και ορίζοντας τον φάκελο για την αποθήκευση της καταγραφής δημιουργούμε ένα log file. Ύστερα επισκεπτόμαστε το www.ntua.gr. Το αντίστοιχο screenshot της επίσκεψης αυτής φαίνεται παρακάτω στο Wireshark.

Επίσκεψη στο ntua.gr



κατ αντιστοιχία φαίνεται το ίδιο log file μέσω του snort (με την παράμετρο -r).

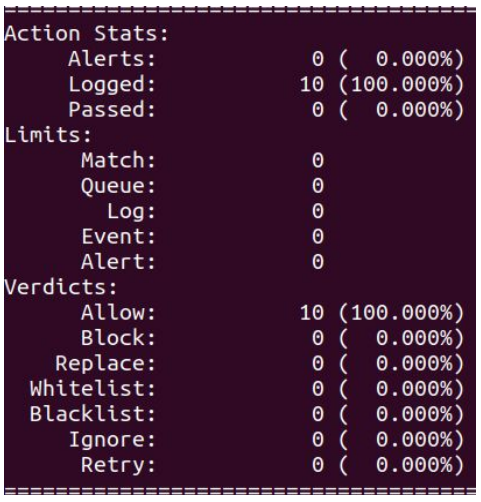


Ο κανόνας που συντάσσεται είναι ο εξής:

log TCP 192.168.1.0/24 any -> 147.102.222.210 [80,443]

Καταγραφή των πακέτων με πρωτόκολλο tcp από όλο το δίκτυο 192.168.1.0-192.168.1.255 σε οποιαδήποτε πόρτα πρός το ntua.gr (147.102.222.210) στις πόρτες 80 (http) και 443 (https).

Εκτελούμε playback στην καταγραφή για το ntua.gr με τον κανόνα ενεργό
sudo snort -r snort_logs/ntua_log_with_rule.1611328657 -c snort_rules/snort_rule.txt
Προκύπτει:



4.4

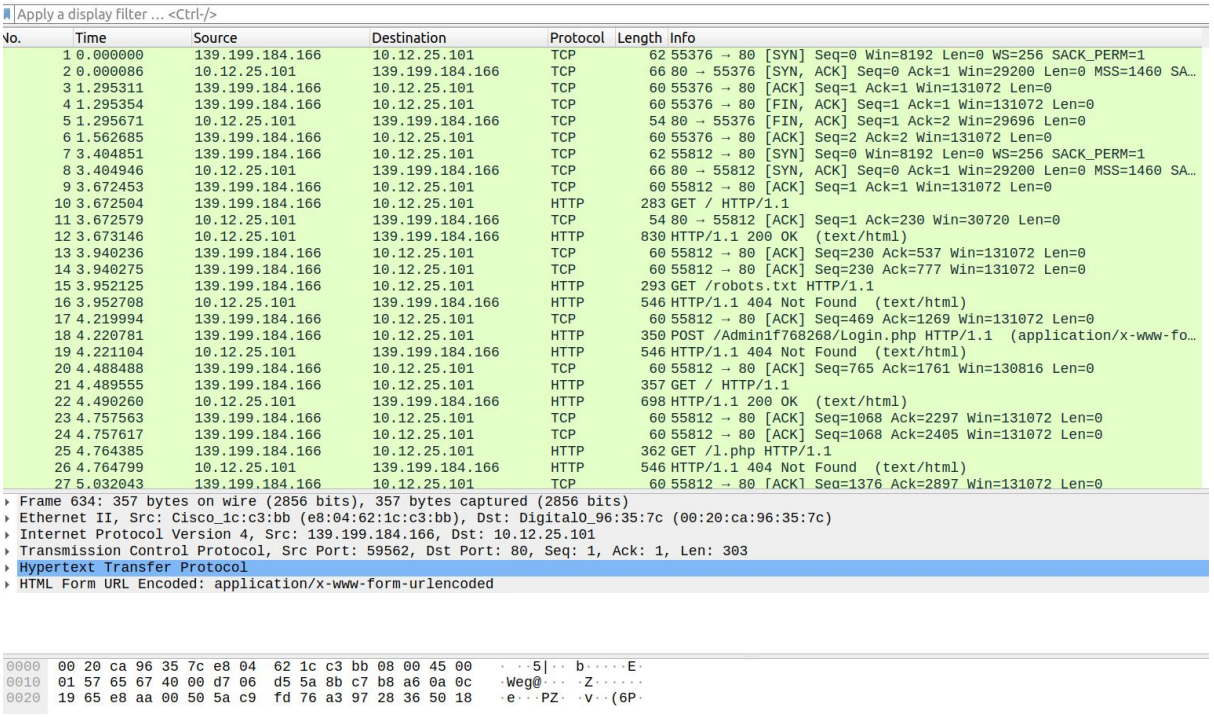
Για την απομόνωση των συγκεκριμένων κινήσεων τύπου http,SMTP, ftp και telnet απαιτείται να γνωρίζουμε τις πόρτες στις οποίες δρουν καθώς και τα πρωτόκολλα επικοινωνίας (tcp, udp). Μέσω του παρακάτω List of TCP and UDP port numbers προκύπτουν οι επόμενοι κανόνες (σε ξεχωριστά αρχεία ο καθένας).

FTP: log tcp any any <> any [20,21]
HTTP: log tcp any any <> any [80,443]
log udp any any <> any [80,443]
Telnet: log tcp any any <> any [23,443]
SMTP: log tcp any any <> any [25,465,587]

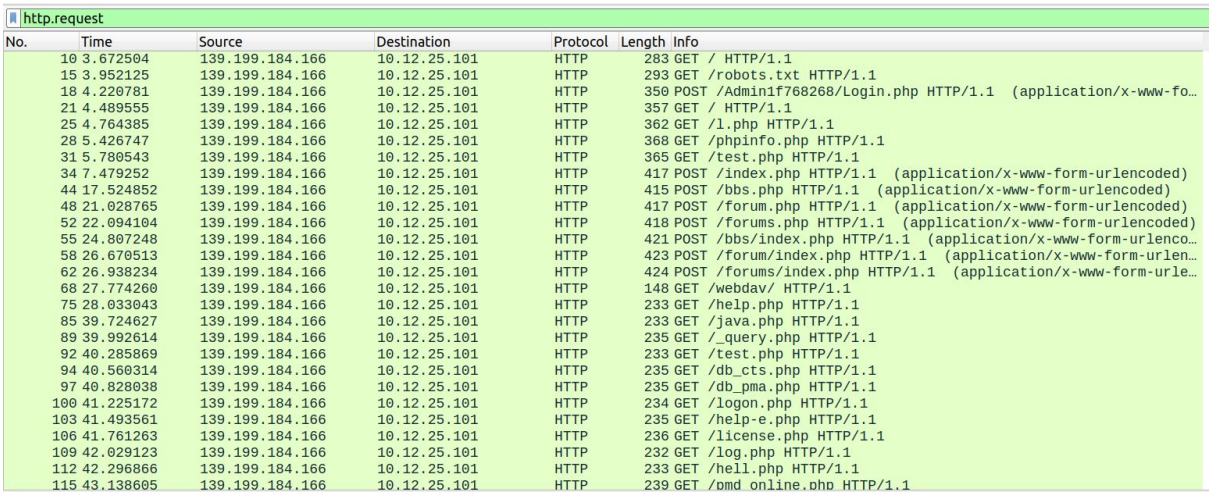
4.5

Το malware-traffic-analysis pcap file που χρησιμοποιήθηκε μπορεί να βρεθεί στον παρακάτω σύνδεσμο: Malware traffic

Χρησιμοποιώντας το wireshark πάνω σε ένα malware analysis pcap file μπορούμε να δούμε αρχικά τα εξης



Όλη η κίνηση του εν λόγω αρχείου είναι μεταξύ του επιτιθέμενου 139.199.184.166 και του webserver 10.12.25.101 (φυσικά αυτή είναι η sanitized IP). Εφόσον πρόκειται για ένα webserver ας δούμε αρχικά τα http requests που έχουν γίνει προς αυτόν μέσω του φίλτρου (http.request).



http.request

No.	Time	Source	Destination	Protocol	Length	Info
10	3.672504	139.199.184.166	10.12.25.101	HTTP	283	GET / HTTP/1.1
15	3.952125	139.199.184.166	10.12.25.101	HTTP	293	GET /robots.txt HTTP/1.1
18	4.220781	139.199.184.166	10.12.25.101	HTTP	350	POST /Admin1f768268/Login.php HTTP/1.1 (application/x-www-fo…
21	4.489555	139.199.184.166	10.12.25.101	HTTP	357	GET / HTTP/1.1
25	4.764385	139.199.184.166	10.12.25.101	HTTP	362	GET /l.php HTTP/1.1
28	5.426747	139.199.184.166	10.12.25.101	HTTP	368	GET /phpinfo.php HTTP/1.1
31	5.780543	139.199.184.166	10.12.25.101	HTTP	365	GET /test.php HTTP/1.1
34	7.479252	139.199.184.166	10.12.25.101	HTTP	417	POST /index.php HTTP/1.1 (application/x-www-form-urlencoded)
44	17.524852	139.199.184.166	10.12.25.101	HTTP	415	POST /bbs.php HTTP/1.1 (application/x-www-form-urlencoded)
48	21.028765	139.199.184.166	10.12.25.101	HTTP	417	POST /forum.php HTTP/1.1 (application/x-www-form-urlencoded)
52	22.094104	139.199.184.166	10.12.25.101	HTTP	418	POST /forums.php HTTP/1.1 (application/x-www-form-urlencoded)
55	24.807248	139.199.184.166	10.12.25.101	HTTP	421	POST /bbs/index.php HTTP/1.1 (application/x-www-form-urlenco…
58	26.670513	139.199.184.166	10.12.25.101	HTTP	423	POST /forum/index.php HTTP/1.1 (application/x-www-form-urlen…
62	26.938234	139.199.184.166	10.12.25.101	HTTP	424	POST /forums/index.php HTTP/1.1 (application/x-www-form-urle…
68	27.774260	139.199.184.166	10.12.25.101	HTTP	148	GET /webdav/ HTTP/1.1
75	28.033043	139.199.184.166	10.12.25.101	HTTP	233	GET /help.php HTTP/1.1
85	39.724627	139.199.184.166	10.12.25.101	HTTP	233	GET /java.php HTTP/1.1
89	39.992614	139.199.184.166	10.12.25.101	HTTP	235	GET /_query.php HTTP/1.1
92	40.285869	139.199.184.166	10.12.25.101	HTTP	233	GET /test.php HTTP/1.1
94	40.560314	139.199.184.166	10.12.25.101	HTTP	235	GET /db_cts.php HTTP/1.1
97	40.828038	139.199.184.166	10.12.25.101	HTTP	235	GET /db_pma.php HTTP/1.1
100	41.225172	139.199.184.166	10.12.25.101	HTTP	234	GET /logon.php HTTP/1.1
103	41.493561	139.199.184.166	10.12.25.101	HTTP	235	GET /help-e.php HTTP/1.1
106	41.761263	139.199.184.166	10.12.25.101	HTTP	236	GET /license.php HTTP/1.1
109	42.029123	139.199.184.166	10.12.25.101	HTTP	232	GET /log.php HTTP/1.1
112	42.296866	139.199.184.166	10.12.25.101	HTTP	233	GET /hell.php HTTP/1.1
115	43.138605	139.199.184.166	10.12.25.101	HTTP	239	GET /pmd_online.php HTTP/1.1

Είναι εμφανές ότι πρόκειται για κακόβουλο χρήστη καθώς από τα πρώτα κιόλας requests που κάνει προσπαθεί να αποκτήσει πρόσβαση σε πόρους που ένας κανονικός χρήστης δεν θα ενδιαφερόταν όπως το robots.txt, /admin, phpinfo.php κλπ. Ο επιτιθέμενος προσπαθεί να κάνει κάποια απαρίθμηση στόχων (target enumeration) προκειμένου να βρεί με ποιον τρόπο θα προσβάλλει τον web server.

Ακολουθώντας ένα HTTP stream του πρώτου πακέτου μπορούμε να δούμε τα εξής:

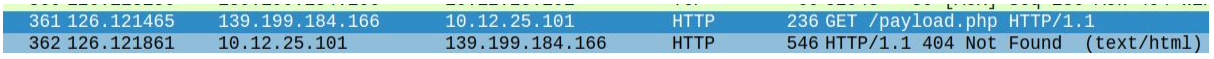
```
GET / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:31.0) Gecko/20100101 Firefox/31.0
Host: 128.199.64.235
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Wed, 25 Dec 2019 06:28:52 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Wed, 04 Dec 2019 22:03:56 GMT
ETag: "1d4-598e7fd31c852"
Accept-Ranges: bytes
Content-Length: 468
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

Ο επιτιθέμενος χρησιμοποιεί Windows NT 6.1 που αντιστοιχεί στα Windows 7 ενώ ο web server είναι είναι ένα μηχάνημα που τρέχει Apache/2.4.29 (Ubuntu).

Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι προσπάθειες να αποκτήσει πρόσβαση σε κάποιο command line prompt (πιθανώς για την εγκαθιδρυση ενος reverse shell).

Όπως φαίνεται δεν τα κατάφερε καθώς προσπαθεί χωρίς επιτυχία να εκτελέσει το payload.php και ο server επιστρέφει 404 not found επομένως δεν κατάφερε ποτέ να το ανεβάσει στον server.



361	126.121465	139.199.184.166	10.12.25.101	HTTP	236	GET /payload.php HTTP/1.1
362	126.121861	10.12.25.101	139.199.184.166	HTTP	546	HTTP/1.1 404 Not Found (text/html)

Ελέγχοντας την TCP κίνηση σε πόρτες πέραν της 80 λαμβάνουμε τα εξής:

!(tcp.port eq 80)

No.	Time	Source	Destination	Protocol	Length	Info
2423	1122.719783	139.199.184.166	10.12.25.101	TCP	62	52711 → 8080 [SYN] Seq=0 Win=8192 Len=0 WS=256 SACK_PERM=1
2424	1122.719840	10.12.25.101	139.199.184.166	TCP	54	8080 → 52711 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2425	1123.547186	139.199.184.166	10.12.25.101	TCP	62	[TCP Retransmission] 52711 → 8080 [SYN] Seq=0 Win=8192 Len=0 …
2426	1123.547254	10.12.25.101	139.199.184.166	TCP	54	8080 → 52711 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2427	1124.344022	139.199.184.166	10.12.25.101	TCP	60	[TCP Retransmission] 52711 → 8080 [SYN] Seq=0 Win=8192 Len=0 …
2428	1124.344086	10.12.25.101	139.199.184.166	TCP	54	8080 → 52711 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2429	1125.719627	139.199.184.166	10.12.25.101	TCP	62	53268 → 8983 [SYN] Seq=0 Win=8192 Len=0 WS=256 SACK_PERM=1
2430	1125.719695	10.12.25.101	139.199.184.166	TCP	54	8983 → 53268 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2431	1126.547681	139.199.184.166	10.12.25.101	TCP	62	[TCP Retransmission] 53268 → 8983 [SYN] Seq=0 Win=8192 Len=0 …
2432	1126.547759	10.12.25.101	139.199.184.166	TCP	54	8983 → 53268 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2433	1127.344536	139.199.184.166	10.12.25.101	TCP	60	[TCP Retransmission] 53268 → 8983 [SYN] Seq=0 Win=8192 Len=0 …
2434	1127.344605	10.12.25.101	139.199.184.166	TCP	54	8983 → 53268 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Βλέπουμε ότι γίνονται προσπάθειες για την πρόσβαση σε πόρτες όπως η 8080, 52711,53268,8983 επομένως ένας καλός κανόνας για το snort θα ήταν

alert tcp any any -> 10.12.25.101 [8080,52711,53268,8983] (msg:“Malware detected by PHILIPPOS MAVREPIS!!!” sid:1000001)

Εκτελώντας την παρακάτω εντολή για το αρχείο που εξετάστηκε αναμένουμε να προκύψει alert για τα 6 πακέτα που στόχευαν πόρτες εκτός της 80.

```
(base) philip@Desktop-FN8LH:~$ sudo snort -dev -r '/home/philip/Downloads/2019-12-25-traffic-analysis-exercise.pcap' -c snort_rules/malware_analysis.txt
```

```
===========================================
Action Stats:
     Alerts:             6 (  0.246%)
     Logged:             6 (  0.246%)
     Passed:             0 (  0.000%)
Limits:
      Match:             0
      Queue:             0
        Log:             0
      Event:             0
      Alert:             0
Verdicts:
      Allow:          2438 (100.000%)
      Block:             0 (  0.000%)
    Replace:             0 (  0.000%)
  Whitelist:             0 (  0.000%)
  Blacklist:             0 (  0.000%)
     Ignore:             0 (  0.000%)
      Retry:             0 (  0.000%)
===========================================
Snort exiting
```

```
[**] [1:1000001:0] “Malware detected by PHILIPPOS MAVREPIS!!!” [**]
[Priority: 0]
12/25-08:47:32.874300 E8:04:62:2D:C3:BB -> 00:20:CA:96:35:7C type:0x800 len:0x3C
139.199.184.166:52711 -> 10.12.25.101:8080 TCP TTL:215 TOS:0x0 ID:7618 IpLen:20 DgmLen:44 DF
******S* Seq: 0x37AFC093  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (3) => NOP NOP SackOK

[**] [1:1000001:0] “Malware detected by PHILIPPOS MAVREPIS!!!” [**]
[Priority: 0]
12/25-08:47:34.249905 E8:04:62:2D:C3:BB -> 00:20:CA:96:35:7C type:0x800 len:0x3E
139.199.184.166:53268 -> 10.12.25.101:8983 TCP TTL:215 TOS:0x0 ID:9996 IpLen:20 DgmLen:48 DF
******S* Seq: 0xA7758446  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (5) => NOP WS: 8 NOP NOP SackOK

[**] [1:1000001:0] “Malware detected by PHILIPPOS MAVREPIS!!!” [**]
[Priority: 0]
12/25-08:47:35.077959 E8:04:62:2D:C3:BB -> 00:20:CA:96:35:7C type:0x800 len:0x3E
139.199.184.166:53268 -> 10.12.25.101:8983 TCP TTL:215 TOS:0x0 ID:11477 IpLen:20 DgmLen:48 DF
******S* Seq: 0xA7758446  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (5) => NOP WS: 8 NOP NOP SackOK
```

Ένας πιο γενικός (strict) κανόνας με την υπόθεση ότι θέλουμε όλα τα πακέτα να πηγαίνουν στην πόρτα 80 θα ήταν:

alert tcp any any -> 10.12.25.101 !80 (msg:“Malware detected by PHILIPPOS MAVREPIS!!!” sid:1000002)