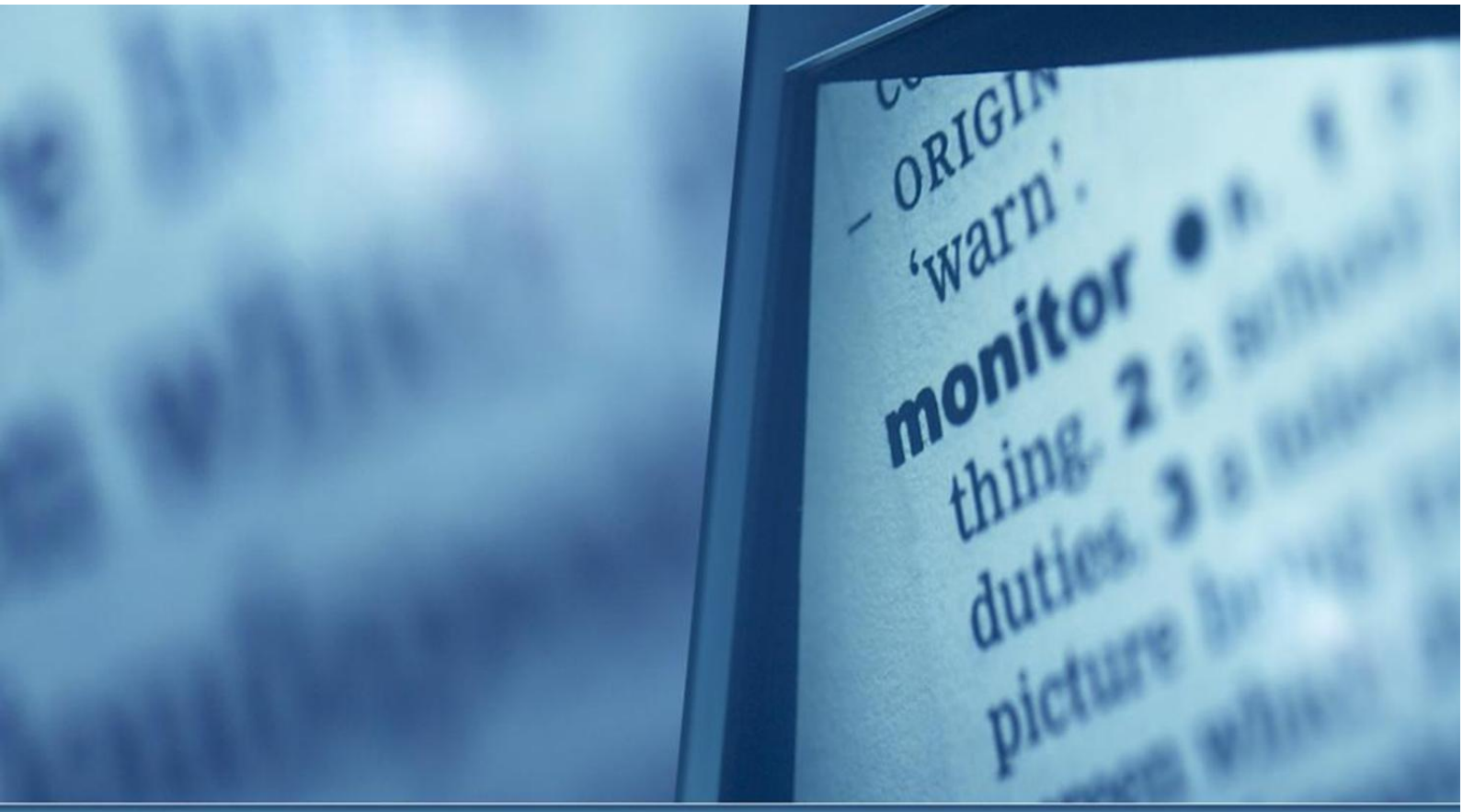


Snort IDS

4. ΤΟ ΛΟΓΙΣΜΙΚΟ SNORT

4.1 ΓΕΝΙΚΑ

Το Snort είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα (Open Source), αποτροπής εισβολών (network intrusion prevention system -NIPS)και ανίχνευσης εισβολών (network intrusion detection system -NIDS) σε δίκτυα. Μέχρι την έκδοση 1.9 χρησιμοποιούνταν σε δίκτυα μικρού μεγέθους και με μικρό εύρος ζώνης(bandwidth), δηλαδή μέχρι της τάξης των 100Mbps. Από την έκδοση 2.0 και έπειτα άλλαξε ο μηχανισμός εντοπισμού (detection engine) με την νέα «Hi-performance Multi-rule inspection engine» τεχνική και έτσι το Snort μπορεί να χρησιμοποιείται σε δίκτυα με εύρος ζώνης της τάξης Gigabit.

Δημιουργήθηκε από τον Marty Roesch το 1998 και σήμερα έχει περάσει στην εταιρία Sourcefire, όπου ο Roesch είναι ο ιδρυτής και προϊστάμενος στο τμήμα της τεχνολογικής ανάπτυξης της εταιρείας. Το 2009 αναρτήθηκε στο InfoWorld's Open Source Hall of Fame ως μία από τις «μεγαλύτερες σε τμήματα λογισμικού ανοιχτού πηγαίου κώδικα όλων των εποχών». Ο κώδικας είναι γραμμένος στην γλώσσα προγραμματισμού C και τρέχει σε όλα σχεδόν τα λειτουργικά συστήματα υπολογιστών (Cross-platform).

4.2 ΛΟΓΟΙ ΕΠΙΛΟΓΗΣ ΤΟΥ SNORT

Υπάρχει πληθώρα πακέτων λογισμικών συστημάτων ανίχνευσης ή αποτροπής εισβολών σε δίκτυα. Ο λόγος που επιλέχτηκε το Snort είναι για την ικανότητά του να εκτελεί σε πραγματικό χρόνο ανάλυση της κίνησης και την καταγραφή πακέτων σε Internet protocol (IP) δίκτυα. Ικανότητα ανίχνευσης μεγάλου εύρους επιθέσεων όπως buffer overflows, σάρωση θυρών (port scans), επιθέσεις που εκμεταλλεύονται σφάλματα των λειτουργικών συστημάτων, αδυναμίες των CGI κ.α. Δεν είναι απαιτητικό σε πόρους για την λειτουργία του. Επίσης είναι εύκολα διαμορφώσιμο και ευέλικτο ανάλογα με τις εκάστοτε ανάγκες του δικτύου, διότι όλα τα αρχεία διαμόρφωσης αλλά και των κανόνων που χρειάζονται για την ρύθμιση των παραμέτρων είναι στη διαθεσιμότητα του χρήστη/διαχειριστή του δικτύου.

Η εταιρεία προβαίνει σε συχνές αναβαθμίσεις του λογισμικού, έτσι μπορεί να ενημερώνεται για διορθώσεις ή προσθήκες χαρακτηριστικών όπως και μέσω της ιστοσελίδας του για νέες υπογραφές επιθέσεων. Δίνει την δυνατότητα στον χρήστη να δημιουργεί τους δικούς του κανόνες(rules) και να αλλάζει την βάση μέσα από την λειτουργία plug-ins, δηλαδή κώδικας που προαιρετικά εμπεριέχεται κατά την εγκατάσταση του λογισμικού και προσφέρει δυνατότητες όπως η ενεργός ανταπόκριση σε κακόβουλη κίνηση (malicious traffic). Τέλος, είναι φιλικό στην χρήση του σε σύγκριση με άλλα ανοικτού κώδικα λογισμικά, και λόγω της δωρεάν διανομής του υπό την άδεια της GNU GPL, είναι αρκετά διαδεδομένο στο χώρο με αποτέλεσμα την ύπαρξη ικανοποιητικού υλικού τεκμηρίωσης για την εγκατάσταση καθώς και για την λειτουργία του.

4.3 ΤΡΟΠΟΙ ΛΕΙΤΟΥΡΓΙΑΣ

Το Snort εκτός από την λειτουργία του σαν ανιχνευτής εισβολών μπορεί να ρυθμιστεί και σε άλλες λειτουργίες που περιγράφονται παρακάτω:

- Sniffer mode (Αναλυτής κίνησης δικτύου), το πρόγραμμα διαβάζει τα πακέτα του δικτύου και τα εμφανίζει στην κονσόλα (οθόνη) σε φιλική μορφή προς τον χρήστη, δηλαδή εκτελεί μία απλή καταγραφή κίνησης του δικτύου. Με διάφορα φίλτρα (Berkley packet filter-BPF) που μπορεί να χρησιμοποιηθούν από τον χρήστη, δίνεται η δυνατότητα να οριστεί το είδος των πακέτων που θα εμφανίζονται ως προς το πρωτόκολλο, τον αποστολέα, τον παραλήπτη και διάφορα άλλα χαρακτηριστικά ενός πακέτου. Δηλαδή, αν ο χρήστης πληκτρολογήσει την λέξη κλειδί icmp στο snort τότε θα εμφανίζονται μόνο τα πακέτα αυτού του πρωτόκολλου.
- Packet logger mode (Καταγραφικό πακέτων), το πρόγραμμα καταγράφει τα πακέτα που διαβάζει από το δίκτυο στο δίσκο, αντί να τα εμφανίζει απλά στην οθόνη. Αυτή η λειτουργία βοηθά σε περιπτώσεις όπου απαιτείται η λεπτομερής εξέταση των πακέτων που αναγιγνώσκονται. Το Snort μπορεί ν' αποθηκεύει τα πακέτα σε διάφορες μορφές, όπως για παράδειγμα σε binary μορφή (tcpdump format) με την οποία μπορούν να χρησιμοποιηθούν σαν είσοδο σε διάφορα άλλα προγράμματα ανάλυσης πακέτων και πρωτοκόλλων, σε ASCII μορφή ώστε να είναι δυνατή η ανάγνωσή τους, σε XML μορφή ή και να οργανωθούν σε βάσεις δεδομένων. Η συγκεκριμένη λειτουργία, δεν λειτουργεί συνήθως ανεξάρτητα από τις λειτουργίες του sniffer ή την NIDS, αλλά παράλληλα με αυτά.
- Network Intrusion Detection System – NIDS mode (ανίχνευση εισβολής σε δίκτυο), συγκρίνει την κίνηση του δικτύου μ' ένα προκαθορισμένο σύνολο υπογραφών που είναι γνωστές ως κανόνες, όπου ορίζονται από τον χρήστη και εκτελεί διάφορες ενέργειες με βάση ότι έχει εντοπίσει. Είναι η πιο κοινή λειτουργία (common mode) που τρέχει το Snort και χρειάζεται και στο εργαστηριακό κομμάτι της εργασίας αυτής. Συνήθως εκτελείται από

την γραμμή εντολών (command line) σε κάθε λειτουργικό σύστημα (Unix ή Windows). Υπάρχουν βέβαια λογισμικά που προσφέρουν γραφικό περιβάλλον, όπως είναι το IDScenter στα windows και το Demarc Puresecure για windows και για Unix. Τα λογισμικά αυτά όμως δεν θα μπορέσουν να χρησιμοποιηθούν, διότι είναι συμβατά για παλαιότερες εκδόσεις των λειτουργικών συστημάτων. Η τεχνική που χρησιμοποιεί το Snort για την διαδικασία ανίχνευσης εισβολών είναι κατά κύριο λόγο η μέθοδος ανίχνευσης κακής συμπεριφοράς (Misuse Detection) με την χρήση των υπογραφών(Signatures) ενός βλαβερού(malicious) πακέτου. Το snort όμως ειδικά μετά την έκδοση 2.0 συνδυάζει στην λειτουργία της ανάλυσης των γεγονότων για την ανίχνευση πιθανών επιθέσεων και κάποιες από τις μεθόδους του πρωτόκολλου ανίχνευσης διαταραχών (Protocol Anomaly Detection) και του πρωτοκόλλου κακής συμπεριφοράς (Misuse Detection). Οι μηχανισμοί αυτοί υλοποιούνται κατά κύριο λόγο από τους προεπεξεργαστές (preprocessors) που εξηγούνται αναλυτικά παρακάτω, αλλά και από το νέο μηχανισμό του snort 2.0 που συντάσσει τους κανόνες (rules) σε κατηγορίες.

4.4 ΚΑΝΟΝΕΣ (RULES) Ή ΥΠΟΓΡΑΦΕΣ (SIGNATURES) ΤΩΝ IDS

4.4.1 ΒΑΣΙΚΕΣ ΔΙΑΦΟΡΕΣ

Οι κανόνες (rules) και οι υπογραφές (signatures) είναι ισοδύναμα σαν έννοιες και συνήθως χρησιμοποιούνται σαν συνώνυμες λέξεις. Υπάρχουν όμως κάποιες διαφορές[16] μεταξύ τους, που είναι οι εξής:

- Υπογραφές (signatures), είναι τα ειδικά χαρακτηρίστηκα του πακέτου που το χαρακτηρίζουν σαν ύποπτο ή βλαβερό(malicious). Τα χαρακτηριστικά αυτά είναι στο payload ή την επικεφαλίδα (header) του πακέτου και είναι

[16]Δημήτρης Πρίτσος, ISLAB HACK: Βασικές Έννοιες & Προγραμματισμός του Snort 2.0, Αθήνα 2003 σελίδα 5.

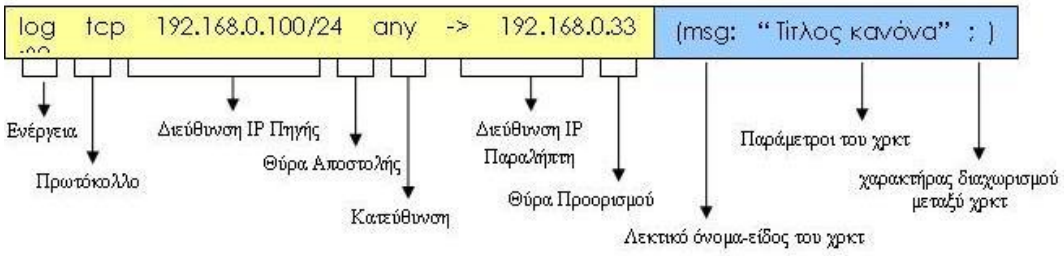
πρότυπα (patterns) από συμβολοσειρές (strings) που χαρακτηρίζονται σαν υπογραφή (Signature) ενός «κακού» πακέτου. Γενικά η περιγραφή ενός πακέτου που είναι malicious όταν γίνεται με ένα signature είναι στατική. Δηλαδή, ένα signature περιγράφει ένα υπαρκτό χαρακτηριστικό (positive pattern much) στο payload και μερικά χαρακτηριστικά στην επικεφαλίδα (header)του πακέτου.

- Κανόνες (rules), είναι κανόνες οι οποίοι περιγράφουν στο snort ή άλλο IDS τα χαρακτηριστικά ενός πακέτου που μπορεί να είναι μέρος μίας γνωστής επίθεσης καθώς και την ενέργεια που θα εκτελεστεί κατά τον εντοπισμό του. Η περιγραφή ενός πακέτου με ένα rule είναι αρκετά πιο δυναμική. Αφενός σε ένα rule μπορεί να περιγράφονται περισσότερα του ενός υπαρκτά χαρακτηριστικά στο payload αφετέρου μπορούν να περιγράφονται χαρακτηριστικά που δεν πρέπει να έχει ένα πακέτο για να θεωρηθεί ύποπτο. Τέλος ένα rule μπορεί να περιγράφει ένα ολόκληρο stream και όχι ένα πακέτο μόνο για τις περιπτώσεις που γίνεται statefull intrusion detection.

4.5 ΟΙ ΚΑΝΟΝΕΣ (RULES)

Οι κανόνες είναι ένα από το σημαντικότερο τμήμα των συστημάτων ανίχνευσης εισβολών. Όπως αναφέραμε και στον προσδιορισμό της έννοιας παραπάνω, πρόκειται για ένα πρότυπο με το οποίο γίνεται αναζήτηση στα διακινούμενα πακέτα του δικτύου. Με την εύρεση κάποιου πακέτου που φέρει χαρακτηριστικά ίδια με αυτά του πρότυπου, θεωρείται από το λογισμικό ως επίθεση. Έτσι μπορεί να ανιχνεύει μία προσπάθεια σύνδεσης από μία IP «ύποπτη», τον αντικανονικό συνδυασμό των πακέτων TCP κάνοντας ένα έλεγχο της διεύθυνσης της πηγής. Επίσης μπορεί να ανιχνεύει, μία προσπάθεια επίθεσης με Denial of Service χρησιμοποιώντας την μέθοδο της πολλαπλής αποστολής ίδιας εντολής, κάτι που αντιμετωπίζεται με τον έλεγχο του αριθμού που μία εντολή εκτελείται και να παράγεται ειδοποίηση όταν ξεπεραστεί το όριο που έχει οριστεί. Ακόμα μπορεί να ανιχνεύει μία επίθεση σε ένα FTP server,

δημιουργώντας μία υπογραφή που θα στηρίζεται σε διαδοχή καταστάσεων (stage tracking), ειδοποιώντας όταν κάποιος προσπαθήσει να κάνει κάποια κίνηση χωρίς να έχει περάσει από την απαιτούμενη διαδικασία. Δημιουργώντας τον κατάλληλο κανόνα μπορεί να ανιχνεύει ένα email με ιό, απλώς ελέγχοντας το όνομα του θέματος ή των συνημμένων αρχείων. Τέλος παρατηρείται από τα παραπάνω παραδείγματα υπογραφών και τους τρόπους που αυτές λειτουργούν για την ανίχνευση των επιθέσεων, πως μπορούν να είναι από αρκετά απλές, ελέγχοντας κάποιο πεδίο των πακέτων, ως σύνθετες όπου αναλύουν την σύνδεση με βάση το χρησιμοποιούμενο πρωτόκολλο. Σύμφωνα με την εικόνα που εμφανίζεται, θα κάνουμε μία σύντομη ανάλυση ενός κανόνα (rule).



Εικόνα 8 Κανόνας

Οι Κανόνες(rules) του Snort μπορούν να γραφτούν σε απλή περιγραφική γλώσσα σε ASCII μορφή και κάθε ένα από αυτά αποτελείται από δύο λογικά μέρη, τον Rule Header (επικεφαλίδα κανόνα) και τα Rule Options (ιδιότητες/χαρακτηριστικά κανόνα). Ο Rule Header περιέχει τις εξής πληροφορίες:

- **Action**: Είναι η ενέργεια που θα εκτελέσει το Snort όταν ταιριάξει κάποιο πακέτο με ένα *Rule*. Η ενέργεια αυτή έχει να κάνει με την αντίδραση (Response) του Snort κατά την ανίχνευση μίας πιθανής επίθεσης. Η ενέργεια (action) μπορεί να είναι :
    - ***Alert***, η οποία θα δημιουργήσει μία ειδοποίηση για το γεγονός που εντόπισε και στη συνέχεια θα καταγράψει το πακέτο. Οι ειδοποιήσεις είναι ο τρόπος με τον οποίο το Snort επισημαίνει το γεγονός της ανίχνευσης μίας επίθεσης.

- **Log**, η οποία θα καταγράψει το πακέτο στον δίσκο.
- ***Pass***, η οποία θα αγνοήσει το πακέτο.
- ***Activate***, η οποία θα προκαλέσει μία ειδοποίηση και στη συνέχεια θα ενεργοποιήσει ένα δυναμικό κανόνα(*dynamic Rule)*.
- ***Dynamic***, η οποία θα περιμένει μέχρι να ενεργοποιηθεί από ένα activate *Rule* και στη συνέχεια θα ενεργήσει σαν ένα *log Rule*. Ο χρήστης έχει την δυνατότητα να ορίσει και δικούς του τύπους από ενέργειες (actions).

- **Protocol**: Είναι το είδος του πρωτοκόλλου στο οποίο ανήκει το πακέτο που θα εξεταστεί. Το πρωτόκολλο μπορεί να είναι ip, tcp, icmp, udp.
- **Source IP**: Είναι η IP διεύθυνση αποστολέα που βρίσκεται στον IP header του πακέτου, σε συνδυασμό με την μάσκα του δικτύου (netmask) στο οποίο μπορεί να ανήκει, εκφρασμένη με CIDR τρόπο γραφής. Με τον CIDR τρόπο γραφής γίνεται δυνατό να ορισθεί μία ομάδα (block) από συνεχείς IP διευθύνσεις.
- **Source Port**: Είναι η πόρτα αποστολής που έχει νόημα στα tcp και udp πακέτα. Στο συγκεκριμένο παράδειγμα με το λεκτικό any εννοείται οποιαδήποτε πόρτα.
- **Destination IP**: Είναι η IP διεύθυνση του παραλήπτη του πακέτου. Ο τρόπος γραφής της είναι ο ίδιος με αυτόν που ισχύει για την Source IP και στο συγκεκριμένο παράδειγμα η IP διεύθυνση του παραλήπτη με την netmask που έχει ορισθεί, αντιπροσωπεύει έναν μόνο αριθμό τον 192.168.0.33.
- **Destination Port**: Είναι η πόρτα προορισμού του πακέτου.

Τα χαρακτηριστικά των κανόνων(Rule Options) περιέχουν πληροφορία που αναφέρεται στα χαρακτηριστικά για τα οποία θα ελεγχθεί το πακέτο. Επίσης στα Rule Options μπορούν να ορισθούν και κάποιες επιπρόσθετες ενέργειες που θα εκτελεστούν για κάποιο πακέτο που θα ταιριάξει με τους κανόνες. Το

κάθε option που περιέχεται στο Rule Options τμήμα του κανόνα, αποτελείται από τα Option Keyword και τα Option Arguments.

i. **Option Keyword**: Είναι το λεκτικό που υποδηλώνει το όνομα-είδος του option. Το λεκτικό από την έκδοση snort 2.0 και μετά έχει την δυνατότητα να εκφραστεί σαν Regular Expression του UNIX. Αυτό δίνει την δυνατότητα να περιγραφούν οι κανόνες με μία γενική μορφή ώστε να μπορούν να περιγράψουν γενικές επιπτώσεις επιθέσεων, όπως το Buffer Overflow. Για παράδειγμα, στην τελευταία περίπτωση, μπορεί να περιγράφει ένας κανόνας με τέτοια μορφή ώστε όταν συναντάτε ένας μεγάλος αριθμός από NOPs σε ένα πακέτο, αυτό να θεωρείται σαν ύποπτο για buffer overflow exploit. Έτσι, δίνεται η δυνατότητα να εκφραστεί, μέσα από ένα κανόνα, ένας τρόπος ανάλυσης βασισμένος στις ανωμαλίες (Anomaly based) ενός πακέτου ή stream. Χαρακτηριστική είναι η περίπτωση του Buffer Overflow, το οποίο παραμένει αρκετά δύσκολο στον εντοπισμό του με αυτό τον τρόπο, και είναι μία πολύ χρήσιμη εναλλακτική για να πιάνονται όλες οι παραλλαγές ενός είδη γνωστού Buffer Overflow Exploit.
ii. **Option Argument**: Είναι οι παράμετροι που δέχεται το option σε σχέση με τις οποίες θα ελεγχθεί το πακέτο. Τα Option Arguments για κάθε option, πρέπει να διαχωρίζονται με τον χαρακτήρα ':' από το αντίστοιχο Option Keyword.
iii. **Option Separator**: Είναι ο χαρακτήρας διαχωρισμού ';' μεταξύ δύο χαρακτηριστικών.

Ο συγκεκριμένος κανόνας στην **Εικόνα 8** έχει ένα χαρακτηριστικό. Αυτό είναι το όνομα msg, το οποίο δηλώνει ότι αν ταιριάξει κάποιο πακέτο με αυτό τον κανόνα, τότε μαζί με την ειδοποίηση (ή το πακέτο που θα καταγραφεί) θα τυπωθεί και κάποιο μήνυμα, το οποίο είναι αυτό που ακολουθεί μέσα στα εισαγωγικά και το οποίο αποτελεί το Option Argument αυτού του χαρακτηριστικού.

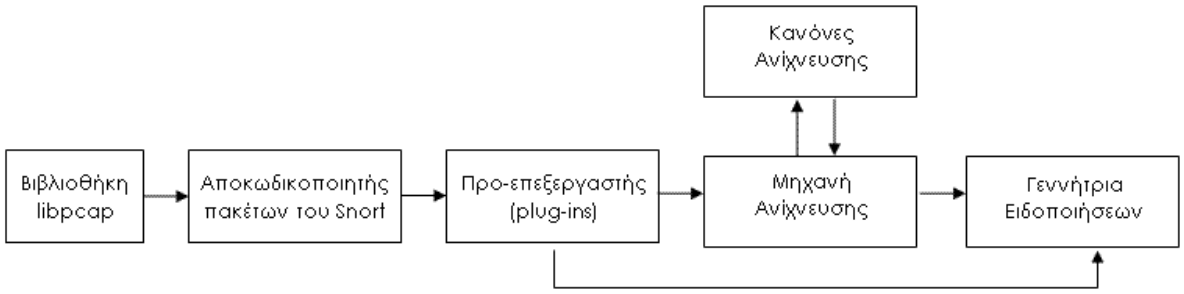
Το Snort διανέμεται με πάνω από 2500 έτοιμους κανόνες, για χρήση στην ανίχνευση γνωστών επιθέσεων, ενώ για την δημιουργία νέων κανόνων προσφέρει μία μεγάλη γκάμα από χαρακτηριστικά που μπορεί ο χρήστης να χρησιμοποιήσει, τα οποία του δίνουν την ευελιξία να εκτελεί λεπτομερής και σε βάθος περιγραφή των χαρακτηριστικών του κάθε πακέτου, για το οποίο θέλει να γίνει έλεγχος για τον εντοπισμό μίας επίθεσης. Τα Rules χωρίζονται σε δύο κατηγορίες στα **Generic Rules** και στα **Unique Rules**. Αυτά ορίζονται αυτομάτως από το snort αναλόγως με της πληροφορίες που φέρει το header του Rule.

- **Generic**, είναι οι κανόνες που στον ορισμό των IP δικτύων ή πόρτων έχουν την λέξη κλειδί "**any**". Αυτό σημαίνει ότι ο κανόνας είναι έτσι ορισμένος ώστε να ελέγχει οποιοδήποτε πακέτο που έχει υπογραφή στο payload του αυτό που περιγράφεται στα χαρακτηριστικά του κανόνα.
- **Unique Rules**, είναι αυτά που έχουν συγκεκριμένη έκταση δικτύων και πορτών που εξετάζουν.

4.6 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ SNORT

Στην ενότητα αυτή, θα μελετηθεί η λειτουργία του λογισμικού και ο τρόπος όπου παράγονται τα αποτελέσματά του. Το Snort αποτελείται από πέντε βασικά στάδια επεξεργασίας των δεδομένων που λαμβάνει και είναι:

- Η βιβλιοθήκη συλλογής πακέτων (Packet capture libraty-libpcap)
- Τον αποκωδικοποιητή πακέτων (Packet decoder)
- Τον προ-επεξεργαστή (Preprocessor)
- Την μηχανή ανίχνευσης (Detection Engine)
- Πακέτα λογισμικού αποτύπωσης εξόδου (Output Plug-ins)



Εικόνα 9 Στάδια επεξεργασίας ενός δικτυακού πακέτου που ελέγχει το Snort

4.7 ΒΙΒΛΙΟΘΗΚΗ ΣΥΛΛΟΓΗΣ ΠΑΚΕΤΩΝ (PACKET CAPTURE LIBRARY – LIBPCAP)

Το Snort για την συλλογή των πακέτων που διακινούνται στο δίκτυο που παρακολουθεί, μέσω της κάρτας δικτύου του υπολογιστή, χρησιμοποιεί την βιβλιοθήκη pcap (packet capture library). Αυτή είναι μία διασύνδεση προγραμματισμού εφαρμογών (Application Programming Interface – API), που εκτός από την συλλογή πακέτων από το δίκτυο, στις τελευταίες εκδόσεις, δίνεται η δυνατότητα να μεταδίδει και πακέτα στο επίπεδο συνδέσμου (link layer) του TCP/IP. Η συγκεκριμένη βιβλιοθήκη είναι σχεδιασμένη να επικοινωνεί με τις υπόλοιπες εφαρμογές στην γλώσσα προγραμματισμού C/C++.

Το γεγονός ότι η βιβλιοθήκη είναι ανεξάρτητο τμήμα του λογισμικού snort δίνει την δυνατότητα να χρησιμοποιείται σε διαφορετικά λειτουργικά , ενισχύοντας την ανεξαρτησία του snort. Το λειτουργικό σύστημα των Windows χρησιμοποιεί την βιβλιοθήκη pcap με την ονομασία Winpcap. Χαρακτηριστικό της βιβλιοθήκης αυτής είναι ότι συλλέγει τα πακέτα σε ακατέργαστη μορφή, δηλαδή όπως αυτά μεταφέρονται στο δίκτυο, μη επιτρέποντας στο εκάστοτε λειτουργικό σύστημα την αλλαγή σε αυτά. Το snort επωφελείται από αυτήν την ιδιότητα, αφού χρειάζονται όλες οι πληροφορίες που περιέχονται σ' ένα πακέτο για να ανιχνεύσει ορισμένες επιθέσεις.

4.8 ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗΣ ΠΑΚΕΤΩΝ (PACKET DECODING)

Η λειτουργία του είναι να λαμβάνει τα διαφορετικού τύπου πακέτα του δικτύου (Ethernet, SLIP, PPP κτλ) με την χρήση της βιβλιοθήκης libpcap. Έτσι κάθε πακέτο που φτάνει στο δίκτυο το πιάνει και το δίνει στην μηχανή αποκωδικοποίησης. Στην συνέχεια επιλέγεται τι είδους πακέτο είναι με βάση το δεύτερο επίπεδο του OSI (OSI-layer2) και από εκεί καλείται η κατάλληλη συνάρτηση που θα κάνει την πρώτη αποκωδικοποίηση του πακέτου. Για παράδειγμα, αν είναι ένα πακέτο Ethernet, θα κληθεί η κατάλληλη συνάρτηση για Ethernet decoding. Η διαδικασία αυτή συνεχίζεται για κάθε επίπεδο του OSI μέχρι το πακέτο να αποκωδικοποιηθεί πλήρως. Αυτό πετυχαίνεται εκμεταλλευόμενο την ιδιότητα της ενθυλάκωσης που χρησιμοποιούν τα πρωτόκολλα του διαδικτύου. Στην πραγματικότητα, ο αποκωδικοποιητής πακέτων είναι μία σειρά από αποκωδικοποιητές όπου, ο καθένας αποκωδικοποιεί συγκεκριμένα στοιχεία των πρωτοκόλλων. Τα αποτελέσματα της αποκωδικοποίησης για κάθε πακέτο δομούνται σε μία δομή την struct _Packet και στην συνέχεια προωθούνται στο επόμενο στάδιο επεξεργασίας.

4.9 ΠΡΟ-ΕΠΕΞΕΡΓΑΣΤΕΣ (PREPROCESSORS)

Οι προεπεξεργαστές είναι plug-ins του Snort που εκτελούνται μετά την αποκωδικοποίση πακέτων και πριν την μηχανή ανίχνευσης. Επιτρέπει στο λογισμικό να επεκτείνει την λειτουργικότητά του, δίνοντας την δυνατότητα σε χρήστες και προγραμματιστές να εισάγουν plug-ins αρκετά εύκολα. Μπορούν να χρησιμοποιηθούν είτε για να ελέγξουν τα πακέτα για ύποπτη δραστηριότητα είτε για να τα επεξεργαστούν έτσι, ώστε η μηχανή ανίχνευσης να μπορεί να τα αξιοποιήσει αποδοτικότερα. Οι προεπεξεργαστές καλούνται προς εκτέλεση μία μόνο φορά για κάθε πακέτο. Θα πρέπει να σημειωθεί ότι υπάρχουν είδη επιθέσεων τα οποία δεν θα μπορούσαν να ανιχνευθούν από το σύστημα του Snort χωρίς την επιπλέον επεξεργασία των προεπεξεργαστών, όπως για παράδειγμα ο frag2 που ανασυνθέτει τα κατακερματισμένα πακέτα σε ένα νέο

πακέτο ώστε να μπορούν να εφαρμόζονται οι κανόνες στο νέο πακέτο και όχι στο κάθε κομμάτι του «παλιού» πακέτου.

Οι προεπεξεργαστές είναι πολύ σημαντικό χαρακτηριστικό του IDS λόγω του ότι τα plug-ins μπορούν να ενεργοποιηθούν ή ν' απενεργοποιηθούν κατά βούληση του διαχειριστή του λογισμικού. Αν για παράδειγμα, δεν επιθυμείται ο έλεγχος του δικτύου για σαρώσεις θυρών, δίνεται η δυνατότητα απενεργοποίησης του εν λόγω plug-in χωρίς να επηρεαστεί το υπόλοιπο σύστημα επεξεργασίας. Οι παράμετροι που αφορούν στους προεπεξεργαστές βρίσκονται και μπορούν να διαμορφωθούν μέσω του αρχείου snort.conf ανάλογα με τις ανάγκες του δικτύου.

Η τελευταίες εκδόσεις του Snort 2.9 και έπειτα, χρησιμοποιούν τους κάτωθι προεπεξεργαστές:

- **Frag3**

  Έχει δημιουργηθεί για να αντικαταστήσει τον παλαιότερο frag2 προσφέροντας γρηγορότερη εκτέλεση, απλούστερη διαχείριση δεδομένων και αντιμετώπιση τεχνικών αποφυγής ανίχνευσης. Ο συγκεκριμένος προεπεξεργαστής έχει σαν στόχο τα πακέτα που περνούν κατακερματισμένα από το δίκτυο που παρακολουθείται από το snort, να τα επαναδομεί στην αρχική τους μορφή (ένα πακέτο). Με αυτό το μηχανισμό δύναται η δυνατότητα στο λογισμικό να δοκιμάζει τους κανόνες και να καταλαβαίνει αν γίνεται προσπάθεια επίθεσης.

- **Stream5**

  Έχει αντικαταστήσει τον Stream4 αλλά και τον Flow που υπήρχαν σε προηγούμενες εκδόσεις του snort. Ο προεπεξεργαστής Stream5 επιτρέπει την ανασυγκρότηση της ροής δεδομένων TCP καθώς και την ανάλυση με βάση την κατάσταση της ροής. Έχει την δυνατότητα να παρακολουθεί πολλές ταυτόχρονες ροές TCP πακέτων. Τέλος, έχει την δυνατότητα παρακολούθησης πακέτων UDP.

- **sfPortscan**

  Ο προεπεξεργαστής αυτός έχει αναπτυχθεί από την Sourcefire και έχει σχεδιαστεί για να ανιχνεύει την πρώτη φάση σε μία δικτυακή επίθεση, την φάση της αναγνώρισης. Στην φάση αυτή, ο επιτιθέμενος προσπαθεί να ανακαλύψει τι είδος δικτυακά πρωτόκολλα και υπηρεσίες υποστηρίζει ο διακομιστής του δικτύου. Μίας και ο επιτιθέμενος δεν έχει γνώση του στόχου του, τα περισσότερα ερωτήματα που στέλνει στον διακομιστή θα έχουν αρνητική απάντηση, αφού πολλές από τις υπηρεσίες δεν υπάρχουν. Λαμβάνοντας υπόψη πως πρόκειται για «νόμιμη» δικτυακή επικοινωνία, οι αρνητικές απαντήσεις από διακομιστές είναι σπάνιες και πόσο μάλλον πολλαπλές αρνητικές απαντήσεις σε ένα δεδομένο χρόνο. Με αυτό τον τρόπο ο sfportscan προσπαθεί να βρει αν συμβαίνει επίθεση με σάρωση θυρών του δικτύου.

Ένα από τα γνωστότερα εργαλεία σάρωσης θυρών που χρησιμοποιείται σήμερα είναι το Nmap, όπου θα χρησιμοποιηθεί και για τις ανάγκες του εργαστηριακού μέρους.

- **RPC Decode**

  Ο σκοπός του προεπεξεργαστή είναι να κανονικοποιήσει τις πολλαπλές κατατμημένες εγγραφές σε μία ολοκληρωμένη εγγραφή, ώστε να είναι δυνατή η αναγνώριση της υπογραφής μιας κακόβουλης εγγραφής από την μηχανή ανίχνευσης. Αν είναι ενεργοποιημένος ο Stream5, θα επεξεργαστεί μόνο την κίνηση του πελάτη (client).

- **Performance Monitor**

  Ο συγκεκριμένος προεπεξεργαστής επιτρέπει την μέτρηση της πραγματικής και θεωρητικής απόδοσης του Snort. Όταν είναι ενεργοποιημένος, μπορεί να τυπώνει τα στατιστικά στοιχεία είτε στην κονσόλα είτε σε ένα αρχείο για μετέπειτα επεξεργασία. Μερικά από την πληθώρα στοιχείων που παράγει είναι το ποσοστό χαμένων πακέτων, η

χρήση του δικτύου, η χρήση της CPU και πολλά στατιστικά όσο αφορά τις συνδέσεις του δικτύου.

- **HTTP Inspect**

  Είναι ένας γενικά αποκοδικοποιητής HTTP για εφαρμογές χρήστη. Ο προεπεξεργαστής βρίσκει στον buffer του δικτύου τα πεδία των HTTP δεδομένων και τα κανονικοποιεί. Με τον όρο κανονικοποίηση εννοείται η διαδικασία «μετάφρασης» μιας ασαφούς συλλογής χαρακτηριστικών, όπως οι Unicode, σε μια συλλογή χαρακτηριστικών που είναι αναγνωρίσιμη από το Snort. Η κωδικοποίηση των δεδομένων HTTP είναι μέθοδος που χρησιμοποιείται ευρέως από τους crackers για να καλύψουν τα ίχνη μίας επίθεσης από το σύστημα ανίχνευσης εισβολής. Χωρίς τον προεπεξεργαστή, αυτό με μία μεταμφίεση των πακέτων ώστε να μην ταιριάζει στις υπάρχουσες υπογραφές ανίχνευσης, ο διακομιστής ιστοσελίδων θα θεωρούσε έγκυρο ένα τέτοιο URL.

- **SMTP Preprocessor**

  Ο προεπεξεργαστής αυτός χρησιμοποιείται για την αποκωδικοποίηση SMTP κίνησης. Σε έναν δεδομένο προσωρινό χώρο αποθήκευσης δεδομένων, μπορεί να αποκωδικοποιήσει το πρωτόκολλο και να εντοπίσει τις εντολές SMTP καθώς και τις απαντήσεις τους. Έχει την δυνατότητα, εκτός από την κανονικοποίηση της ροής δεδομένων SMTP, να ελέγχει για αδυναμίες υπερχείλισης της μνήμης (Buffer Overflow) και συμπεριφοράς που δεν είναι ορισμένες στα RFC.

- **POP**

  Είναι ένας αποκωδικοποιητής POP3 για τις εφαρμογές του χρήστη. Λαμβάνει υπόψη την ροή δεδομένων όπου και αποκωδικοποιεί εντολές POP3 και τις απαντήσεις αυτών. Ο προεπεξεργαστής, αποθηκεύει την κατάσταση των μεμονωμένων πακέτων. Ωστόσο, η διατήρηση της σωστής κατάστασης των πακέτων αυτών, εξαρτάται από την επανασυναρμολόγηση του διακομιστή. Θα πρέπει να είναι

ενεργοποιημένος ο προεπεξεργαστής Stream5 και οι θύρες του POP να εμπεριέχονται στις παραμέτρους του stream5 για την λειτουργία και σωστή επανασυναρμολόγηση του POP. Τέλος ο POP χρησιμοποιεί GID 142 για την καταγραφή γεγονότων.

- **IMAP**

  Ο προεπεξεργαστής αυτός είναι αποκωδικοποιητής IMAP4 για εφαρμογές του χρήστη. Βρίσκει και αποκωδικοποιεί στο δίαυλο δεδομένων, εντολές και απαντήσεις IMAP4. Μαρκάρει τις εντολές, τα δεδομένα τις επικεφαλίδας των πακέτων και εξάγει τα συνημμένα IMAP4 αποκωδικοποιώντας τα κατάλληλα. Για την λειτουργία αυτού του προεπεξεργαστή χρειάζεται η ενεργοποίηση του stream5 και χρησιμοποιεί GID 141 για την καταγραφή γεγονότων.

- **FTP/Telnet Preprocessor**

  Αυτός ο προεπεξεργαστής είναι μια βελτιωμένη έκδοση του παρωχημένου αποκωδικοποιητή Telnet, ο οποίος παρέχει τη δυνατότητα statefull ελέγχου ροών δεδομένων FTP και Telnet. Είναι ικανός να αποκωδικοποιήσει την ροή δεδομένων, να αναγνωρίσει τις εντολές και τις απαντήσεις FTP και Telnet καθώς και να κανονικοποιήσει τα πεδία αυτών. Ο προεπεξεργαστής ελέγχει τόσο τις αιτήσεις του πελάτη όσο και τις απαντήσεις του διακομιστή.

- **SHH**

  Ο προεπεξεργαστής SHH έχει σχεδιαστεί για να ανιχνεύει τα ακόλουθα προβλήματα ασφαλείας (exploits): Challenge-Response Buffer Overflow, CRC 32, Secure CRT και το Protocol Mismatch.

- **DNS**

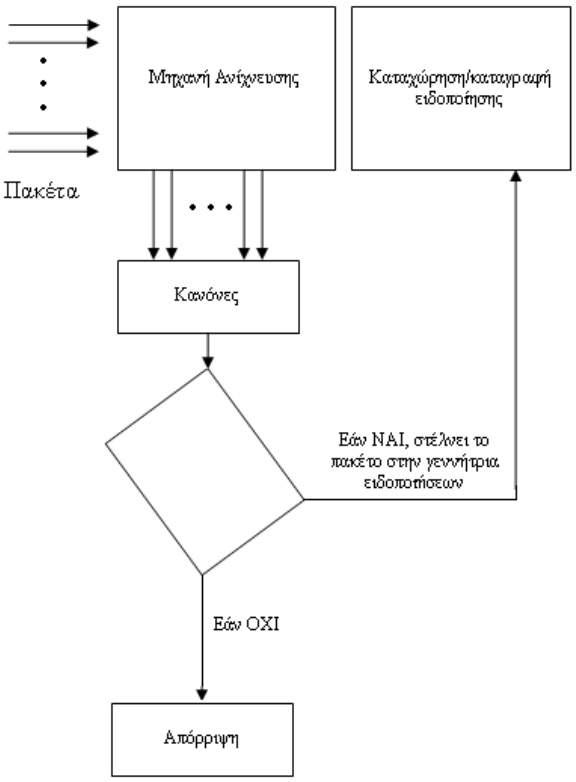
  Χρησιμοποιείται για την αποκωδικοποίηση απαντήσεων DNS και έχει την δυνατότητα να ανιχνεύει τα ακόλουθα προβλήματα ασφαλείας: DNS Client Rdata Overflow, Obsolete Record Types, και Experimental Record Types.

- **DCE/RPC 2**

  Ο κύριος σκοπός του προεπεξεργαστή είναι εκτελεί SMB ανακατάτμηση και DCE/RFC ανασυγκρότηση για να αποφύγει τους κανόνες αποφυγής αυτών των μεθόδων.

4.10 Μηχανή Ανίχνευσης (Detection Engine)

Αφού τα πακέτα θα ελεγχθούν από τους προ-επεξεργαστές που είναι ενεργοποιημένοι, τον λόγο έχει η μηχανή ανίχνευσης. Αυτή η διαδικασία θα μπορούσε να οριστεί ως η «καρδιά» του Snort. Σκοπός της είναι η συλλογή πληροφοριών από τον αποκωδικοποιητή πακέτων και τους προεπεξεργαστές, συγκρίνοντας τα περιεχόμενα των πακέτων με ένα σύνολο κανόνων με βάση την ανίχνευση του plug-in. Αν διαπιστωθεί κάποια ομοιότητα, στέλνει τα πακέτα στην διαδικασία καταχώρησης ή αποτύπωσης εξόδου αυτών και παραγωγής ειδοποίησης.



Εικόνα 10 Διάγραμμα διαδικασίας μηχανής ανίχνευσης του Snort

Οι κανόνες αυτοί περιέχουν υπογραφές για τις επιθέσεις. Δηλαδή η μηχανή ανίχνευσης είναι υπεύθυνη για την δημιουργία των υπογραφών, αφού επεξεργαστεί τους κανόνες. Η επεξεργασία των κανόνων γίνεται με την σειρά που βρίσκονται στο αρχείο και τοποθετούνται σε μια εσωτερική δομή δεδομένων. Η διαδικασία αυτή συμβαίνει κατά την εκκίνηση του snort, γεγονός που σημαίνει ότι, αν τροποποιηθεί κάποιος από τους κανόνες, θα πρέπει να γίνει επανεκκίνηση του λογισμικού.

4.11 ΠΑΚΕΤΑ ΛΟΓΙΣΜΙΚΟΥ ΑΠΟΤΥΠΩΣΗΣ ΕΞΟΔΟΥ (OUTPUT PLUG – INS)

Όταν στον έλεγχο της μηχανής ανίχνευσης ένα πακέτο αναγνωριστεί πως ταιριάζει με κάποιον από τους κανόνες, ενεργοποιείται μία ειδοποίηση, η οποία δημιουργεί και καταγράφει το συμβάν σε επιθυμητή μορφή. Το snort υποστηρίζει μια ποικιλία plug-ins, για την αποτύπωση των δεδομένων. Έτσι μερικές από τις υπηρεσίες που χρησιμοποιεί η έκδοση 2.9 του snort είναι:

- **Alert_syslog**

  Αυτή η υπηρεσία αποτυπώνει την έξοδο των αποτελεσμάτων του συστήματος του λογισμικού snort. Μπορεί να χρησιμοποιηθεί και για την καταγραφή πληροφοριών από άλλα διαφορετικά λογισμικά, όπως firewalls, server http κ.α.

- **Alert_fast**

  Τυπώνει τις ειδοποιήσεις που παράγονται σε διάταξη μίας σειράς ανά εγγραφή στο αρχείο που θα επιλεγεί. Είναι ο γρηγορότερος τρόπος αποτύπωσης ειδοποιήσεων συγκριτικά με την υπηρεσία alert full που θα ορίσουμε αμέσως μετά, αφού δεν εγγράφει τις κεφαλίδες των πακέτων στο αρχείο.

- **Alert_full**

  Το συγκεκριμένο plug-in δημιουργεί έναν κατάλογο για κάθε διεύθυνση που παράγεται ειδοποίηση και μέσα σε αυτόν αποθηκεύει σε αρχείο το αποκωδικοποιημένο περιεχόμενο των πακέτων, συμπεριλαμβανομένων και των κεφαλίδων του. Είναι ένας απαρχαιωμένος τρόπος καταγραφής, γιατί δεσμεύει μεγάλο χώρο για την αποθήκευση των δεδομένων. Μπορεί να χρησιμοποιηθεί σε χαμηλής χωρητικότητας δίκτυα.

- **Log_tcpdump**

  Είναι μία υπηρεσία που καταγράφει τις ειδοποιήσεις σε ένα αρχείο σύμφωνα με την διαμόρφωση που υποστηρίζει το πρόγραμμα tcpdump.