

Università Politecnica delle Marche

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione



Tesi di Laurea

Progettazione e realizzazione di una sandbox open source per l'analisi automatica dei malware ed integrazione con la piattaforma SOAR in SOC distribuiti operanti in contesti eterogenei

Design and implementation of an open source sandbox for automatic malware analysis and integration with the SOAR platform in distributed SOC's operating in heterogeneous contexts

Relatore

Prof. Domenico Ursino

Correlatore

Ing. Edoardo Balducci

Candidato

Xiao Li Savio Feng

Anno Accademico 2020-2021

Indice

Introduzione	11
1 La cybersecurity	15
1.1 Che cos'è la cybersecurity?	15
1.2 Tipi di cyber threats	17
1.2.1 Minacce informatiche recenti	18
1.3 Analisi dei principali cyber attack noti a livello globale	19
1.3.1 Distribuzione degli attaccanti per tipologia	19
1.3.2 Distribuzione delle vittime per categoria	21
1.3.3 Distribuzione delle vittime per area geografica	22
1.3.4 Distribuzione delle tecniche di attacco	23
1.3.5 Analisi della severity degli attacchi	23
2 Cybertech e progetto sandbox	27
2.1 Azienda Cybertech	27
2.1.1 SOC	28
2.1.2 SOAR	32
2.2 Progetto sandbox	33
2.2.1 Contesto di lavoro	33
2.2.2 Le principali tecnologie utilizzate	33
3 Panoramica di Cuckoo	37
3.1 Introduzione a Cuckoo	37
3.1.1 La storia di Cuckoo	39
3.2 Funzionamento di Cuckoo sandbox	40
4 Installazione di Cuckoo su CentOS8 ed Ubuntu	43
4.1 Introduzione ai requisiti di Cuckoo	43
4.1.1 Installazione delle librerie di Python	44
4.1.2 Installazione di VirtualBox	46
4.1.3 Installazione di tepdump	46
4.1.4 Installazione di Volatility	47
4.1.5 Installazione di M2Crypto	48
4.2 Installazione di Cuckoo	48

5	Configurazione del sistema di Cuckoo	51
5.1	Macchina Host	51
5.1.1	cuckoo.conf	52
5.1.2	auxiliary.conf	53
5.1.3	machinery.conf	53
5.1.4	memory.conf	55
5.1.5	processing.conf	55
5.1.6	reporting.conf	56
5.2	Macchina Guest	57
5.3	Configurazione delle regole di Routing	61
5.4	Troubleshooting	62
6	Analisi dei malware	63
6.1	Cuckoo submit	63
6.2	Web interface	67
6.3	Cuckoo malware analysis	68
6.3.1	Risultati dell'analisi	71
7	Integrazione con la piattaforma SOAR	79
7.1	Cortex XSOAR	79
7.1.1	Integrazione di Cuckoo su Cortex XSOAR	80
8	Conclusioni	85
	Riferimenti bibliografici	87
	Ringraziamenti	89

Elenco delle figure

1.1	Grafico che mostra come, con l'avanzare degli anni, la conoscenza necessaria per effettuare un attacco sofisticato sia diminuita	16
1.2	Grafico che mostra il numero di attacchi nel periodo che va dal 2018 al primo semestre del 2021.	19
1.3	La tabella mostra le varie tipologie di attaccanti dal 2018 al primo semestre del 2021, nonché un confronto tra il primo semestre del 2021 e il secondo semestre del 2020.	20
1.4	Il grafico a barre mostra in percentuale la distribuzione degli attaccanti dal 2018 al primo semestre del 2021.	21
1.5	La tabella mostra la distribuzione delle vittime per categoria dall'anno 2018 al primo semestre del 2021, ed un confronto tra il primo semestre del 2021 e il secondo semestre del 2020.	21
1.6	Il grafico a barre mostra la prime 10 delle categorie prese di mira dagli attaccanti.	22
1.7	Il grafico a torta mostra la distribuzione delle vittime per continente nel primo semestre del 2021.	22
1.8	Nella tabella sono mostrate le varie tecniche di attacco nonché un confronto tra il primo semestre del 2021 e il secondo semestre del 2020.	23
1.9	Il grafico a barre mostra le varie tecniche di attacco in percentuale fatte dal 2018 al primo semestre del 2021.	24
1.10	Il grafico a barre mostra la variazione della severity in percentuale dal 2018 al primo semestre del 2021.	24
1.11	Il grafico a barre mostra la severity per le varie categorie di attaccanti del primo semestre 2021.	25
1.12	Il grafico a barre mostra la severity per le varie categorie di attaccanti del 2020.	25
1.13	Il grafico a barre mostra la severity per le varie tecniche di attacco effettuate nel primo semestre del 2021.	26
1.14	Il grafico a barre mostra la severity per le varie tecniche di attacco effettuate nel 2020.	26
2.1	La Figura mostra il logo dell'azienda Cybertech	27
2.2	Schema con le varie componenti principali del SOC	29

2.3	Stratificazione del SOC in Cybertech	31
2.4	Le componenti che formano un SOAR	32
2.5	Le componenti orchestrate da un SOAR	33
2.6	Logo di OpenVPN	34
2.7	Logo di MobaXterm	34
2.8	Logo di TeamViewer	35
2.9	Logo di Cuckoo sandbox	35
2.10	Logo di VirtualBox	35
2.11	Logo di Cortex XSOAR	35
3.1	Cuckoo sandbox	38
3.2	Google Summer of Code	39
3.3	Architettura del funzionamento di Cuckoo	41
4.1	Logo di Ubuntu	43
4.2	Logo di CentOS	44
4.3	Avvio di Cuckoo	49
5.1	File di configurazione di Cuckoo	52
5.2	Cuckoo.conf (Parte 1)	52
5.3	Cuckoo.conf (Parte 2)	53
5.4	Auxiliary.conf	53
5.5	VirtualBox.conf	54
5.6	Memory.conf	55
5.7	Processing.conf	56
5.8	Reporting.conf	56
5.9	Macchina virtuale con Windows 7 installato	57
5.10	Impostazioni di rete della Virtual Box	58
5.11	Inserimento dell'immagine iso delle GuestAdditions	58
5.12	Installazione della GuestAdditions sulla macchina Guest	59
5.13	Configurazione delle properties di rete della macchina Guest	59
5.14	Esecuzione di agent.py	60
5.15	Creazione dell'istantanea sulla macchina Guest	61
5.16	Snapshot	61
6.1	Opzioni di cuckoo submit	64
6.2	Funzionamento di ORM	64
6.3	File local_settings.py	68
6.4	Avvio da terminale dell'interfaccia web	68
6.5	Pagina iniziale di Cuckoo	69
6.6	Pagina web di Cuckoo con i risultati dell'analisi URL	69
6.7	Score dell'analisi URL	69
6.8	Struttura delle directory delle analisi	71
6.9	Screenshot generati dall'analisi fatta da Cuckoo	72
6.10	Screenshot di esempio dell'analisi URL generato da Cuckoo	73
6.11	Report dell'analisi URL (Parte 1)	73
6.12	Report dell'analisi URL (Parte 2)	74
6.13	Report dell'analisi URL (Parte 3)	74

6.14 Report dell'analisi URL (Parte 4)	74
6.15 Report dell'analisi URL (Parte 5)	75
6.16 Report dell'analisi URL (Parte 6)	75
6.17 Report dell'analisi URL (Parte 7)	75
6.18 Report dell'analisi URL (Parte 8)	75
6.19 Report dell'analisi URL (Parte 9)	75
6.20 Elenco delle sezioni con le informazioni complete dell'analisi	76
6.21 Report specifico dell'analisi URL (Parte 1)	76
6.22 Report specifico dell'analisi URL (Parte 2)	77
6.23 Report specifico dell'analisi URL (Parte 3)	77
6.24 Sezione recent delle analisi fatte	78
7.1 Sezione integrazioni su Cortex XSOAR	80
7.2 Avvio del server API	81
7.3 Integrazione di Cuckoo installata su Cortex XSOAR	81
7.4 Codice dell'integrazione di Cuckoo	82
7.5 Impostazioni di Cuckoo Sandbox su Cortex XSOAR	82
7.6 Messaggio di successo per l'avvenuta integrazione di Cuckoo su Cortex XSOAR	83
7.7 Comandi eseguibili su Cortex XSOAR	83
7.8 Lista dei task di Cuckoo	84

Elenco dei listati

4.1	Comandi per installare pip di Python2	44
4.2	Comandi per l'installazione delle librerie di Python per il funzionamento corretto di Cuckoo su Ubuntu	44
4.3	Comando per l'installazione di MongoDB su Ubuntu	45
4.4	Comando per l'installazione di PostgreSQL su Ubuntu	45
4.5	Comandi per l'installazione dei pacchetti di Python su CentOS8	45
4.6	Comando per l'abilitazione del repository MongoDB su CentOS8	45
4.7	Configurazione dei parametri per l'installazione di MongoDB su CentOS8	45
4.8	Comando per l'installazione di MongoDB su CentOS8	46
4.9	Comando per avviare il servizio MongoDB	46
4.10	Comando per l'installazione di VirtualBox su Ubuntu	46
4.11	Comandi per l'installazione di VirtualBox su CentOS8	46
4.12	Comandi per l'installazione di tcpdump su Ubuntu	47
4.13	Comandi per il setting delle funzionalità di tcpdump	47
4.14	Comandi per l'installazione di tcpdump su CentOS8	47
4.15	Comandi per l'installazione di Volatility su Ubuntu	47
4.16	Comandi per l'installazione di Volatility su CentOS8	47
4.17	Comandi per l'installazione di M2Crypto su Ubuntu	48
4.18	Comandi per l'installazione di M2Crypto su CentOS8	48
4.19	Comandi per l'installazione di Cuckoo su Ubuntu	48
4.20	Comandi per l'installazione di Cuckoo su CentOS8	48
4.21	Comando per avviare Cuckoo	49
5.1	Comando per installare vim	51
5.2	Comandi per creare la vboxnet0	61
5.3	Comandi per configurare le regole di Routing sulla macchina host	62
5.4	Comandi per configurare le regole di Routing sulla macchina Guest	62
5.5	Comandi per risolvere il Troubleshooting sulla vboxnet	62
6.1	Esempi di applicazione del comando cuckoo submit	63
6.2	Esempio di query con e senza un ORM	65
6.3	Esempio di utilizzo add_path()	66
6.4	Esempio di utilizzo add_url()	67
6.5	Comandi disponibili per avviare il servizio di interfaccia web	67

10 **Elenco dei listati**

6.6	Comando per avviare l'analisi con un pacchetto specifico	71
7.1	Comando per avviare il server API	80
7.2	Comando per visualizzare tutti i task di Cuckoo	81

Introduzione

In questi ultimi anni, il numero di attacchi informatici è aumentato in maniera esponenziale, sia a livello quantitativo che qualitativo (per la gravità del loro impatto).

Questi attacchi, secondo il rapporto Clusit di ottobre 2021, sono più che raddoppiati in tutto il mondo e di essi l'81% ha avuto scopi criminali, con l'obiettivo di estorcere alle vittime denaro o dati che avessero valore economico. Infatti, una delle categorie di attacchi informatici in maggior aumento è proprio quella Ransomware, un software malevolo che cifra i dati del sistema della vittima con l'obiettivo di ottenere un riscatto.

Per difendersi e gestire al meglio queste minacce, ogni azienda ha bisogno di qualcuno che tenga costantemente sotto controllo quello che avviene nei traffici dati; pertanto, la sola sicurezza perimetrale, basata sull'installazione di firewall, non è più sufficiente. Gli incidenti che possono causare gravi danni ad un'organizzazione sono imprevedibili e, pertanto, vanno gestiti al meglio.

La soluzione a questa situazione è quella di utilizzare i servizi offerti da Security Operation Center (SOC). Esso è costituito da persone, tecnologie e processi. Le sue dimensioni dipendono dalle esigenze che ha la specifica azienda di gestire la parte relativa alla sicurezza informatica. Generalmente, per le aziende più grandi, può essere conveniente crearne uno interno. Le medie e piccole aziende, invece, si appoggiano su aziende esterne che offrono tali servizi.

Il SOC controlla, previene, rileva e analizza tutti gli elementi che riguardano la sicurezza di un sistema informatico, seguendo una precisa procedura preventivamente stabilita. Anche l'organizzazione di un Security Operation Center è, infatti, in primis, un'attività strategica.

Il SOC è un centro operativo a 360°. Si occupa, infatti, sia della gestione che del monitoraggio dell'infrastruttura a livello di sicurezza, ma anche di tutti quei servizi proattivi, quali, la security awareness, l'early warning, il vulnerability assessment ed il security assessment.

Il SOC si occupa di:

1. Rilevare gli incidenti (data breach detection).
2. Rispondere tempestivamente ad essi (incident response).
3. Risolvere tutte le conseguenze che essi hanno causato (remediation).

4. Migliorare proattivamente il livello di protezione dei sistemi aziendali (vulnerability assessment e penetration test).

Per velocizzare ancora di più la risposta agli incidenti, si sfrutta un orchestratore denominato Security Orchestration, Automation and Response (SOAR). Esso permette di coordinare le persone, i processi e le tecnologie che appartengono al SOC, migliorando notevolmente sia i tempi di risposta che quelli di gestione degli incidenti.

Il SOAR permette di avere:

- *Automazione e orchestrazione*, in quanto consente al software di rilevare, risolvere e archiviare l'incidente in maniera automatica.
- *Visione centralizzata delle minacce*, in quanto raccoglie tutte le informazioni degli incidenti in un unico punto.
- *Risparmio di tempo*, in quanto automatizza tanti processi, permettendo, così, di risparmiare tempo.
- *Concatenazione dei playbook*, tramite la quale permette di definire le azioni e le procedure che occorre attivare per rispondere a un determinato incidente.
- *Integrazione con le varie componenti dell'infrastruttura*, in quanto permette di avere piena integrazione con tutti i sistemi aziendali

In questa tesi verrà realizzata una sandbox open source per l'analisi automatica dei malware, che poi verrà integrata nel SOAR utilizzata dall'azienda Cybertech. Tale sistema è importante per l'azienda, in quanto permette di garantire la privacy dei dati del cliente, attraverso l'esecuzione in loco del sistema di analisi dei malware.

Il progetto è stato svolto da remoto durante la pandemia da COVID-19 attraverso OpenVPN e connessione SSH o Team Viewer. La tecnologia di riferimento che si utilizzerà per la progettazione e la realizzazione della sandbox è Cuckoo Sandbox.

Nelle prime fasi del lavoro si sono comprese le motivazioni per le quali Cybertech necessitasse di una sandbox. Fatto ciò, c'è stato un periodo di studio dei concetti relativi al SOC, al SOAR ed a Cuckoo. Dopo di ciò, si sono eseguite l'installazione e la configurazione del sistema Cuckoo Sandbox nei sistemi operativi Ubuntu e CentOS8, per poi testare il suo funzionamento attraverso varie analisi sui malware. Infine, si è integrato il sistema di Cuckoo Sandbox, con la collaborazione del team SOC di Cybertech, su Cortex XSOAR, cioè il SOAR utilizzato da Cybertech.

La tesi è strutturata come di seguito specificato:

- Nel primo capitolo viene descritto in maniera generale, che cosa sia la cybersecurity, mostrando un'analisi dei principali attacchi cyber noti a livello globale.
- Nel secondo capitolo vengono illustrati l'azienda Cybertech e il progetto sandbox proposto da essa.
- Nel terzo capitolo viene fornita una panoramica di che cosa sia Cuckoo e del suo funzionamento.
- Nel quarto capitolo vengono mostrati l'installazione di Cuckoo e i suoi requisiti.
- Nel quinto capitolo viene descritta la configurazione del sistema di Cuckoo Sandbox
- Nel sesto capitolo vengono presentate le possibili analisi effettuabili da Cuckoo insieme ad un esempio delle stesse.

- Nel settimo capitolo viene mostrata l'integrazione di Cuckoo sulla piattaforma SOAR utilizzata da Cybertech.
- Nell'ottavo capitolo vengono presentate le conclusioni riguardo al tirocinio e al lavoro svolto.

La cybersecurity

Nella prima parte di questo capitolo si descriverà, in maniera generale, che cosa sia la cybersecurity con le sue varie tipologie di cyber threats. Nella seconda parte verrà mostrato un quadro generale degli attacchi cyber noti a livello globale dal 2018 fino al primo semestre del 2021.

1.1 Che cos'è la cybersecurity?

Siamo nel periodo della quarta rivoluzione industriale. Oramai l'espressione Industria 4.0 è il motto di tante imprese industriali. Esso permette di avere un aumento della propria competitività ed efficienza tramite l'interconnessione delle risorse e la digitalizzazione delle informazioni.

Siamo nel periodo dell'Internet of Things (IoT), dove qualsiasi oggetto può essere collegato ad Internet, rendendosi riconoscibile ed acquisendo intelligenza. Ma, soprattutto, stiamo vivendo nel periodo del Covid-19, dove le persone hanno acquisito, per necessità, un modo di vivere totalmente digitale, lavorando in smart working e cercando di essere connessi virtualmente tra di loro, purtroppo però, l'aumento di utenza connessa ad Internet, non è proporzionata all'aumento di conoscenza sulla sicurezza informatica, il che permette di avere un numero maggiore di vulnerabilità accessibili dagli attaccanti.

Di conseguenza, con l'avvento dello smart working, i cracker hanno adattato gli attacchi informatici, mirando soprattutto alle vulnerabilità presenti nelle VPN, o infettando malware attraverso diversi vettori di attacco. In particolare, recentemente c'è stato un attacco alla regione Lazio; gli attaccanti sono riusciti ad acquisire delle credenziali di un amministratore di alto livello, paralizzando totalmente il sistema sanitario della regione. La polizia postale, analizzando la VPN (ovvero la rete virtuale utilizzata per accedere ad un sistema informatico da un computer remoto) scopre che non solo sono riusciti ad avere le credenziali di accesso, ma hanno anche infettato il sistema con un Trojan chiamato Emotet, che permette ad essi di eseguire comandi a livello amministratore in qualsiasi momento. Gli attaccanti, inoltre, hanno inserito un Ransomware, che ha criptato tutti i dati (backup compresi) e hanno richiesto un riscatto per la decriptazione degli stessi. Tale tipologia di attac-

co non è affatto rara; anzi, come mostreremo in seguito, ha subito un incremento non indifferente nell'ultimo periodo.

Aumentare, quindi, le connessioni ad Internet significa espandere il Cyberspace, cioè l'area nella quale si possono effettuare i cyber attacchi. La cybersecurity definisce, quindi, la capacità di proteggersi dagli attacchi provenienti dal Cyberspace. In generale gli attacchi cyber vanno a mirare un asset sfruttando una determinata vulnerabilità, cioè attuando quello che viene chiamato exploit.

Come si può vedere in Figura 1.1, con l'avanzare degli anni, la conoscenza per poter effettuare un attacco sofisticato è diminuita sempre di più, permettendo, così, ad un range molto più ampio di persone di effettuare cyber attacchi pericolosi.

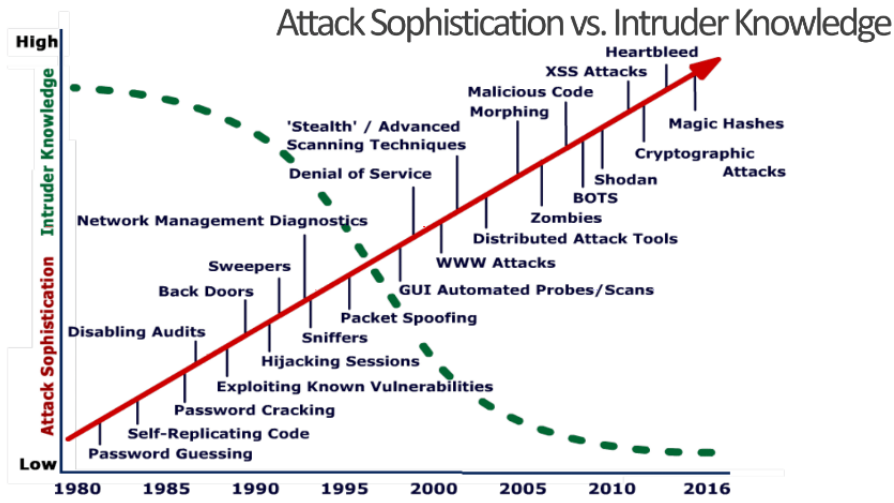


Figura 1.1. Grafico che mostra come, con l'avanzare degli anni, la conoscenza necessaria per effettuare un attacco sofisticato sia diminuita

Un cyber attacco ha come obiettivo quello di violare una delle seguenti policy di sicurezza:

- *Confidentiality*: garantisce che i dati e le risorse non siano utilizzabili o accessibili da parte di soggetti non autorizzati; questo potrebbe, anche, definire il concetto di anonimato, secondo il quale i dati pubblici non possono essere associati ai proprietari.
- *Integrity*: garantisce che i dati non possano essere modificati o cancellati impropriamente da utenti non autorizzati; la violazione dei dati può avvenire a diversi livelli, dall'utente semplice fino all'amministratore.
- *Availability*: garantisce agli utenti autorizzati di poter accedere ai dati quando lo richiedono, senza subire interruzioni di servizio.
- *Authenticity*: garantisce che l'informazione sia autentica; definisce il concetto di authentication in cui si verifica l'identità di una persona.

- *Assurance*: garantisce che l'entità si comporti come ci si aspetta; definisce il concetto di authorization secondo il quale l'entità deve essere autorizzata per poter compiere una determinata azione.
- *Accountability*: garantisce che le azioni di un'entità siano tracciabili; definisce la non repudation delle azioni.
- *Safety*: garantisce che il sistema non causi danni a persone o ambiente.
- *Reliability*: garantisce che il sistema eroghi i servizi in modo affidabile.
- *Resilience*: garantisce la ripresa del corretto funzionamento del sistema dopo aver ricevuto un attacco o una interruzione.

Le cause delle violazioni delle policy di sicurezza possono essere determinate sia da attacchi cyber, ma anche da errori umani; basti ricordarsi del servizio mandato in onda al Tg3, dove viene mostrato l'interno di un centro vaccinale, con un foglio affisso nel muro, riportante utente e password in chiaro per accedere al terminale del centro vaccinale.

Riassumendo, gli attacchi informatici attuati dai cracker hanno come obiettivo quello di accedere, modificare o distruggere informazioni sensibili, ma anche di estorcere denaro o interrompere il funzionamento dei normali processi aziendali. Molte aziende non riescono a garantire un budget consistente per potersi difendere e, soprattutto, c'è bisogno di sensibilizzare le persone e le aziende su tale aspetto. In aggiunta a questo, non è per niente facile implementare le misure di sicurezza perchè gli attaccanti stanno diventando sempre più innovativi e pericolosi. Il mondo è cambiato e dobbiamo accettarlo, preparandoci ad affrontare le nuove minacce. In conclusione possiamo definire la cybersecurity come la protezione dei sistemi informatici dalla cyber criminalità attraverso l'uso di strumenti tecnologici.

1.2 Tipi di cyber threats

Elenchiamo di seguito le principali minacce:

- *Phishing*: È una tecnica che attacca la vittima mediante un'e-mail fraudolenta con l'obiettivo di rubare dati sensibili o infettare il terminale. Non è facile difendersi da tale tecnica in quanto essa gioca molto sull'ingegneria sociale. Se l'e-mail è scritta bene e la fonte si avvicina molto ad una affidabile potrebbe diventare difficile riconoscere un'e-mail malevola. È il tipo più comune di attacco informatico.
- *Malware*: Il malware è un tipo di software progettato per ottenere l'accesso non autorizzato o per causare danni ad un computer. Possiamo dividere i malware in sottocategorie:
 1. *Ransomware*: Il ransomware è un tipo di software dannoso. È progettato per estorcere denaro bloccando l'accesso ai file o al sistema informatico fino al pagamento del riscatto. Tale pagamento, tra l'altro, non garantisce il ripristino dei file o il ripristino del sistema.
 2. *Virus*: un programma autoreplicante che infetta dei file con del codice malevolo in modo da diffondersi nel sistema informatico.

3. *Trojan*: un tipo di malware che si nasconde dentro un programma apparentemente legittimo. I criminali informatici inducono gli utenti ad eseguire o installare questo programma facendo, così, eseguire il codice del trojan nascosto.
 4. *Spyware*: un programma che registra segretamente ciò che fa un utente in modo che i criminali informatici possano utilizzare queste informazioni. Ad esempio, lo spyware potrebbe acquisire i dettagli della carta di credito. Anche questo tipo di malware, come il trojan, ha bisogno di essere eseguito o installato dall'utente.
 5. *Adware*: software pubblicitario che può essere utilizzato per diffondere malware.
 6. *Botnet*: reti di computer infetti da malware che i criminali informatici utilizzano per eseguire attività online senza il permesso dell'utente.
- *Social Engineering*: L'ingegneria sociale è una tattica utilizzata per indurre qualcuno a rivelare informazioni sensibili. Tale tecnica può essere combinata con una qualsiasi delle minacce cyber. Ad esempio, l'attaccante potrebbe fingersi una determinata persona per estorcere i corrispettivi dati sensibili a chi li gestisce, per poi accedere attraverso le sue credenziali ed effettuare un attacco.
 - *SQL Injection*: Una SQL injection è un tipo di attacco informatico utilizzato per prendere il controllo e rubare dati da un database. I criminali informatici sfruttano le vulnerabilità nelle applicazioni basate sui dati per inserire codice dannoso in un database tramite un'istruzione SQL malevola. Questo dà loro accesso alle informazioni sensibili contenute nel database.
 - *Attacco man-in-the-middle*: Un attacco "man-in-the-middle" è un tipo di minaccia informatica in cui un criminale informatico intercetta la comunicazione tra due individui per rubare dati. Ad esempio, su una rete Wi-Fi non sicura un utente malintenzionato potrebbe intercettare i dati trasmessi dal dispositivo della vittima e dalla rete.
 - *Denial of Service*: Sono attacchi volti a rendere inaccessibile alcuni tipi di servizi; si genera un numero di richieste talmente elevato verso il server, rendendo inaccessibile il servizio.
 - *Distributed Denial of Service*: È un attacco DoS fatto da diversi dispositivi.

1.2.1 Minacce informatiche recenti

Le ultime minacce informatiche da cui gli individui e le organizzazioni devono proteggersi, segnalate dai governi di Regno Unito, Stati Uniti e Australia sono:

- *Dridex malware*: Dridex è un trojan finanziario con una gamma di funzionalità. Colpisce le vittime dal 2014; infetta i computer tramite e-mail di phishing o malware esistente. Capace di sottrarre password, dettagli bancari e dati personali che possono essere utilizzati in transazioni fraudolente, ha causato enormi perdite finanziarie pari a centinaia di milioni. In particolar modo si è diffuso attraverso una campagna malspam a livello mondiale. Gli attacchi di phishing sono mascherati da fatture QuickBooks (un software di contabilità).
- *Romance scams*: Nel febbraio 2020, l'FBI ha avvertito i cittadini statunitensi di essere consapevoli delle frodi che i criminali informatici commettono utilizzando i siti di incontri, chat room e app. Gli autori si approfittano delle persone che

cercano nuovi partner, inducendo le vittime a fornire dati personali o foto intime con l'obiettivo di estorsione. È una tecnica che sfrutta totalmente l'ingegneria sociale. L'FBI riferisce che le minacce informatiche romantiche hanno colpito 114 vittime nel New Mexico nel 2019, con perdite finanziarie pari a 1,6 milioni di dollari.

- *Emotet malware*: Emotet è un sofisticato trojan che può rubare dati e caricare anche altri malware. L'obiettivo era quello di accedere a dispositivi stranieri e spiare i dati privati sensibili. La sua diffusione avviene tramite e-mail di spam. L'e-mail contiene un collegamento dannoso o un documento infetto; il malware, una volta che infetta il sistema, si diffonde come un worm, cercando di infiltrarsi in altri computer della rete. Il virus è polimorfico, ovvero il suo codice cambia leggermente ogni volta che vi si accede.

1.3 Analisi dei principali cyber attack noti a livello globale

In questa sezione mostreremo i principali attacchi cyber noti a livello globale secondo il rapporto Clusit di ottobre 2021. Il periodo preso in considerazione parte dal 2018 e arriva al primo semestre del 2021. Analizzeremo, poi, come è variata la severity negli ultimi anni e vedremo che ci sono dei risultati interessanti.

1.3.1 Distribuzione degli attaccanti per tipologia

Per mostrare la distribuzione degli attacchi per tipologia, si considera un campione costituito da 13.014 attacchi noti che hanno avuto un impatto rilevante in termini di perdita economica, reputazione o altro, avvenuti a partire dal 2011.

Nella Figura 1.2, viene mostrato l'andamento del numero degli attacchi informatici dal 2018 al primo semestre del 2021; il trend è chiaramente positivo. L'andamento cresce sempre, tranne nel periodo tra il secondo semestre del 2018 e il primo semestre del 2019.

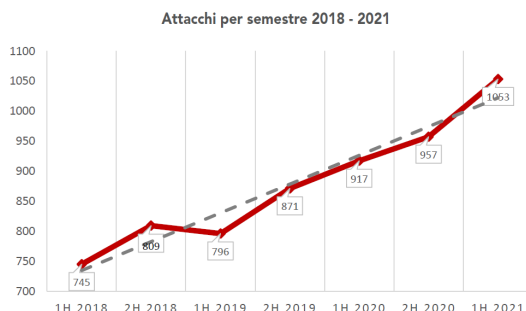


Figura 1.2. Grafico che mostra il numero di attacchi nel periodo che va dal 2018 al primo semestre del 2021

Nella Figura 1.3 si nota come nel secondo semestre 2020 il numero di attacchi gravi (975), rispetto a quelli del primo semestre 2021 (1053), abbia avuto un incremento del 10%. In particolare si ha un trend positivo del 18.2% per l'Information Warfare e del 21.1% per la Cybercrime, mentre diminuiscono gli attacchi della categoria Cyber Espionage del 36.7%, dopo un picco avuto nel 2020 dovuto allo sviluppo del vaccino per la cura del Covid-19. L'hacktivismo nel primo semestre del 2021 ottiene un enorme decremento rispetto al secondo semestre del 2020 pari al 66.7%. Nell'ultima riga si ha la somma degli attacchi di Espionage e Information Warfare in quanto, rispetto al passato, ad oggi non è facile distinguere le due tipologie di attacco.

ATTACCANTI PER TIPOLOGIA	2018	2019	2020	2H 2020	1H 2021	1H 2021 su 2H 2020	Trend 2021
Cybercrime	1.229	1.381	1.518	764	925	21.1%	↑
Espionage-Sabotage	203	203	264	150	95	-36.7%	↔
Hacktivismo	64	48	48	21	7	-66.7%	↓
Information Warfare	58	35	44	22	26	18.2%	↑
Espionage-Sabotage + Inf. Warfare	261	238	308	172	121	-29.65%	↔

Figura 1.3. La tabella mostra le varie tipologie di attaccanti dal 2018 al primo semestre del 2021, nonché un confronto tra il primo semestre del 2021 e il secondo semestre del 2020.

Di seguito vengono elencate le categorie descritte sopra:

- *Cybercrime*: Attività criminali effettuate mediante l'uso di strumenti informatici.
- *Hacktivismo*: Azioni, o attacchi informatici, effettuate per finalità politiche o sociali.
- *Espionage/Sabotage*: Azioni attuate per ottenere informazioni sensibili ed avere un vantaggio che sia politico, economico o altro.
- *Information Warfare*: Tecniche di gestione e di uso delle informazioni con lo scopo di ottenere un vantaggio di qualche tipo.

Gli attacchi mostrati nella Figura 1.4 fanno capire come nel 2018 la maggior parte degli attacchi erano di tipo Cybercrime; questo è rimasto costante negli anni rimanendo la principale forma di attacco. L'Espionage, probabilmente, è diminuito per la capacità di camuffare l'attacco. Ricordiamo che i dati sono relativi ad attacchi noti pubblici; quindi, se negli ultimi anni gli attaccanti hanno raffinato le tecniche di attacco di questa categoria, potrebbe darsi che essi non siano diminuiti, ma che, semplicemente, gli attaccanti hanno imparato a nascondersi meglio.

Tipologie e distribuzione attaccanti 2018 - 1H 2021

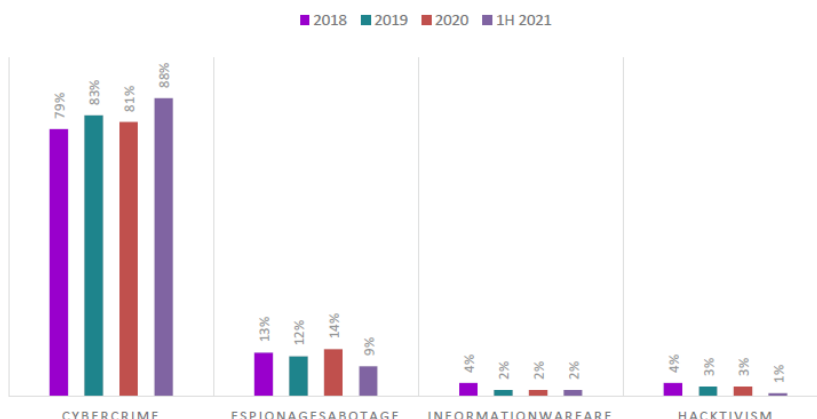


Figura 1.4. Il grafico a barre mostra in percentuale la distribuzione degli attaccanti dal 2018 al primo semestre del 2021.

1.3.2 Distribuzione delle vittime per categoria

In Figura 1.5 viene mostrata la distribuzione delle vittime per categoria dal 2018 al primo semestre del 2021. Come si può notare in figura, essa è composta da 20 macrocategorie con una tassonomia delle vittime derivata da standard internazionali. Nel confronto tra il secondo semestre 2020 e il primo semestre 2021 si nota subito che si ha un trend positivo soprattutto per le categorie “Transportation / Storage” (+108.7%), “Professional, Scientific, Technical” (+85.2%) e “News Multimedia” (+65.2%). Diminuiscono il numero di attacchi per “Telecommunications” (-43.8%) ed “Information Communication Technology” (-24.2%) ed, in misura minore, verso gli altri campi. Da notare come gli attacchi della categoria Multiple Targets sono diminuiti in quanto sono aumentati gli attacchi di tipo Ramsonware con single target.

VITTIME PER CATEGORIA	2018	2019	2020	2H 2020	1H 2021	1H 21 su 2H 20	TREND	VITTIME PER CATEGORIA	2018	2019	2020	2H 2020	1H 2021	1H 21 su 2H 20	TREND
Government, Military, Law Enforcement	220	233	224	120	167	39.2%	↑	Manufacturing	32	32	61	32	47	46.9%	↑
Healthcare	161	186	210	117	139	-18.8%	↓	News, Multimedia	70	69	43	23	38	65.2%	↑
Multiple Targets	326	406	401	158	121	-23.4%	↓	Organizations	40	35	46	29	30	3.4%	↔
Information Communication Technology	191	233	269	149	113	-24.2%	↓	Arts, Entertainment	68	55	40	19	26	36.8%	↑
Education	106	140	174	103	100	-2.9%	↔	Energy, Utilities	24	25	39	13	19	46.2%	↑
Financial, Insurance	162	107	122	66	60	-9.1%	↔	Hospitality	44	27	22	12	17	41.7%	↑
Professional, Scientific, Technical	18	19	59	27	50	85.2%	↑	Other Services	9	14	21	13	13	0.0%	-
Wholesale, Retail	33	45	54	31	50	61.3%	↑	Telecommunications	13	19	32	16	9	-43.8%	↓
Transportation, Storage	35	20	44	23	48	108.7%	↑	Construction	1	2	7	4	3	-25.0%	↔
								Agriculture, Forestry, Fishing	0	0	5	2	3	50.0%	↑

Figura 1.5. La tabella mostra la distribuzione delle vittime per categoria dall’anno 2018 al primo semestre del 2021, ed un confronto tra il primo semestre del 2021 e il secondo semestre del 2020.

Nella Figura 1.6 sono mostrate le prime 10 categorie prese di mira. “Government” si conferma al primo posto assoluto anche nel I semestre del 2021 (16% del totale). Si nota molto bene la variazione di “peso” tra la categoria Multiple Targets e le altre, che sono quasi tutte in crescita

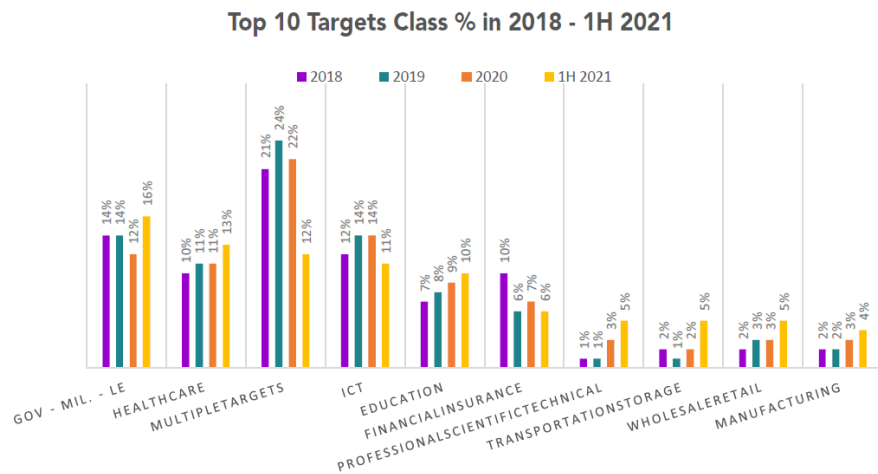


Figura 1.6. Il grafico a barre mostra la prime 10 delle categorie prese di mira dagli attaccanti.

1.3.3 Distribuzione delle vittime per area geografica

Nella Figura 1.7 si vede come la maggior parte delle vittime siano stati gli americani. In seconda posizione abbiamo gli europei.

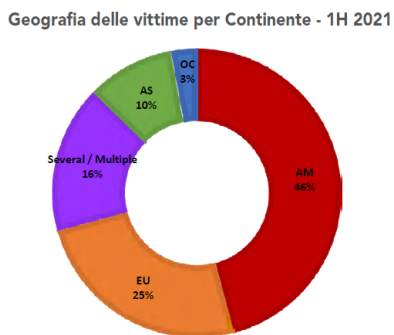


Figura 1.7. Il grafico a torta mostra la distribuzione delle vittime per continente nel primo semestre del 2021.

1.3.4 Distribuzione delle tecniche di attacco

In Figura 1.8 si vede come, dal primo semestre del 2021, rispetto al secondo semestre del 2020, si ha un aumento degli attacchi di tipo malware che oramai sono il 43% del totale. Le tecniche sconosciute sono anch'esse in aumento e tale categoria nasce dal fatto che molti attacchi diventano di dominio pubblico a seguito di un data breach, e non sempre in questi casi vi è una descrizione precisa della modalità dell'attacco da parte dell'azienda. L'attacco Denial of Service diminuisce del 42.9%; questo non dovrebbe stupirci in quanto con l'aumento dello smart working e con la pandemia del Covid-19 si è preferito mirare ad altre tecniche di attacco. Le vulnerabilità note sono aumentate del 41.4%; questo significa che gli attaccanti fanno affidamento sull'efficacia del malware prodotto a costi sempre più decrescenti e sullo sfruttamento di vulnerabilità note per gli attacchi.

Tecniche di attacco	2018	2019	2020	2H 2020	1H 2021	1H 2021 su 2H 2020	TREND
Malware	601	737	776	411	454	10.5%	↑
Unknown	429	309	368	202	230	13.9%	↑
Vulnerabilities	143	158	198	116	164	41.4%	↑
Phishing, Social Engineering	170	291	299	108	94	-13.0%	↓
Multiple Techniques	64	57	85	43	48	11.6%	↑
Identity Theft, Account Cracking	67	71	90	44	31	-29.5%	↓
Web Attack	43	21	17	12	20	66.7%	↑
Denial Of Service	37	23	34	21	12	-42.9%	↓
TOTALE	1.554	1.667	1.874	957	1.053		

Figura 1.8. Nella tabella sono mostrate le varie tecniche di attacco nonché un confronto tra il primo semestre del 2021 e il secondo semestre del 2020.

1.3.5 Analisi della severity degli attacchi

In Figura 1.9 vengono mostrate le varie tecniche di attacco utilizzate dal 2018 al primo semestre del 2021; notiamo subito come sia alta la percentuale dei malware rispetto alle altre tipologie di attacco. Nella figura viene analizzato l'impatto di ciascun attacco. Gli attacchi vengono classificati in quattro categorie: Critical, High, Medium e Low.

Nella Figura 1.10 possiamo vedere che nel 2020 gli attacchi con impatto Critical rappresentavano il 13% del totale, quelli High il 36%, quelli Medium il 32% ed, infine, quelli Low il 19%. Complessivamente, gli attacchi gravi con effetti molto importanti (High) o devastanti (Critical) nel 2020 erano il 49% del campione. Nel primo semestre 2021, invece, la situazione è molto diversa; gli attacchi gravi con severity High sono il 49%, quelli Critical il 25%. In questo caso gli attacchi con impatto Critical e High sono il 74%, il che dovrebbe preoccuparci.

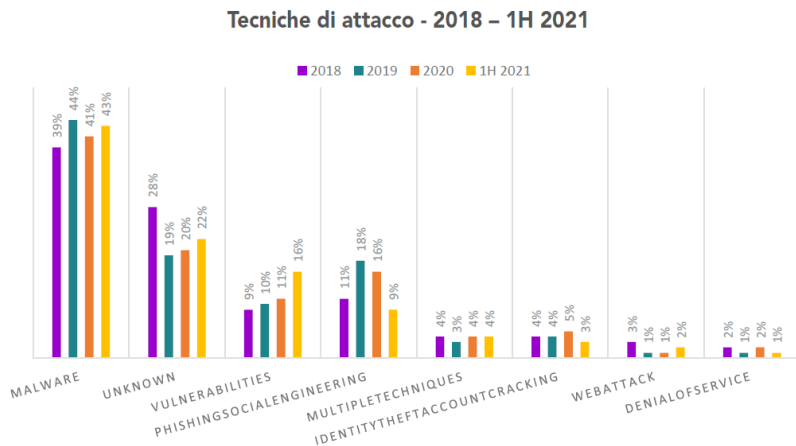


Figura 1.9. Il grafico a barre mostra le varie tecniche di attacco in percentuale fatte dal 2018 al primo semestre del 2021.

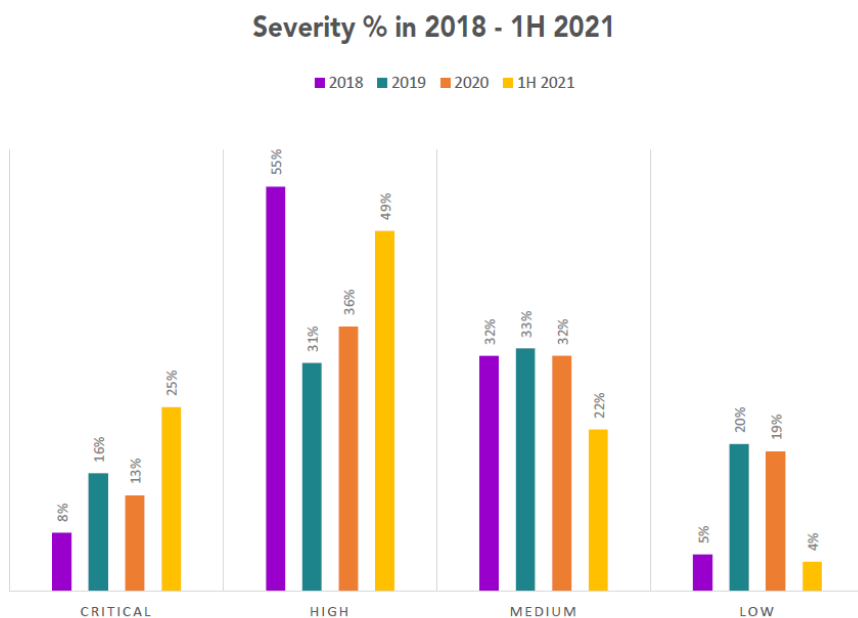


Figura 1.10. Il grafico a barre mostra la variazione della severity in percentuale dal 2018 al primo semestre del 2021.

Dalle Figure [1.11](#) e [1.12](#) si evince che nel primo semestre 2021 gli attacchi con Severity Critical realizzati per finalità Cybercriminal sono sensibilmente aumentati rispetto al 2020. L'attacco di tipo Espionage/Sabotage ha avuto un piccolo incremento sulla Severity di tipo Critical, e possiamo notare come Hacktivism sia diminuito rispetto al 2020.

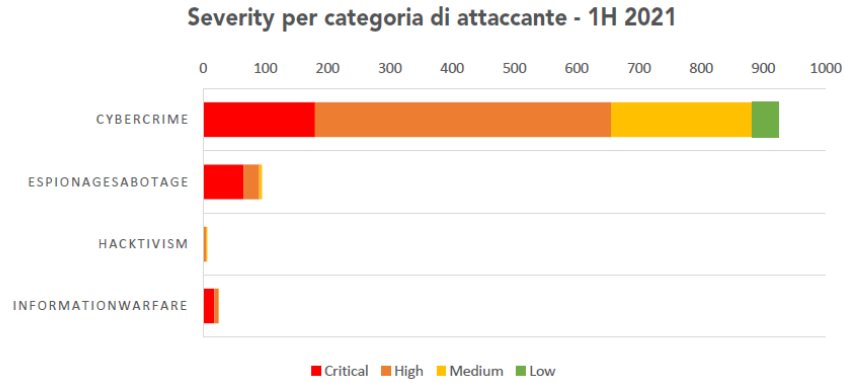


Figura 1.11. Il grafico a barre mostra la severity per le varie categorie di attaccanti del primo semestre 2021.

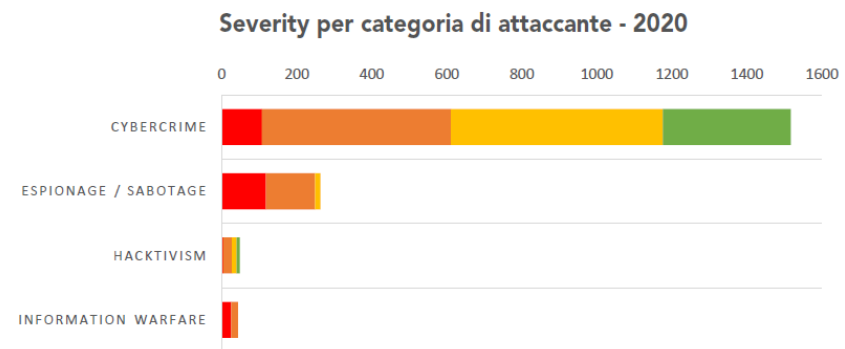


Figura 1.12. Il grafico a barre mostra la severity per le varie categorie di attaccanti del 2020.

Nella Figura 1.13 vengono mostrate le tecniche di attacco nel primo semestre 2021. Gli incidenti con impatto più critico sono quelli realizzati tramite Malware. Anche in questo caso l'evoluzione del trend è negativo. Confrontando i dati del primo semestre 2021 con quelli del 2020 mostrati in Figura 1.14, si nota un incremento significativo di attacchi con impatto “Critical” realizzati tramite Malware.

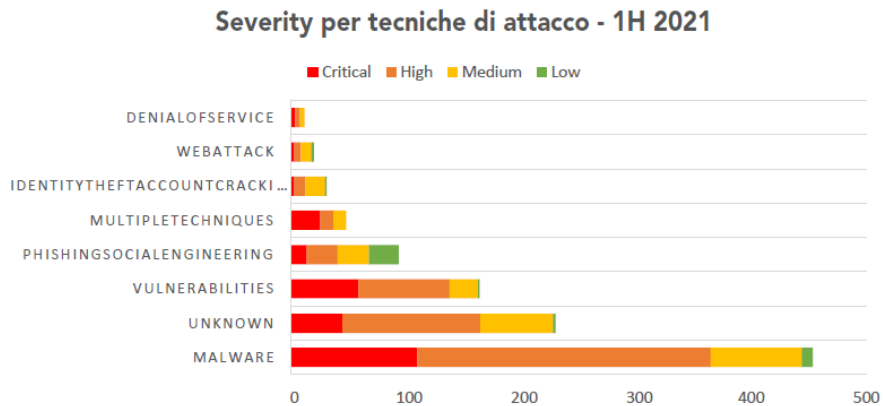


Figura 1.13. Il grafico a barre mostra la severity per le varie tecniche di attacco effettuate nel primo semestre del 2021.

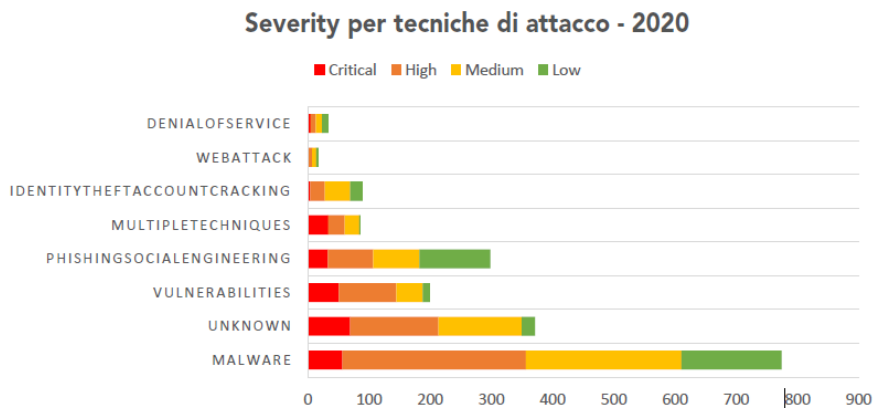


Figura 1.14. Il grafico a barre mostra la severity per le varie tecniche di attacco effettuate nel 2020.

Cybertech e progetto sandbox

Nella prima parte di questo capitolo si descriverà, in un'ottica generale, l'azienda Cybertech. Successivamente, si spiegherà cosa è un Security Operation Center (SOC) e un Security Orchestration, Automation and Response (SOAR). Nella seconda parte verranno descritte il progetto sandbox proposto dalla Cybertech e il contesto di lavoro nella quale è stato svolto con le principali tecnologie utilizzate.

2.1 Azienda Cybertech

L'azienda Cybertech, il cui logo è mostrato in Figura 2.1 è specializzata in sicurezza informatica. In particolar modo, è formata da tre gruppi di specialisti:

- *Network Operations Center (NOC)*: è composto da persone altamente specializzate nelle reti, e si occupa del monitoraggio e del controllo della rete.
- *Security Operation Center (SOC)*: è composto da persone altamente specializzate nell'analisi forense, e si occupa dell'analisi, gestione e risposta degli incidenti dovuti ad eventuali attacchi.
- *Penetration Test (PT)*: è composto da persone che seguono l'Ethical Hacking, e che sono altamente specializzate nella ricerca delle vulnerabilità interne ed esterne.



Figura 2.1. La Figura mostra il logo dell'azienda Cybertech

Cybertech ha iniziato ad offrire servizi di Security Operation Center nel 2018; l'anno successivo è entrata a far parte del gruppo Engineering Ingegneria Informatica, come azienda che si occupa di servizi riguardo la Cybersecurity. Essa, attualmente, non protegge solo clienti provenienti dall'Italia, ma estende i suoi servizi in tutta Europa, USA ed anche Sud America.

Affidarsi ad un'azienda come Cybertech è fondamentale, non solo per la sicurezza dei propri dati, ma anche per gestire al meglio tutti quei processi necessari per far mantenere l'azienda conforme a tutte le normative riguardanti la sicurezza informatica, una delle più importanti il GDPR.

Il General Data Protection Regulation (GDPR), è un regolamento dell'Unione Europea per il trattamento dei dati personali. Esso, è entrato in vigore il 24 maggio del 2016, diventando poi operativo dal 25 maggio del 2018. Il GDPR stabilisce le nuove regole per trattare i Dati Personali all'interno dell'Unione Europea e disciplina l'esportazione dei Dati Personali al di fuori dei confini UE. Nel nuovo Regolamento GDPR/18, in aggiunta, si definisce "Dato Personale" qualunque informazione relativa ad un individuo collegata alla sua vita, privata, e professionale, che pubblica, come, per esempio, nomi, foto, indirizzi e-mail, dettagli bancari, informazioni mediche o indirizzi IP di computer.

Tale Regolamento per la Protezione dei Dati definisce, quindi, i requisiti per il rispetto del Codice della Privacy. I diritti degli interessati devono essere gestibili in qualunque fase del ciclo di trattamento dei Dati Personali, sia su Internet che nei sistemi informatici. Tali diritti sono:

- il diritto all'Oblio del Dato Personale sui motori di ricerca su Internet;
- il diritto alla Cancellazione del Dato Personale;
- il diritto al Blocco del Trattamento del Dato Personale.

E questa è solo una delle tante leggi che devono rispettare le varie aziende.

2.1.1 SOC

L'azienda Cybertech è molto attiva nei servizi sulla Security Operation Center. È, quindi, importante capire cosa siano questi tipi di servizi. In questa sezione forniremo una descrizione generale degli stessi.

Un Security Operation Center è un'unità che fornisce servizi finalizzati alla sicurezza dei sistemi informativi; esso può essere interno all'azienda stessa o esterno verso clienti.

Il SOC monitora e analizza le attività su reti, server, endpoint, database, applicazioni, siti web e altri sistemi. L'obiettivo è la ricerca di comportamenti anomali che potrebbero indicare un attacco alla sicurezza del sistema. Esso opera sfruttando anche strumenti per la raccolta e per la correlazione di eventi, come i Security Information Event Management (SIEM).

Esso deve garantire che i potenziali attacchi siano adeguatamente identificati, analizzati, difesi, investigati e segnalati. E tale servizio deve essere garantito 24 ore al giorno, e 7 giorni su 7, ottenendo, così, un'analisi di tipo proattiva.

Si riporta, di seguito, una definizione di SOC data da *Pierre Jacobs, Barry Irwin* e *Alapan Arnab*, specialisti che lavorano per il Department Of Computer Science della Rhodes University a Grahamstown, in Sud Africa.

Un Security Operations Center (SOC) può essere definito come un'organizzazione di sicurezza informatica centralizzata che assiste le aziende nell'identificazione, gestione e risoluzione degli attacchi cyber distribuiti. L'obiettivo finale di un SOC, quindi, è quello di migliorare la posizione di sicurezza di un'organizzazione rilevando e rispondendo a minacce e attacchi, prima che abbiano un impatto sul business.

Nella Figura 2.2 si mostrano le varie componenti che costituiscono l'unità SOC.



Figura 2.2. Schema con le varie componenti principali del SOC

Si elencano, di seguito, alcuni servizi offerti dal SOC:

- *Security Incident Detection and Monitoring*: tale servizio garantisce il rilevamento delle anomalie, che si possono verificare nei flussi di dati che entrano ed escono dall'azienda. Questa raccolta viene effettuata dai SIEM, che, in caso di anomalie, genereranno un security alert. Esso viene confrontato con le informazioni di contesto fornite da servizi di cyber threat intelligence. Tale confronto viene attuato da sistemi automatizzati e configurati dall'unità SOC, in modo da allineare le azioni con il concetto di Prevention, oltre che di Detection e di Monitoring. In base, poi, ai risultati del confronto, si decide come gestire l'incidente, ovvero se richiedere un Incident Response con conseguente report o se effettuare direttamente un Remediation. Questa attività è effettuata dal SOC di livello 1.
- *Incident Response*: tale servizio, è effettuato dal SOC di livello2 o dal CERT (Computer Emergency Response Team) oppure dal CSIRT (Computer Security Incident Response Team). Dopo l'analisi del SOC di livello 1, si rilasciano

delle segnalazioni più specifiche. Il gruppo IR le analizzerà e le integrerà con le informazioni di threat intelligence in suo possesso, nonché con i dati relativi agli asset IT coinvolti nell'incidente. Il team, quindi, coordina l'attività di Remediation coinvolgendo tutte le parti in gioco.

- *DDoS Mitigation*: gli attacchi Distributed Denial of Service sono sempre i più attuati e difficili da contrastare, in quanto vengono attuati all'improvviso da vari dispositivi provenienti da tutto il mondo. Il servizio di DDoS Mitigation cerca di mitigare le conseguenze di un attacco DDoS seguendo tutte le procedure necessarie per la risoluzione dell'incidente di sicurezza.
- *Vulnerability Assessment*: tale servizio è svolto dal SOC di livello 3; l'obiettivo è quello di verificare in modo preventivo la sicurezza dei sistemi informatici con l'uso di tecniche come i penetration test; tale parte di analisi è altamente proattiva.
- *Analisi proattiva e gestione dei sistemi informatici*: tale servizio permette di avere un'analisi proattiva h24 dei sistemi informatici. Si sfruttano i sistemi di anti-intrusione come IDS, IPS e firewall, permettendo, così, una gestione centralizzata delle pratiche di sicurezza informatica con conseguente possibilità di identificare gli attacchi cyber provenienti dalla Cyberspace. Il SOC ha come vantaggio un'ottima scalabilità degli strumenti antintrusione, in quanto aggiungere o rimuovere IDS a quelli già esistenti non è un problema.
- *Security Device Management*: tale servizio, si occupa di rimediare ad un incidente informatico. Esso si suddivide in due principali processi:
 1. *Fault Management*: tale processo deve garantire il funzionamento continuo dell'infrastruttura di sicurezza interna o del cliente. L'attività prevede il monitoraggio costante di tutti i sistemi di sicurezza, la rilevazione e la segnalazione di tutti i fault, l'identificazione delle azioni da effettuare per il rimedio dell'incidente, l'implementazione delle azioni identificate ed infine il ripristino delle configurazioni in caso di perdita dei dati.
 2. *Configuration Management*: tale processo deve garantire costantemente l'allineamento delle regole di firewall alle esigenze del cliente. L'attività di configurazione e modifica delle policy di sicurezza è definita in base all'indirizzo sorgente, l'indirizzo di destinazione, il protocollo di rete, il protocollo di servizio etc.
- *Security alert*: tale servizio serve per avvisare i clienti della scoperta di una nuova vulnerabilità o di un comportamento anomalo, in modo tale da poter mitigare o annullare l'impatto di essi in modo tempestivo.
- *Security assessment*: tale servizio ha come obiettivo la valutazione del grado di sicurezza di un'azienda, esso è composto da due attività:
 1. *Vulnerability assessment*: tale attività, individua, le vulnerabilità note su reti informatiche, dispositivi, software, web application, etc. Per ogni assessment si effettua una valutazione contestuale e specifica in base alle esigenze.
 2. *Penetration test*: tale attività individua e sfrutta le vulnerabilità note e sconosciute dei sistemi informatici, dei servizi o degli applicativi web. Attraverso di essa si effettua un exploit per avere una stima del livello di sicurezza informatica e dell'impatto che essa causa. Si divide in varie fasi arrivando infine ad avere un report di tutte le attività svolte, delle vulnerabilità trovate e delle soluzioni proposte per far fronte ad esse.

- *Reportistica*: tale servizio, oltre a fornire le informazioni in dettaglio riguardo anomalie o tentativi di intrusione, permette al cliente di comprendere in maniera più specifica, quello che è successo attraverso una rielaborazione accurata di tutte le informazioni tecniche.
- *Assistenza tecnica*: tale servizio permette ai clienti di avere un'assistenza tecnica specialistica per tutte le problematiche riguardanti la sicurezza informatica; esso può essere effettuato sia da remoto che on-site, a seconda del contratto stipulato tra le parti.

Nella Figura 2.3 è rappresentata la stratificazione del SOC in Cybertech.

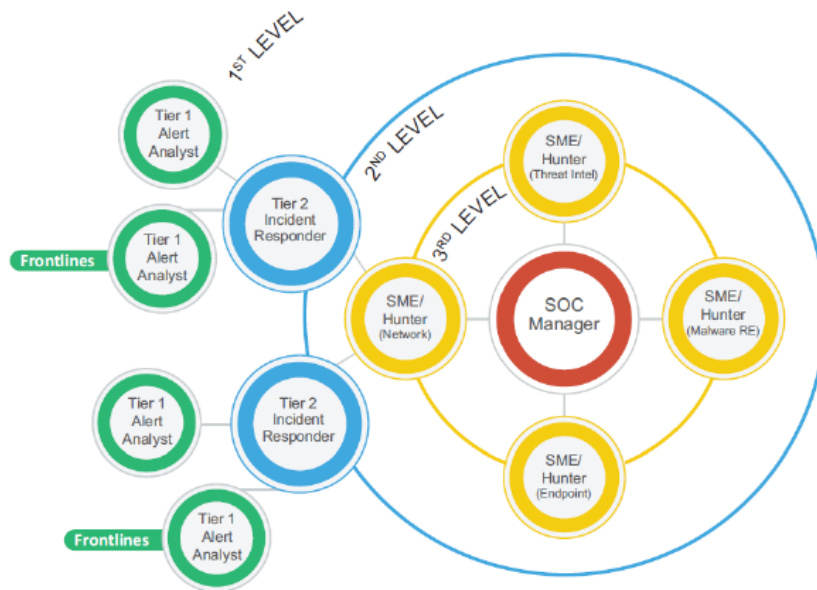


Figura 2.3. Stratificazione del SOC in Cybertech

Come si evince dalla figura, la stratificazione è a tre livelli con al centro il SOC Manager. I livelli sono:

1. *SOC di livello1*: effettua una prima analisi degli alert generati dal SIEM. L'obiettivo è quello di eliminare il maggior numero di falsi positivi, inviando poi gli alert considerati veri al secondo livello, che effettuerà un'analisi più approfondita. In questo livello si svolge l'attività di analisi forense.
2. *SOC di livello2*: gli alert che arrivano al secondo livello sono tutti quelli che il primo livello non riesce a risolvere, e che, quindi necessitano di una maggiore conoscenza. In questo livello si effettua un'analisi più approfondita dell'incidente analizzando, anche, il periodo precedente ad esso ed integrando le informazioni con quelle disponibili provenienti da diverse fonti. Se anche dopo l'analisi fatta dal livello 2 non si riesce a trovare la soluzione adatta per l'incidente, si passa al livello 3.

3. *SOC di livello 3*: tutti gli incidenti che arrivano a questo livello sono categorizzati come critici. Appena arrivano si agisce immediatamente. Il team che gestisce tale livello è composto da membri altamente specializzati nella sicurezza informatica.
4. *SOC Manager*: è la persona che, in questa stratificazione, gestisce tutte le risorse dell'azienda (personale, budget, etc.), e quindi, a livello strategico, è la persona importante.

2.1.2 SOAR

Cybertech, quindi, ha un SOC distribuito ed eterogeneo, ha sedi in tutta Europa e clienti operanti in tanti settori diversi. Per gestire e velocizzare tutti i processi della sicurezza informatica, Cybertech sfrutta un orchestratore, cioè un Security Orchestration, Automation and Response (SOAR).

Nella Figura 2.4 è mostrata la configurazione tipica di un SOAR, composto dal Security Orchestration and Automation (SOA), Threat Intelligence Platform (TIP) e Security Incident Response Platforms (SIR).



Figura 2.4. Le componenti che formano un SOAR

Un SOAR ha come obiettivo quello di far lavorare diverse tecnologie di sicurezza in maniera coordinata, armonizzata e automatizzata. L'automatizzazione permette di avere il minimo intervento umano possibile, garantendo risposte ad eventi, e quindi, l'esecuzione di processi per la risoluzione degli stessi in maniera automatica.

Un grande vantaggio che si ha con un SOAR è quello di riuscire a raccogliere tutte le informazioni provenienti da diverse fonti (threat source) in un unico punto, velocizzando e facilitando le loro analisi. Oltre a questo, il SOC permette, anche, di creare delle dashboard in modo da facilitare la visualizzazione delle informazioni ai vari clienti.

La Figura 2.5 mostra un SOAR che orchestra le principali componenti, cioè, le persone, i processi e le tecnologie.

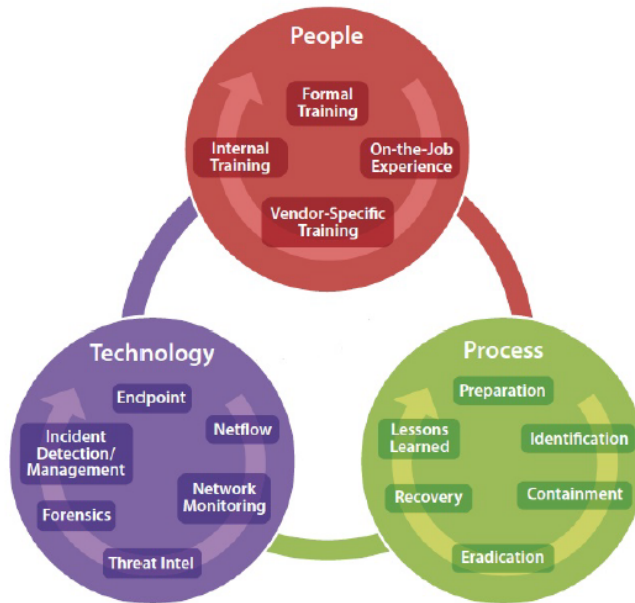


Figura 2.5. Le componenti orchestrate da un SOAR

2.2 Progetto sandbox

Il progetto sandbox nasce dalla necessità, da parte di Cybertech, di avere una sandbox locale per l'analisi automatica dei malware. Il vantaggio principale di questo progetto è quello di garantire la privacy di tutto ciò che viene analizzato, cosa che non si avrebbe sfruttando analizzatori malware online. Quindi, si andrà a sviluppare e a realizzare una sandbox, in un ambiente open source (Ubuntu e CentoS8), per l'analisi automatica dei malware, per poi integrarlo con la piattaforma SOAR utilizzata da Cybertech.

La sandbox che si utilizzerà sarà Cuckoo sandbox, e verrà installata e configurata in due sistemi operativi, ovvero Ubuntu e CentOS8.

2.2.1 Contesto di lavoro

Il progetto è svolto durante il periodo del COVID-19, e quindi in remoto. L'azienda consegna un certificato per l'autenticazione tramite VPN, e la connessione all'host si effettua attraverso SSH o Team Viewer. Le credenziali di accesso vengono assegnate dall'azienda stessa durante il primo giorno di tirocinio.

2.2.2 Le principali tecnologie utilizzate

Elenchiamo, di seguito, le principali tecnologie utilizzate durante lo svolgimento del progetto:

- *OpenVPN*: è una VPN open source utilizzata per creare dei tunnel crittografici punto-punto sicuri fra due host attraverso una rete non sicura (Internet). La libreria che sfrutta per la cifratura e autenticazione è quella relativa a OpenSSL. L'autenticazione può avvenire in 3 modi: certificati digitali, chiave segreta condivisa o credenziali utente/password. Nella Figura 2.6 è mostrato il logo di OpenVPN.
- *MobaXterm*: è un toolbox per gli accessi remoti; fornisce un'interfaccia grafica semplice e intuitiva per gestire le sessioni SSH, X11, RDP, VNC, FTP etc. Nella Figura 2.7 viene mostrato il logo di MobaXterm.
- *TeamViewer*: è un software per l'accesso e il controllo remoto degli host; permette di avere un accesso veloce al computer remoto e di trasferire file tra i due host in comunicazione, consentendo sia solo la visualizzazione del desktop della macchina, sia il controllo totale di essa. Nella Figura 2.8 è mostrato il logo di TeamViewer.
- *Cuckoo sandbox*: è una sandbox, open source, per l'analisi automatica dei malware. Nella Figura 2.9 è mostrato il logo di Cuckoo.
- *VirtualBox*: è un software gratuito e open source utilizzato per l'esecuzione di macchine virtuali; esso supporta come sistemi operativi host Windows, GNU/Linux e macOS. Esso, altresì, supporta come sistemi operativi guest Windows, GNU/Linux e OS/2 Warp. Nella Figura 2.10 è mostrato il logo di VirtualBox.
- *Cortex XSOAR*: è una piattaforma estesa di Security Orchestration, Automation e Response, che permette di velocizzare la risposta agli incidenti informatici. Essa integra tutti gli strumenti di sicurezza. Nel nostro caso integreremo, poi, Cuckoo su Cortex. Nella Figura 2.11 è mostrato il logo di Cortex XSOAR.



Figura 2.6. Logo di OpenVPN



Figura 2.7. Logo di MobaXterm



Figura 2.8. Logo di TeamViewer



Figura 2.9. Logo di Cuckoo sandbox



Figura 2.10. Logo di VirtualBox



Figura 2.11. Logo di Cortex XSOAR

Panoramica di Cuckoo

Nella prima parte di questo capitolo si descriverà la tecnica di sandboxing e si introdurrà Cuckoo. Nella seconda parte verrà descritto il funzionamento di Cuckoo sandbox.

3.1 Introduzione a Cuckoo

In sicurezza informatica, la tecnica del sandboxing è una tecnica che permette di verificare le funzionalità di un determinato codice non testato, o di programmi non attendibili, in un ambiente isolato.

Un esempio di utilizzo di una sandbox è Libra Esva Sandbox; essa integra tre motori di scansione antispam e antivirus e due sandbox per l'analisi degli allegati (QuickSand Defense) e degli URL (UrlSand Defense).

UrlSand Defense permette di reindirizzare tutti gli URL presenti nelle e-mail per analizzarli in un sandbox cloud. Tutti gli URL con comportamenti anomali vengono automaticamente bloccati.

QuickSand, invece, analizza tutti i contenuti attivi, come le macro, gli eseguibili incorporati o il codice javascript, durante il download dei messaggi e-mail, e li classifica in uno dei seguenti stati: sicuro, sospetto, indeterminato o crittografato. L'utente dopo la classificazione può decidere se ricevere il file, disinfettarlo e consegnarlo oppure bloccarlo. Tuttavia, chi scrive malware sa bene che prima o poi verrà testato in un ambiente virtuale; perciò l'attaccante cercherà sempre di scrivere il codice malevolo in maniera tale da renderlo il più invisibile possibile ai radar dei ricercatori.

Possiamo, quindi, definire una sandbox come un ambiente sterile nel quale eseguire tutto ciò che vogliamo monitorare in maniera sicura.

Elenchiamo, di seguito, alcuni vantaggi nell'utilizzare una sandbox:

- *Mitigazione del cyber risk*: le applicazioni che girano all'interno di una sandbox hanno meno probabilità di poter accedere direttamente alle chiamate di sistema, alle applicazioni, ai dati e, quindi, di conseguenza, alla macchina host.
- *Monitoraggio delle risorse hardware*: con l'utilizzo delle sandbox è possibile monitorare le risorse hardware, impostandone anche i limiti di allocazione (CPU, memoria, spazio disco, etc.).

- *Monitoraggio dei malware in ambiente isolato*: permette di analizzare e monitorare il comportamento dei malware in ambienti isolati e sicuri (sandboxing malware).

La sandbox che progetteremo e svilupperemo in questa tesi riguarda proprio il sandboxing malware; esso è un'applicazione pratica all'approccio dell'analisi dinamica. Pertanto, invece di analizzare staticamente il file binario, lo esegue e lo monitora in tempo reale. Questa tecnica viene utilizzata per completare l'analisi statica, ottenendo, così, informazioni aggiuntive sul comportamento del malware; essa, quindi, non sostituisce l'analisi statica, ma la completa.

Bisogna porre molta attenzione nelle fasi di creazione di un ambiente isolato, ad esempio una macchina virtuale, in quanto, questa rappresenta la parte più critica e importante per il deployment di una sandbox; pertanto, va eseguita con attenzione e con una solida pianificazione.

Di conseguenza, il piano di progettazione deve definire quale sistema operativo utilizzare e quale software installare e, soprattutto, quali versioni di essi, poiché la scelta di queste ultime si rileverà di particolare importanza durante l'analisi degli exploit.

Nella Figura 3.1 è rappresentata una configurazione generica del funzionamento di Cuckoo, esso sarà il sistema open source utilizzeremo per l'analisi automatizzata dei malware. Cuckoo lavora in modalità sandbox e viene utilizzato per eseguire e analizzare automaticamente i file, generando in output un report con i risultati dell'analisi del malware, fatta all'interno del sistema operativo isolato.

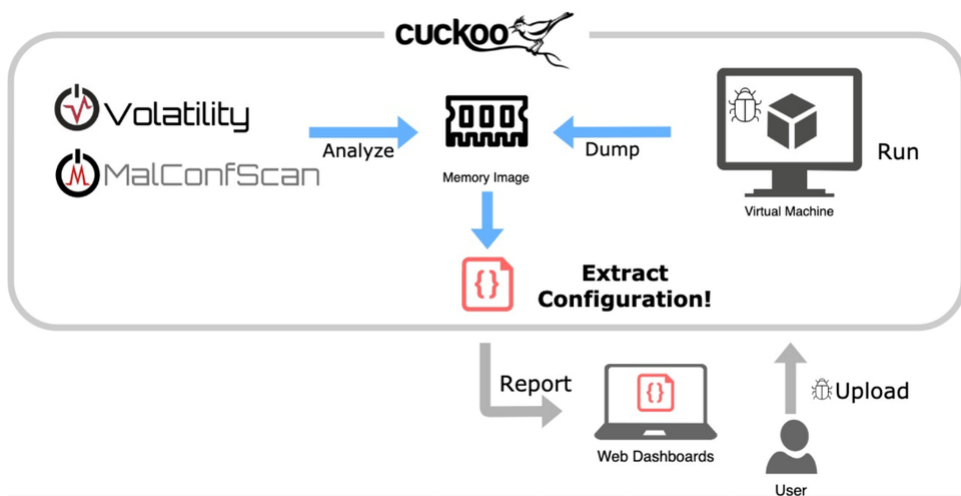


Figura 3.1. Cuckoo sandbox

L'analisi automatizzata dei malware non è di tipo deterministica, e il suo successo dipende da molteplici fattori. Spesso è difficile riprodurre un ambiente virtualizzato identico a quello reale; perciò, l'obiettivo principale è quello di creare un sistema in grado di gestire tutti i requisiti di cui ha bisogno rendendolo il più realistico possibile.

Cuckoo può generare nel report in output i seguenti risultati:

- tracce delle chiamate eseguite da tutti i processi generati dal malware;
- memory dumps dei processi malware;
- full memory dumps delle macchine;
- file creati, eliminati e scaricati dal malware durante la sua esecuzione;
- screenshot fatti durante l'esecuzione del malware;
- network traffic trace in formato PCAP.

3.1.1 La storia di Cuckoo

Cuckoo sandbox è iniziato come progetto in Google Summer of Code nel 2010 all'interno di The HoneyNet Project. È stato originariamente progettato e sviluppato da *Claudio Guarnieri*, che tuttora è il capo progetto e lo sviluppatore principale.

La prima versione beta è stata pubblicata il 5 febbraio 2011.

Il 2 novembre 2011 Cuckoo ha rilasciato la Versione 0.2 come prima release stabile. Nel marzo del 2011 Cuckoo è stato nuovamente selezionato come progetto supportato da Google Summer Of Code 2011, sempre all'interno di the HoneyNet Project, e durante questo periodo le sue funzionalità sono state estese da *Dario Fernandes*.

Alla fine di novembre 2011, *Alessandro Tanasi* è entrato a far parte del team di sviluppo ed ha ampliato le funzionalità di elaborazione e di reporting di Cuckoo.

A dicembre del 2011 viene rilasciato Cuckoo Versione 0.3 e nel febbraio del 2012 la Versione 0.3.2.

Nel 2012, a fine gennaio, è stato aperto Malwr.com, un'istanza di Cuckoo sandbox pubblica e gratuita, che permette di analizzare i file caricati attraverso un'interfaccia completa.

A marzo 2012 Cuckoo sandbox vince il primo round del programma Magnificent7 organizzato da Rapid7.

Durante l'estate del 2012 *Jurriaan Bremer* è entrato a far parte del team di sviluppo e refactoring del componente di analisi di Windows, migliorando sensibilmente la qualità dell'analisi.

Il 24 luglio 2012 viene rilasciato Cuckoo sandbox Versione 0.4.

Il 20 dicembre 2012 esce Cuckoo sandbox Versione 0.5 "To The End Of The World".

Il 15 aprile 2013 viene rilasciato Cuckoo sandbox Versione 0.6, e poco dopo, viene rilasciato la seconda versione di Malwr.com.



Figura 3.2. Google Summer of Code

Il 1 agosto 2013 *Claudio Guarnieri*, *Jurriaan Bremer* e *Mark Schloesser* hanno presentato Mo' Malware Mo' Problems - Cuckoo sandbox in soccorso al Black Hat Las Vegas.

Il 9 gennaio 2014 viene rilasciato Cuckoo sandbox Versione 1.0.

Nel marzo 2014 Cuckoo Foundation nasce come organizzazione no-profit dedicata alla crescita di Cuckoo sandbox e dei progetti e delle iniziative a latere.

Il 7 aprile 2014 viene rilasciato Cuckoo sandbox Versione 1.1.

Il 7 ottobre 2014, Cuckoo sandbox Versione 1.1.1 viene rilasciato dopo che *Robert Michel* ha rivelato una vulnerabilità critica.

Il 4 marzo 2015, Cuckoo sandbox 1.2 è stato rilasciato con una vasta gamma di miglioramenti riguardanti l'usabilità di Cuckoo.

Durante l'estate 2015 Cuckoo sandbox ha iniziato lo sviluppo dell'analisi del malware per Mac OS X come progetto Google Summer of Code all'interno di The HoneyNet Project. *Dmitry Rodionov* si è qualificato per il progetto e ha sviluppato un analizzatore funzionante per Mac OS X.

Il 21 febbraio 2016 viene rilasciata la Versione 2.0 Release Candidate 1. Questa versione viene fornita con quasi due anni di sforzi combinati per rendere Cuckoo sandbox un progetto migliore per l'uso quotidiano.

Il team che gestisce attualmente Cuckoo è formato da *Claudio Guarnieri*, *Alessandro Tanasi*, *Jurriaan Bremer*, *Mark Schloesser*, *Koen Houtman*, *Ricardo van Zutphen* e *Ben de Graaff*.

3.2 Funzionamento di Cuckoo sandbox

Cuckoo è progettato sia per essere utilizzato come applicazione standalone che per essere integrato in framework più grandi, grazie al suo design estremamente modulare.

Cuckoo sandbox è in grado di:

- Analizzare diversi file dannosi (eseguibili, documenti d'ufficio, file pdf, e-mail, etc.) e siti Web dannosi in ambienti virtualizzati Windows, Linux, macOS e Android.
- Tracciare le chiamate API e il comportamento generale del file e distillare questo in informazioni e firme di alto livello, comprensibili da chiunque.
- Scaricare e analizzare il traffico di rete, anche se crittografato con SSL/TLS.
- Eseguire analisi avanzate della memoria del sistema virtualizzato infetto tramite Volatility.

Cuckoo è personalizzabile in qualsiasi aspetto, a partire dall'ambiente di analisi per arrivare all'elaborazione dei risultati, e al reporting.

Ogni analisi viene avviata in una macchina virtuale o fisica isolata. I componenti principali dell'infrastruttura di Cuckoo sono una macchina Host (il software di gestione) e una serie di macchine Guest (macchine virtuali o fisiche per l'analisi).

L'Host esegue il componente principale della sandbox che gestisce l'intero processo di analisi, mentre i Guest sono gli ambienti isolati in cui i campioni di malware vengono effettivamente eseguiti e analizzati in modo sicuro.

La Figura [3.3](#) mostra l'architettura del funzionamento di Cuckoo.

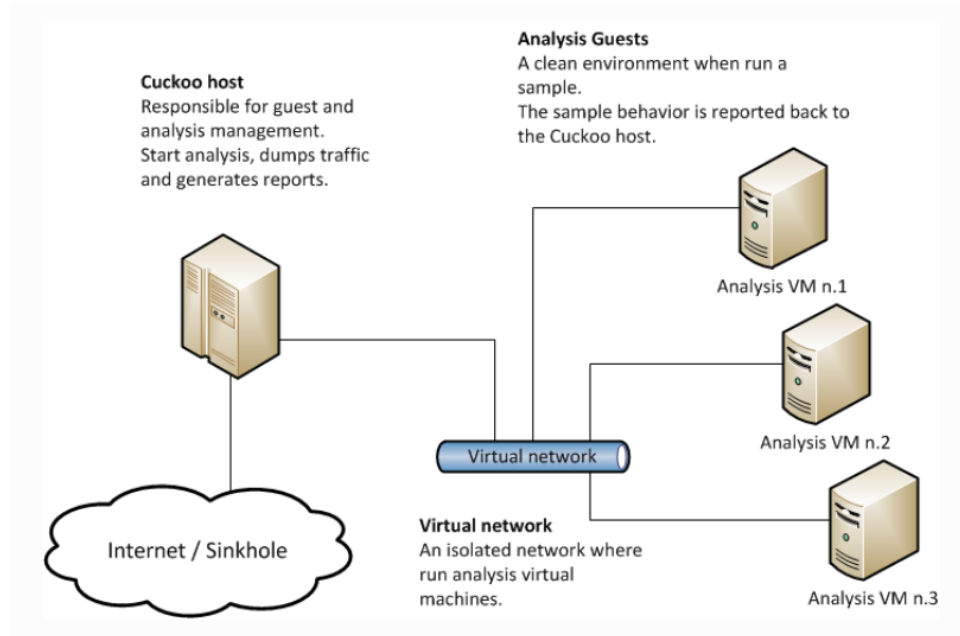


Figura 3.3. Architettura del funzionamento di Cuckoo

Nel nostro caso d'uso, utilizzeremo una macchina virtuale come host. Essa sfrutterà virtualbox per virtualizzare altri ambienti isolati, lavorando, quindi, su una macchina virtuale (host) che virtualizza altre macchine virtuali (Guest).

La macchina host invia il file, gli URL, etc. in analisi nella macchina Guest; quest'ultima manda ciascun file in esecuzione per fare l'analisi e il monitoraggio dei suoi comportamenti, con conseguente report dei risultati. Ogni volta che si esegue un'analisi, la macchina Guest si riporta allo Snapshot di riferimento, in modo da ritornare nell'ambiente preconfigurato per l'analisi dei malware. Le analisi possono essere fatte anche in parallelo su più macchine.

Installazione di Cuckoo su CentOS8 ed Ubuntu

In questo capitolo verranno mostrati tutti i requisiti necessari per un'installazione corretta di Cuckoo sui sistemi operativi Ubuntu e CentOS8.

4.1 Introduzione ai requisiti di Cuckoo

Il Sistema operativo Ubuntu è nato nel 2004 ed è basato su Linux, precisamente sul ramo unstable di Debian. Esso è composto principalmente da software libero, ma supporta anche software proprietari. La decisione di Cybertech di installare Cuckoo su tale sistema operativo è dovuto alla sua filosofia di software libero.

Ubuntu 20.04.3 LTS è installato nella sua versione utente; perciò, per collegarci alla macchina, utilizzeremo *OpenVPN* e *TeamViewer*.

Nella Figura [4.1](#) viene mostrato il logo di Ubuntu.



Figura 4.1. Logo di Ubuntu

Il sistema operativo CentOS8 è un'alternativa meno costosa, ma meno supportata rispetto ad Ubuntu ed è un sistema operativo nato per offrire una piattaforma enterprise per tutti coloro che vogliono utilizzare GNU/Linux per usi professionali. Esso è una distribuzione Linux derivata da Red Hat Enterprise Linux. Tale sistema è concepito principalmente per ambienti server. L'unica differenza concreta tra RedHat8 e CentOS8 è che quest'ultima non garantisce il supporto tecnico.

CentOS8 è installato in versione server; perciò, per collegarci alla macchina, utilizzeremo *OpenVPN* e *MobaXterm* in SSH. In Figura 4.2 viene mostrato il logo di CentOS.



Figura 4.2. Logo di CentOS

L'installazione dei requisiti di Cuckoo verrà mostrata sia per il sistema operativo Ubuntu che per CentOS8, e verrà eseguita da terminale.

Si ricorda di fare molta attenzione alle versioni dei pacchetti, in quanto questi possono facilmente compromettere l'installazione corretta di Cuckoo e generare errori.

Attualmente Cuckoo supporta completamente solo la versione di Python 2.7. Questi sono i requisiti minimi che deve avere la macchina host per far partire Cuckoo; in particolar modo, 2 GB di RAM e 50GB del disco servono per la virtualizzazione.

I requisiti hardware della macchina host sono:

- 320 GB di disco rigido;
- 4 GB di RAM;
- CPU quad-core.

4.1.1 Installazione delle librerie di Python

Tutti i componenti di Cuckoo sono scritti in Python e, per tale motivo, vanno installate tutte le librerie necessarie al suo funzionamento.

`pip` è un tool che ci permette di cercare, scaricare ed installare package Python che si trovano sul Python Package Index.

```
1 $ curl https://bootstrap.pypa.io/pip/2.7/get-pip.py --output get-pip.py
2 $ sudo python2 get-pip.py
```

Listato 4.1. Comandi per installare `pip` di Python2

`sudo apt-get install` è il comando per installare i pacchetti come `sudo user`, e permette di eseguire le operazioni con i privilegi di root.

```
1 $ sudo apt-get install python python-pip
2 $ sudo apt-get install python-virtualenv python-setuptools
3 $ sudo apt-get install libjpeg-dev zlib-dev swig
```

Listato 4.2. Comandi per l'installazione delle librerie di Python per il funzionamento corretto di Cuckoo su Ubuntu

MongoDB è un DBMS non relazionale, orientato ai documenti. Esso è di tipo NoSQL e gestisce al meglio i documenti in formato JSON, rendendo l'integrazione dei dati più veloce per alcuni tipi di applicazione. È necessaria la sua installazione per il funzionamento corretto dell'interfaccia Web.

```
1 $ sudo apt-get install mongodb
```

Listato 4.3. Comando per l'installazione di MongoDB su Ubuntu

Se si vuole utilizzare PostgreSQL come database, è necessario installare i corrispettivi pacchetti. PostgreSQL è un DBMS ad oggetti rilasciato con licenza libera.

```
1 $ sudo apt-get install postgresql libpq-dev
```

Listato 4.4. Comando per l'installazione di PostgreSQL su Ubuntu

```
1 $ sudo yum update
2 $ sudo yum install git -y
3 $ sudo yum --enablerepo=extras install epel-release
4 $ sudo yum install python2
5 $ sudo yum install python2-pip
6 $ sudo yum install python2-devel
7 $ sudo yum install libffi-devel
8 $ sudo yum install openssl-devel (alternativa di libssl-devel)
9 $ sudo yum install python2-virtualenv
10 $ sudo yum install python2-setuptools
11 $ sudo yum install libjpeg-devel
12 $ sudo yum install zlib-devel
13 $ sudo yum install swig
14 $ sudo yum install postgresql
15 $ sudo yum install libpq-devel
```

Listato 4.5. Comandi per l'installazione dei pacchetti di Python su CentOS8

Per l'installazione di MongoDB su CentOS8 bisognerà eseguire una procedura alternativa, in quanto MongoDB non è disponibile nei repository core di CentOS8.

Le operazioni vanno effettuate come utente root o utente con privilegi sudo. Si andrà ad utilizzare i comandi di nano.

```
1 $ sudo dnf install nano
2 $ sudo nano /etc/yum.repos.d/mongodb-org.repo
```

Listato 4.6. Comando per l'abilitazione del repository MongoDB su CentOS8

Per installare i pacchetti di MongoDB, si abilita il repository MongoDB creando un nuovo file repository denominato `mongodb-org.repo` all'interno della directory `/etc/yum.repos.d/`

```
1 [mongodb-org-4.2]
2 name=MongoDB Repository
3 baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.2/x86_64/
4 gpgcheck=1
5 enabled=1
6 gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc
```

Listato 4.7. Configurazione dei parametri per l'installazione di MongoDB su CentOS8

Dopo aver settato i parametri, installeremo il meta pacchetto `mongodb-org`.

```
1 sudo dnf install mongodb-org
```

Listato 4.8. Comando per l'installazione di MongoDB su CentOS8

Tramite il pacchetto `mongodb-org`, verranno installate come parte di `mongodb-org` i seguenti pacchetti:

- `mongodb-org-server`: il demone `mongod`, gli script e le configurazioni `init` corrispondenti.
- `mongodb-org-mongos`: il demone `mongos`.
- `mongodb-org-shell`: la shell `mongo`; questa è un'interfaccia JavaScript interattiva per MongoDB utilizzata per eseguire attività amministrative tramite riga di comando.
- `mongodb-org-tools`: contiene diversi strumenti MongoDB per l'importazione e l'esportazione di dati, statistiche e altre utilità.

Finita l'installazione, si può avviare ed abilitare il servizio `mongodb` attraverso il seguente comando:

```
1 sudo systemctl enable mongod --now
```

Listato 4.9. Comando per avviare il servizio MongoDB

4.1.2 Installazione di VirtualBox

Cuckoo supporta quasi tutte le soluzioni software di virtualizzazione. Noi utilizzeremo VirtualBox.

VirtualBox ci permetterà di creare le macchine virtuali Guest per l'esecuzione dei malware. La versione che si utilizza è la più recente ad oggi, ovvero la Versione 6.1.26.

```
1 $ sudo apt install virtualbox
```

Listato 4.10. Comando per l'installazione di VirtualBox su Ubuntu

La versione di VirtualBox installata su CentOS8 è la Versione 5.2, in quanto una delle più compatibili.

```
1 $ sudo dnf config-manager add repo=https://download.virtualbox.org/virtualbox/rpm/el/virtualbox.repo
2 $ sudo yum install VirtualBox-5.2
```

Listato 4.11. Comandi per l'installazione di VirtualBox su CentOS8

4.1.3 Installazione di tcpdump

Per eseguire il dump dell'attività di rete eseguita dal malware durante la sua esecuzione, si necessita di uno sniffer di rete volto ad acquisire il traffico di rete, in modo

da scaricarlo in un file. Cuckoo, per fare ciò, utilizza tcpdump, una soluzione open source.

La disabilitazione attraverso il comando `aa-disable` è richiesta quando si utilizza come cartella di default CWD, in quanto AppArmor impedirebbe la creazione dei file PCAP.

```
1 $ sudo apt-get install tcpdump apparmor-utils
2 $ sudo aa-disable /usr/sbin/tcpdump
3 $ sudo apt-get install tcpdump
```

Listato 4.12. Comandi per l'installazione di `tcpdump` su Ubuntu

Poiché, a differenza di Cuckoo, `tcpdump` richiede i privilegi di root, si sono impostate delle funzionalità specifiche per non fare andare in conflitto Cuckoo e `tcpdump`.

```
1 $ sudo groupadd pcap
2 $ sudo usermod -a -G pcap cuckoo
3 $ sudo chgrp pcap /usr/sbin/tcpdump
4 $ sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
```

Listato 4.13. Comandi per il setting delle funzionalità di `tcpdump`

```
1 $ sudo yum install tcpdump
2 $ sudo rpm -aq grep selinux
3 $ sudo yum install policycoreutils
4 $ sudo yum install policycoreutils-python
5 $ sudo yum install setools
6 $ sudo yum install setools-console
7 $ sudo yum install setroubleshoot
8 $ sudo yum install https://centos.pkgs.org/7/centos-z86_64/policycoreutils-python-2.5-34.el7.z86_64.rpm.html
```

Listato 4.14. Comandi per l'installazione di `tcpdump` su CentOS8

4.1.4 Installazione di Volatility

Volatility è uno strumento per eseguire analisi forense sui dump della memoria; esso può aiutare nella rilevazione di rootkit o di altre componenti malevoli nascoste. Cuckoo necessita di una versione pari o superiore alla Versione 2.3.

In questo caso effettuiamo un git clone del repository di volatility per poi procedere con l'installazione da locale.

```
1 $ sudo apt-get install git
2 $ sudo git clone https://github.com/volatilityfoundation/volatility
3 $ cd volatility
4 $ sudo python ./setup.py install
```

Listato 4.15. Comandi per l'installazione di Volatility su Ubuntu

```
1 $ sudo git clone https://github.com/volatilityfoundation/volatility
2 $ cd volatility
3 $ sudo python2 ./setup.py install
```

Listato 4.16. Comandi per l'installazione di Volatility su CentOS8

4.1.5 Installazione di M2Crypto

M2Crypto è il wrapper Python più completo per OpenSSL, con RSA, DSA, DH, EC, HMAC, message digest, cifrari simmetrici (incluso AES), etc.

Per l'installazione di M2Crypto si necessita della libreria SWIG. La versione di M2Crypto installata su Ubuntu è la Versione 0.38.0, ovvero la più recente.

```
1 $ sudo apt-get install swig
2 $ sudo pip install m2crypto
```

Listato 4.17. Comandi per l'installazione di M2Crypto su Ubuntu

Una delle versioni più compatibili con CentOS8 di M2Crypto è la Versione 0.35.2.

```
1 $ sudo yum install https://centos.pkgs.org/8/forensics-z86_64/python2-m2crypto-0.35.2-3.1.el8.z86_64.rpm.html
```

Listato 4.18. Comandi per l'installazione di M2Crypto su CentOS8

4.2 Installazione di Cuckoo

Prima di installare Cuckoo, bisogna creare un nuovo utente. Infatti, anche se si può eseguire Cuckoo direttamente dall'utente principale, è consigliato crearne uno nuovo dedicato solo alla tipologia di configurazione sandbox fatta. Bisogna fare attenzione al fatto che l'utente che esegue Cuckoo deve essere lo stesso utente che esegue le macchine virtuali, altrimenti Cuckoo non riuscirà a indentificare e avviare le macchine virtuali.

Inoltre, è necessario assicurarsi che il nuovo utente appartenga al gruppo `vboxusers` o al gruppo che si utilizza per eseguire VirtualBox.

Un altro consiglio è quello di aggiornare le librerie `pip/setuptools` in quanto molte volte potrebbero essere obsolete, causando problemi durante l'installazione di Cuckoo.

Si installa anche il pacchetto di `distorm3` necessario per l'installazione di Cuckoo.

```
1 $ sudo adduser cuckoo
2 $ sudo usermod -a -G vboxusers cuckoo
3 $ sudo pip install -U pip setuptools
4 $ sudo pip install distorm3
5 $ sudo pip install -U cuckoo
```

Listato 4.19. Comandi per l'installazione di Cuckoo su Ubuntu

```
1 $ sudo pip2 install distorm3
2 $ sudo pip2 install -U cuckoo
```

Listato 4.20. Comandi per l'installazione di Cuckoo su CentOS8

Eseguendo il comando `Cuckoo`, quest'ultimo verrà avviato, come mostrato nella Figura [4.3](#)

Configurazione del sistema di Cuckoo

Nella prima parte di questo capitolo verrà mostrato come configurare la macchina host. Nella seconda parte, verranno mostrate le configurazioni della macchina Guest e le varie cause principali che portano alla generazione di Troubleshooting.

5.1 Macchina Host

Cuckoo è costruito su un architettura master/slave. Il master è la macchina host e si occupa della gestione delle analisi e della generazione dei report. Gli slave sono le macchine Guest virtualizzate, e si occupano di raccogliere dati sull'esecuzione dei malware. La comunicazione tra host e Guest avviene attraverso un'interfaccia virtuale creata e gestita dall'hypervisor che, nel nostro caso, sarà Virtualbox.

Si illustreranno in seguito le configurazioni eseguite per la macchina Ubuntu, in quanto Cybertech, per motivi interni, ha deciso di far continuare il progetto soltanto sul sistema operativo Ubuntu.

Per configurare Cuckoo, bisogna modificare i seguenti file:

- `cuckoo.conf`: utilizzato per configurare il comportamento generale e le opzioni di analisi.
- `auxiliary.conf`: utilizzato per abilitare e configurare i moduli ausiliari.
- `machinery.conf`: utilizzato per definire le opzioni per il software di virtualizzazione (il file ha lo stesso nome del modulo machinery in `cuckoo.conf`); nel nostro caso sarà denominato `VirtualBox.conf`.
- `memory.conf`: utilizzato per la configurazione di Volatility.
- `processing.conf`: utilizzato per abilitare e configurare i moduli di elaborazione.
- `reporting.conf`: utilizzato per abilitare o disabilitare i formati di report.

È sufficiente configurare i file `cuckoo.conf` e `machinery.conf` per far funzionare Cuckoo. Tutti i file `.conf` sono nel path `CWD/conf/`. Per modificare tali file, utilizzeremo lo strumento `vim`.

```
1 $ sudo apt install vim
```

Listato 5.1. Comando per installare `vim`

Nella Figura 5.1 vengono mostrati i vari file `.conf` presenti.

```

x@feng@balducci-virtual-machine:~$ cd .cuckoo/
x@feng@balducci-virtual-machine:~/.cuckoo$ dir
agent      conf          distributed  __init__.py  monitor      signatures  stuff        supervisors.conf  whitelist
analyzer   cuckoo.db     elasticsearch  log          pidfiles     storage     supervisors    web              yara
x@feng@balducci-virtual-machine:~/.cuckoo$ cd conf/
x@feng@balducci-virtual-machine:~/.cuckoo/conf$ dir
auxiliary.conf  cuckoo.conf  kvm.conf  physical.conf  qemu.conf  routing.conf  vmware.conf  xenserver.conf
avd.conf        esx.conf    memory.conf  processng.conf  reporting.conf  virtualbox.conf  vsphere.conf

```

Figura 5.1. File di configurazione di Cuckoo

5.1.1 cuckoo.conf

Il file di configurazione `cuckoo.conf`, contiene le opzioni di configurazione generali di Cuckoo. In generale tutti i file `.conf` sono molto autoesplicativi; perciò, spiegheremo soltanto i campi di nostro interesse. La maggior parte dei campi vuoti vengono riempiti con dei valori di default.

Nella Figura 5.2 possiamo descrivere:

- *version check*: permette di effettuare un controllo sulla versione all'avvio di Cuckoo per vedere se ci sono versioni più recenti.
- *ignore vulnerabilities*: permette, durante l'avvio di Cuckoo, di effettuare un controllo delle vulnerabilità presenti, e se presenti, li segnala.
- *api token*: viene utilizzato come password durante l'integrazione di Cuckoo attraverso cuckoo API.
- *machinery*: permette di inserire il nostro software di virtualizzazione; nel nostro caso sarà VirtualBox.

```

[cuckoo]
# Enable or disable startup version check. When enabled, Cuckoo will connect
# to a remote location to verify whether the running version is the latest
# one available.
version_check = no

# Cuckoo will stop at startup if the version check reports vulnerabilities in
# one of Cuckoo's dependencies. This setting ignores the vulnerabilities
# and starts anyway
ignore_vulnerabilities = no

# The authentication token that is required to access the Cuckoo API, using
# HTTP Bearer authentication. This will protect the API instance against
# unauthorized access and CSRF attacks. It is strongly recommended to set this
# to a secure value.
api_token = bK4qAZUFipyldfYF8_nhEg

# Specify the name of the machinery module to use, this module will
# define the interaction between Cuckoo and your virtualization software
# of choice.
machinery = virtualbox

# Enable creation of memory dump of the analysis machine before shutting
# down. Even if turned off, this functionality can also be enabled at
# submission. Currently available for: VirtualBox and libvirt modules (KVM).
memory_dump = yes

```

Figura 5.2. Cuckoo.conf (Parte 1)

Nella Figura 5.3 è importante inserire con attenzione nel result server l'ip corretto della macchina host, in quanto permette di ricevere i risultati delle analisi.

Non tutti i software di virtualizzazione attivano le interfacce di rete virtuali, finchè non viene avviata una macchina virtuale, e VirtualBox è uno di questi. Cuckoo, quando viene avviato, cerca subito il collegamento con il result server; perciò, bisogna attivare la macchina virtuale, prima di avviare Cuckoo.

```
[resultserver]
# The Result Server is used to receive in real time the behavioral logs
# produced by the analyzer.
# Specify the IP address of the host. The analysis machines should be able
# to contact the host through such address, so make sure it's valid.
# NOTE: if you set resultserver IP to 0.0.0.0 you have to set the option
# 'resultserver_ip' for all your virtual machines in machinery configuration.
ip = 192.168.56.1
```

Figura 5.3. Cuckoo.conf (Parte 2)

5.1.2 auxiliary.conf

I moduli auxiliary sono script eseguibili durante l'analisi dei malware. Nella Figura 5.4 si mostra una parte del file di configurazione.

```
[sniffer]
# Enable or disable the use of an external sniffer (tcpdump) [yes/no].
enabled = yes

# Specify the path to your local installation of tcpdump. Make sure this
# path is correct.
tcpdump = /usr/sbin/tcpdump

# We used to define the network interface to capture on in auxiliary.conf, but
# this has been moved to the "interface" field of each Virtual Machinery
# configuration.

# Specify a Berkeley packet filter to pass to tcpdump.
# Note: packer filtering is not possible when using "nictace" functionality
# from VirtualBox (for example dumping inter-VM traffic).
bpf =

[mitm]
# Enable man in the middle proxying (mitmdump) [yes/no].
enabled = yes

# Specify the path to your local installation of mitmdump. Make sure this
# path is correct.
mitmdump = /usr/local/bin/mitmdump
```

Figura 5.4. Auxiliary.conf

5.1.3 machinery.conf

Questo file di configurazione definisce come Cuckoo deve interagire con il software di virtualizzazione. Poiché si utilizza VirtualBox, il file di configurazione è denominato VirtualBox.conf.

`VirtualBox.conf` è molto importante, in quanto, se non configurato in maniera corretta, genera molteplici errori durante l'avvio di Cuckoo o durante l'analisi dei malware.

Come mostrato in Figura 5.5, le varie opzioni impostabili sono le seguenti:

- `mode`: definisce la modalità nella quale Cuckoo avvierà la macchina virtuale.
- `interface`: definisce l'interfaccia network da considerare
- `machines`: vanno inserite tutte le macchine virtuali utilizzate per l'analisi dei malware.
- `platform`: definisce il sistema utilizzato dalla macchina virtuale.
- `ip`: definisce l'indirizzo IP della macchina Guest.
- `snapshot`: definisce lo snapshot da considerare per avviare l'analisi dei malware.

```
[virtualbox]
# Specify which VirtualBox mode you want to run your machines on.
# Can be "gui" or "headless". Please refer to VirtualBox's official
# documentation to understand the differences.
mode = gui

# Path to the local installation of the VBoxManage utility.
path = /usr/bin/VBoxManage
# If you are running Cuckoo on Mac OS X you have to change the path as follows:
# path = /Applications/VirtualBox.app/Contents/MacOS/VBoxManage

# Default network interface.
interface = vboxnet0

# Specify a comma-separated list of available machines to be used. For each
# specified ID you have to define a dedicated section containing the details
# on the respective machine. (E.g. cuckoo1,cuckoo2,cuckoo3)
machines = cuckoo1

# If remote control is enabled in cuckoo.conf, specify a port range to use.
# Virtualbox will bind the VRDP interface to the first available port.
controlports = 5000-5050

[cuckoo1]
# Specify the label name of the current machine as specified in your
# VirtualBox configuration.
label = cuckoo1

# Specify the operating system platform used by current machine
# [windows/darwin/linux].
platform = windows

# TeamViewer The IP address of the current virtual machine. Make sure that the
# IP address is valid and that the host machine is able to reach it. If not,
# the analysis will fail.
ip = 192.168.56.101

# (Optional) Specify the snapshot name to use. If you do not specify a snapshot
# name, the VirtualBox MachineManager will use the current snapshot.
# Example (Snapshot1 is the snapshot name):
snapshot = Snapshot1
```

Figura 5.5. `VirtualBox.conf`

5.1.4 memory.conf

Questo file di configurazione permette di avere, attraverso Volatility, tantissimi plug-in per l'analisi del dump di memoria. Per utilizzare Volatility bisogna abilitare Volatility nel file `memory.conf` e il memory dump nel file `cuckoo.conf`.

I dump di memoria possono occupare molto spazio; perciò è consigliato, a meno che non se ne abbia stretta necessità, di abilitare il `delete memdump`, in modo da risparmiare spazio sul disco.

Nella Figura 5.6 vengono mostrati i vari plug-in con la relativa sezione di configurazione. Il filter permette di rimuovere le informazioni già note nei report passati e si configura nella sezione `mask`, nella quale si possono inserire un elenco di pid per filtrare i processi già esistenti nella macchina.

```
# Volatility configuration

# Basic settings
[basic]
# Profile to avoid wasting time identifying it
guest_profile = Win7SP1x86
# Delete memory dump after volatility processing.
delete_memdump = no

# List of available modules
# enabled: enable this module
# filter: use filters to remove benign system data from the logs
# Filters are defined in the mask section at below

# Scans for hidden/injected code and dlls
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#malfind
[malfind]
enabled = yes
filter = yes

# Lists hooked api in user mode and kernel space
# Aiuto: it it to be very slow when enabled
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#apihooks
[apihooks]
enabled = no
filter = yes

# Lists official processes. Does not detect hidden processes
# http://code.google.com/p/volatility/wiki/CommandReference23#pslist
[pslist]
enabled = yes
filter = no

# Lists hidden processes. Uses several tricks to identify them
# http://code.google.com/p/volatility/wiki/CommandReferenceMal23#psxview
[psxview]
enabled = yes
filter = no
```

Figura 5.6. Memory.conf

5.1.5 processing.conf

`Processing.conf` permette di configurare tutti i moduli di elaborazione e definisce il modo in cui gestire i dati grezzi raccolti durante l'analisi.

Nella Figura 5.7 si mostrano alcuni moduli con la relativa sezione configurabile.

```

# Enable or disable the available processing modules [yes/no].
# If you add a custom processing module to your Cuckoo setup, you have to add
# a dedicated entry in this file, or it won't be executed.
# You can also add additional options under the section of your module and
# they will be available in your Python class.

[analysisinfo]
enabled = yes

[apkinfo]
enabled = no
# Decompiling dex files with androguard in a heavy operation. For large dex
# files it can really take quite a while - it is recommended to limit to a
# certain filesize.
decompilation_threshold = 5000000

[baseline]
enabled = no

[behavior]
enabled = yes

[memory]
# Create a memory dump of the entire Virtual Machine. This memory dump will
# then be analyzed using Volatility to locate interesting events that can be
# extracted from memory.
enabled = no

```

Figura 5.7. Processing.conf

5.1.6 reporting.conf

Il file `reporting.conf`, contiene tutte le impostazioni riguardanti il report dell'analisi fatta sulla macchina virtuale. È importante abilitare MongoDB per utilizzare l'interfaccia web.

```

# Enable or disable the available reporting modules [on/off].
# If you add a custom reporting module to your Cuckoo setup, you have to add
# a dedicated entry in this file, or it won't be executed.
# You can also add additional options under the section of your module and
# they will be available in your Python class.

[feedback]
# Automatically report errors that occurred during an analysis. Requires the
# Cuckoo Feedback settings in cuckoo.conf to have been filled out properly.
enabled = no

[jsondump]
enabled = yes
indent = 4
calls = yes

[singlefile]
# Enable creation of report.html and/or report.pdf?
enabled = no
# Enable creation of report.html?
html = no
# Enable creation of report.pdf?
pdf = no

[misp]
enabled = no
url =
apikey =
# The minimum Cuckoo score for a MISP event to be created
min_malscore = 0

tag = Cuckoo
upload_sample = no

[mongodb]
enabled = yes
host = 127.0.0.1
port = 27017
db = cuckoo
store_memdump = yes

```

Figura 5.8. Reporting.conf

5.2 Macchina Guest

La macchina Guest viene configurata attraverso VirtualBox, installando Windows 7, come mostrato in Figura 5.9. È importante chiamare la macchina con lo stesso nome inserito nei file di configurazione, altrimenti Cuckoo non riuscirà a trovarla. Si possono creare molteplici macchine per un'analisi parallela di diversi malware.

Nella Figura 5.10 vengono mostrate le impostazioni di rete. Bisogna utilizzare la scheda host only, altrimenti si potrebbero generare degli errori durante la comunicazione tra macchina host e Guest.

La rete host only è una rete interna in cui tutte le VM sono collegate tra loro; anche l'host è collegato ad essa. La rete interna dei Guest è la `vboxnet0`.

Tuttavia, le altre macchine esterne non possono vedere i Guest di questa rete; per questo viene chiamata "Host-only".

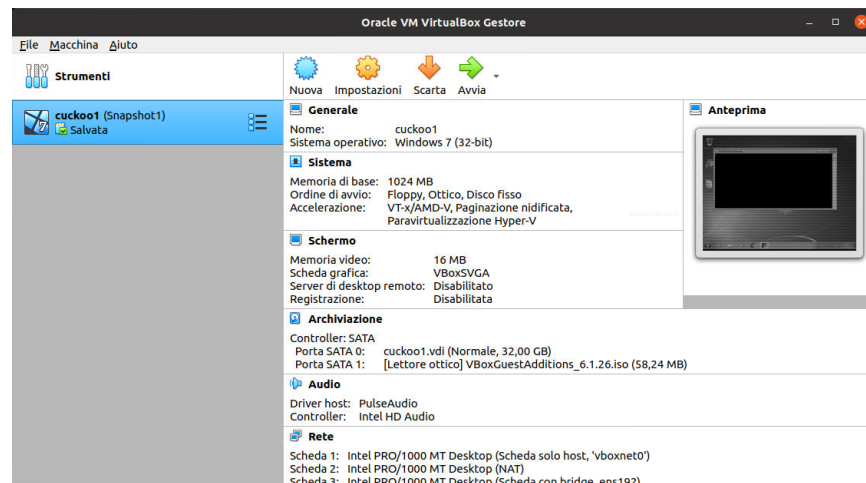


Figura 5.9. Macchina virtuale con Windows 7 installato

Bisognerà, poi, scaricare sulla macchina host la VBoxGuestAdditions e "settarla" su VirtualBox, come mostrato nella Figura 5.11.

Per installare la Guest Addition è necessario:

1. Scaricare il file Extension Pack dal sito ufficiale;
2. Aprire VirtualBox e cliccare su File-Preferenze;
3. Andare su Estensioni e, poi, sul pulsante aggiungi;
4. Selezionare il file delle Extension Pack
5. Avviare la macchina virtuale e cliccare in alto su Dispositivi → Inserisci l'immagine del CD delle Guest-Addition, come mostrato nella Figura 5.12.

Le Guest Addition servono per aggiungere funzionalità importanti alla nostra macchina virtuale. Esse consentono, ad esempio, di visualizzarla a schermo intero.

Nel seguito faremo vedere come è possibile condividere un file presente nella macchina host sulla macchina Guest.

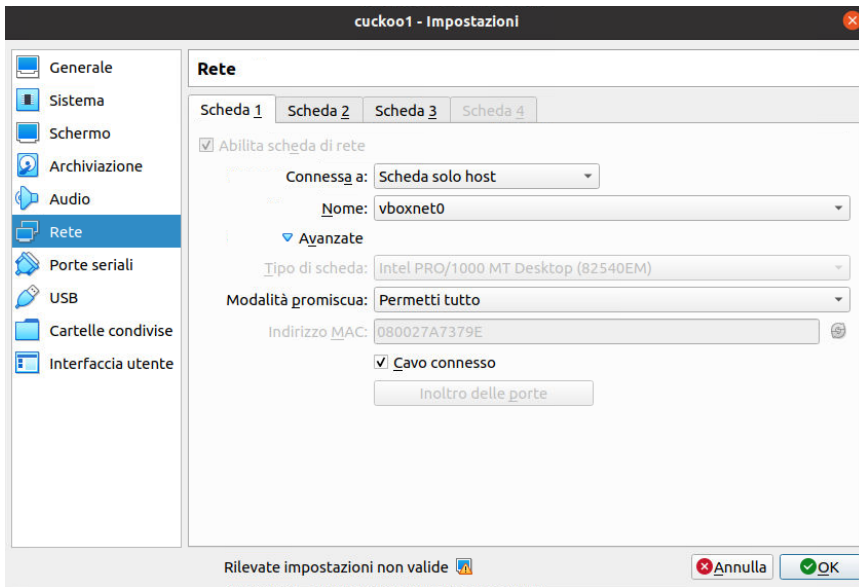


Figura 5.10. Impostazioni di rete della Virtual Box

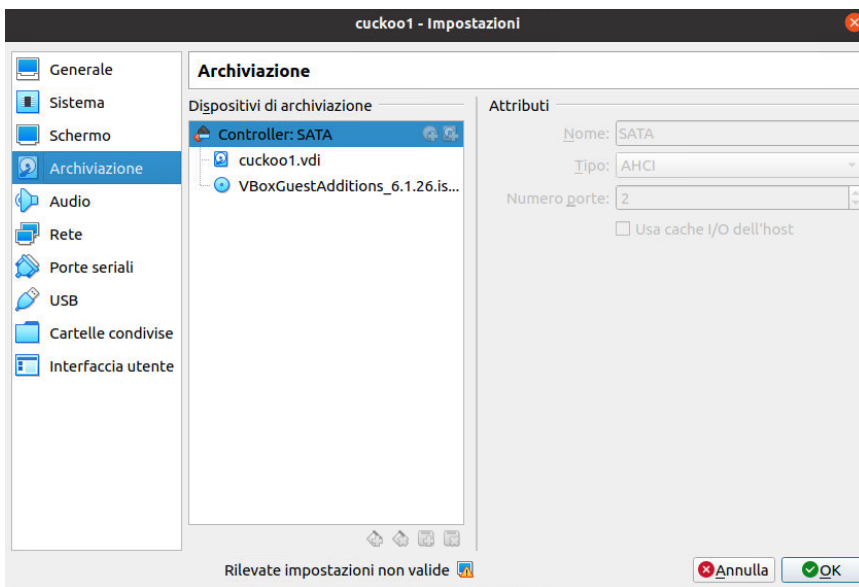


Figura 5.11. Inserimento dell'immagine iso delle Guest.Additions

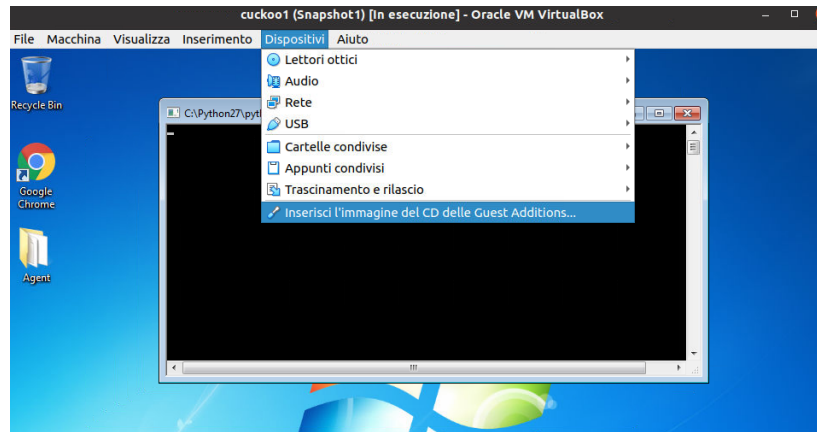


Figura 5.12. Installazione della GuestAdditions sulla macchina Guest

Dopo aver installato le GuestAdditions, è necessario configurare la rete della macchina Guest, come mostrato nella Figura [5.13](#).

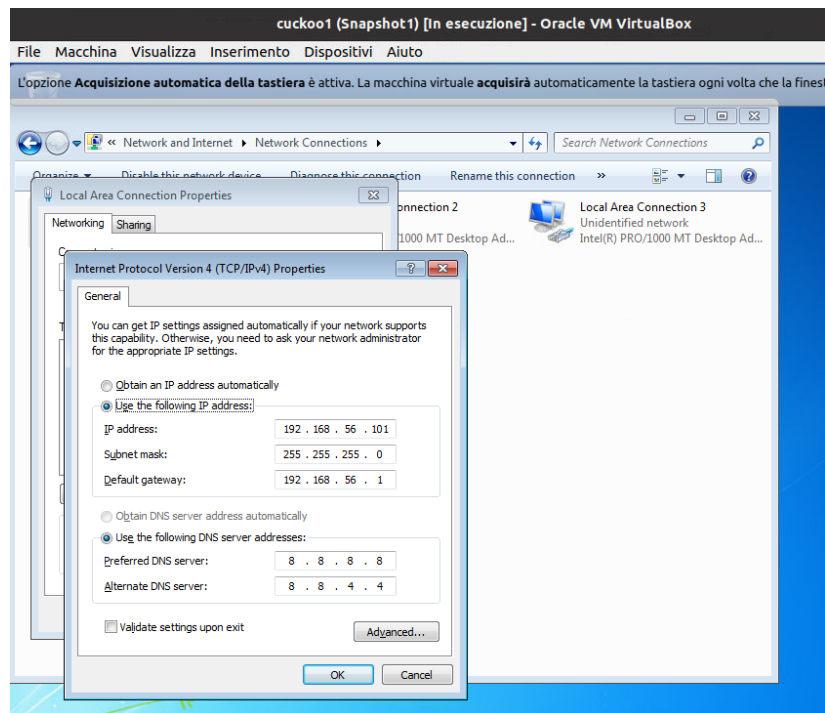


Figura 5.13. Configurazione delle properties di rete della macchina Guest

Per condividere con la macchina virtuale una cartella del nostro pc host dobbiamo cliccare su impostazioni e, dopo aver selezionato la macchina che ci interessa,

dobbiamo andare su "Cartelle condivise" e aggiungere tutte le cartelle desiderate; esse saranno automaticamente disponibili nel nostro sistema Guest.

In particolar modo, condivideremo un file denominato `agent.py` presente nel path `CWD/agent` della macchina host. Esso permette di gestire la comunicazione e lo scambio di dati tra host e Guest. Lo script `agent.py` va eseguito come mostrato nella Figura 5.14; l'agente avvierà un piccolo server API con cui l'host sarà in grado di comunicare. Per l'esecuzione automatica di esso durante l'avvio di Windows, basta spostare il file nella cartella di esecuzione automatica di questo sistema operativo.

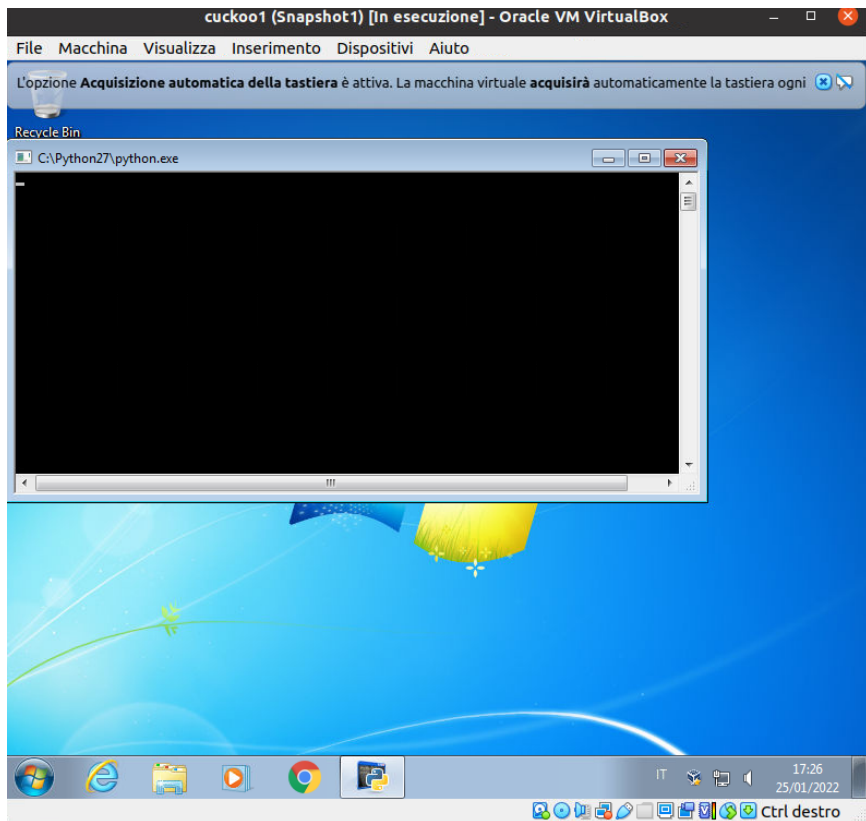


Figura 5.14. Esecuzione di `agent.py`

Dopo aver avviato l'agente sulla macchina Guest, è necessario creare un'istantanea, cioè uno snapshot di riferimento per Cuckoo come mostrato nelle Figure 5.15 e 5.16.

Quando Cuckoo manderà in esecuzione un malware, esso avvierà una macchina virtuale ripristinandola alla situazione dello snapshot scelto, in modo tale da poter avere, ad ogni esecuzione, la stessa situazione di partenza preimpostata.

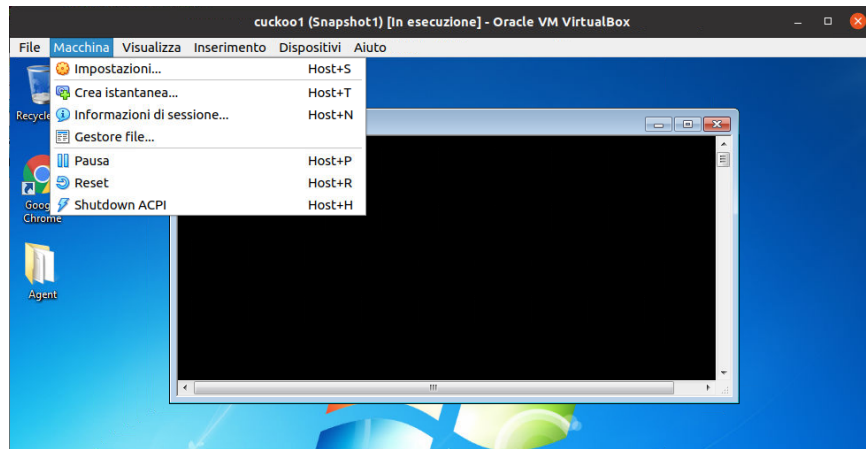


Figura 5.15. Creazione dell'istantanea sulla macchina Guest

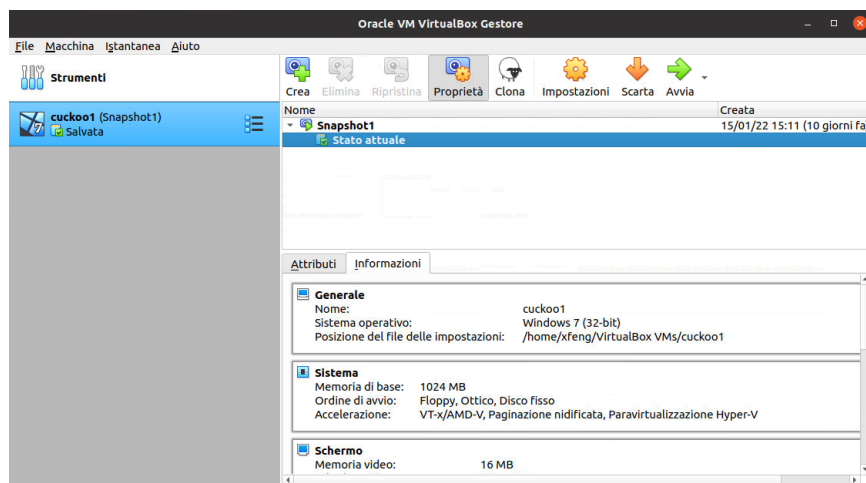


Figura 5.16. Snapshot

5.3 Configurazione delle regole di Routing

Cuckoo permette di includere il routing di rete per le analisi. Si può far avviare più analisi in maniera personalizzata nella stessa macchina; per esempio si può decidere che, nella prima analisi, l'accesso a Internet sia negato, e che la seconda analisi venga effettuata attraverso una VPN, etc.

```
1 $ vboxmanage hostonlyif create
2 $ vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1
```

Listato 5.2. Comandi per creare la `vboxnet0`

Dopo aver installato la macchina Guest, per creare la `vboxnet0`, si eseguono nella macchina host i comandi mostrati nel Listato 5.2.

```

1 $ sudo iptables -t nat -A POSTROUTING -o ens32 -s 192.168.56.0/24 -j MASQUERADE
2 # Default drop.
3 $ sudo iptables -P FORWARD DROP
4 # Existing connections.
5 $ sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
6 # Accept connections from vboxnet to the whole internet.
7 $ sudo iptables -A FORWARD -s 192.168.56.0/24 -j ACCEPT
8 # Internal traffic.
9 $ sudo iptables -A FORWARD -s 192.168.56.0/24 -d 192.168.56.0/24 -j ACCEPT
10 # Log stuff that reaches this point (could be noisy).
11 $ sudo iptables -A FORWARD -j LOG

```

Listato 5.3. Comandi per configurare le regole di Routing sulla macchina host

Nel Listato 5.3 si mostrano le impostazioni definite attraverso delle regole globali. Le seguenti iptables rules consentiranno alle macchine virtuali di accedere alla macchina host Cuckoo.

Le regole di Iptables non sono persistenti tra i riavvii; quindi, se si vuole mantenere tali regole, bisogna utilizzare uno script, o, semplicemente, installare iptables-persistent. A tal fine nella macchina Guest si eseguono i seguenti comandi di Iptables mostrati nel Listato 5.4.

```

1 $ sudo iptables -A FORWARD -o ens32 -i vboxnet0 -s 192.168.56.0/24 -m conntrack --ctstate NEW -j ACCEPT
2 $ sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
3 $ sudo iptables -A POSTROUTING -t nat -j MASQUERADE

```

Listato 5.4. Comandi per configurare le regole di Routing sulla macchina Guest

5.4 Troubleshooting

I principali Troubleshooting possono essere generati dalle seguenti motivazioni:

- malconfigurazione dei file di virtualbox o del file `cuckoo.conf`;
- mancanza della `vboxnet0`; in questo caso, eseguendo i comandi del Listato 5.5 si può risolvere il problema;
- eseguire un aggiornamento in maniera sbagliata;
- fare lo snapshot della macchina Guest con `agent.py` non avviato;
- nominare in maniera diversa lo snapshot rispetto a quanto scritto sul file `VirtualBox.conf`;
- avviare Cuckoo senza aver avviato prima la macchina virtuale;
- firewall non disabilitato sulla macchina Guest;
- utilizzare una versione di Python 3;
- utilizzare un indirizzo dinamico, piuttosto che un l'indirizzo statico, sulla macchina Guest.

```

1 # If the hostonly interface vboxnet0 does not exist already.
2 $ VBoxManage hostonlyif create
3 # Configure vboxnet0.
4 $ VBoxManage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1 --netmask 255.255.255.0

```

Listato 5.5. Comandi per risolvere il Troubleshooting sulla vboxnet

Analisi dei malware

In questo capitolo verrà mostrato l'utilizzo di Cuckoo sandbox per l'analisi automatica dei malware.

6.1 Cuckoo submit

Cuckoo ci permette di effettuare le analisi attraverso il comando `cuckoo submit`. Nel momento in cui si farà la submit, Cuckoo avvierà la macchina virtuale Guest caricando lo snapshot1, ed eseguirà l'analisi malware dentro ad esso. Alla fine di questo processo, la macchina Guest invierà alla macchina host le varie informazioni per la generazione del report.

Il comando `cuckoo submit`, come mostrato in Figura 6.1, ha diverse opzioni ammissibili.

È possibile specificare più directory o file contemporaneamente durante l'impostazione del comando `cuckoo submit`.

```
1 # esegue la submit ad un URL.
2 $ cuckoo submit --url http://www.example.com
3 # esegue la submit ad un local binary.
4 $ cuckoo submit /path/to/binary
5 # esegue la submit ad un local binary impostandolo come priorità alta.
6 $ cuckoo submit --priority 5 /path/to/binary
7 # esegue la submit ad un local binary, impostando un'analisi di 60 secondi.
8 $ cuckoo submit --timeout 60 /path/to/binary
9 # esegue la submit ad un local binary attraverso uno specifico pacchetto.
10 $ cuckoo submit --package <name of package> /path/to/binary
11 # esegue la submit ad un local binary specificando il route.
12 $ cuckoo submit -o route=tor /path/to/binary
13 # esegue la submit inviando il file local binary da eseguire su una macchina virtuale cuckoo specifica.
14 $ cuckoo submit --machine cuckoo1 /path/to/binary
15 # esegue la submit ad un local binary specificando la piattaforma della macchina su cui deve essere eseguita.
16 $ cuckoo submit --platform windows /path/to/binary
17 # esegue la submit di un local binary ed il dump di memoria completa della macchina che ha effettuato l'analisi.
18 $ cuckoo submit --memory /path/to/binary
```

Listato 6.1. Esempi di applicazione del comando `cuckoo submit`

Nel Listato 6.1 si mostrano degli esempi di comandi eseguibili direttamente da terminale.

Cuckoo, per tenere traccia degli invii, utilizza un ORM Python denominato SQLAlchemy; permettendo alla sandbox di utilizzare sistemi di database SQL come SQLite, MySQL, MariaDB, PostgreSQL, etc.

```

$ cuckoo submit --help
Usage: cuckoo submit [OPTIONS] [TARGET]...

Submit one or more files or URLs to Cuckoo.

Options:
-u, --url           Submitting URLs instead of samples
-o, --options TEXT  Options for these tasks
--package TEXT     Analysis package to use
--custom TEXT      Custom information to pass along this task
--owner TEXT       Owner of this task
--timeout INTEGER  Analysis time in seconds
--priority INTEGER Priority of this task
--machine TEXT     Machine to analyze these tasks on
--platform TEXT    Analysis platform
--memory           Enable memory dumping
--enforce-timeout  Don't terminate the analysis early
--clock TEXT       Set the system clock
--tags TEXT        Analysis tags
--baseline         Create baseline task
--remote TEXT      Submit to a remote Cuckoo instance
--shuffle          Shuffle the submitted tasks
--pattern TEXT     Provide a glob-pattern when submitting a
                  directory
--max INTEGER      Submit up to X tasks at once
--unique           Only submit samples that have not been
                  analyzed before
-d, --debug        Enable verbose logging
-q, --quiet        Only log warnings and critical messages
--help            Show this message and exit.
    
```

Figura 6.1. Opzioni di cuckoo submit

Un Object-Relational Mapper (ORM) è una libreria che permette di automatizzare il trasferimento dei dati archiviati nelle tabelle di database relazionali in oggetti che verranno poi utilizzati direttamente nel codice dell'applicazione.

Nella Figura 6.2 viene rappresentato il concetto appena espresso.

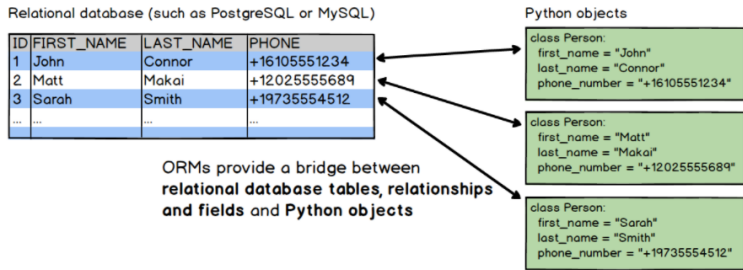


Figura 6.2. Funzionamento di ORM

Gli ORM sono molto utili, in quanto forniscono un'astrazione di alto livello su un database relazionale; questa consente ad uno sviluppatore di scrivere codice Python, anziché SQL, per le operazioni di creazione, lettura, aggiornamento ed eliminazione dei dati nel database. Gli sviluppatori possono utilizzare così il linguaggio di programmazione che desiderano, per lavorare con un database, invece di scrivere istruzioni SQL.

Il Listato 6.2 mostra le query con e senza un ORM; tali istruzioni SQL recuperano ogni riga della tabella `USERS` in cui la zip code column è `94107`.

```

1 # La query senza ORM.
2 SELECT * FROM USERS WHERE zip_code=94107;
3 # La query con Django ORM equivalente in Python.
4 users = Users.objects.filter(zip_code=94107)

```

Listato 6.2. Esempio di query con e senza un ORM

Gli ORM consentono di far muovere un'applicazione tra vari database relazionali. Ad esempio, uno sviluppatore potrebbe utilizzare SQLite per lo sviluppo locale e MySQL in fase di produzione. Tuttavia, è meglio utilizzare per lo sviluppo locale lo stesso database utilizzato nella produzione.

Gli svantaggi di un ORM sono:

- *Impedance mismatch*: la mancata corrispondenza di impedenza è un termine generico utilizzato per le difficoltà che si verificano quando si spostano i dati tra le tabelle relazionali e gli oggetti dell'applicazione. Ciò avviene perché il modo in cui uno sviluppatore utilizza gli oggetti è diverso dal modo in cui i dati vengono archiviati e uniti nelle tabelle relazionali. Infatti, il paradigma orientato agli oggetti si basa su vari principi di ingegneria del software. Il paradigma relazionale, invece, si basa su altri comprovati principi matematici, e poiché i paradigmi sottostanti sono diversi, le due tecnologie non riescono ad avere un matching perfetto.

La discrepanza tecnica può essere superata garantendo una buona formazione ai membri del team riguardo entrambe le tecnologie.

- *Performance ridotta*: una delle preoccupazioni associate a qualsiasi astrazione o framework di livello superiore è la potenziale riduzione delle prestazioni. Con gli ORM, il miglioramento delle prestazioni deriva dalla traduzione del codice dell'applicazione in un'istruzione SQL corrispondente, che, però, potrebbe non essere ottimizzata correttamente. In progetti di grandi dimensioni, gli ORM sono più efficaci rispetto al codice SQL nell'80-90% dei casi d'uso, ma nel 10-20% non lo sono, e in questi casi c'è bisogno di un amministratore di database esperto che scriva le istruzioni SQL ottimizzate, per poi sostituirle al codice SQL generato dall'ORM.
- *Spostamento della complessità del database nel codice dell'applicazione*: il codice per lavorare con i dati di un'applicazione deve risiedere da qualche parte. Prima che gli ORM fossero comuni, le stored procedure del database venivano utilizzate per incapsulare la logica del database stesso. Con un ORM il codice di manipolazione dei dati risiede, invece, nella base del codice Python dell'applicazione. L'aggiunta della logica di gestione dei dati nel codice base, in genere, non è un problema se si fa una solida progettazione dell'applicazione; tuttavia, esso porta ad un aumento della quantità totale di codice Python.

Cuckoo utilizza SQLAlchemy, un ORM Python molto buono in quanto riesce ad ottenere un livello di astrazione adeguato, rendendo anche le query più facili da scrivere.

Cuckoo è progettato per essere facilmente integrato in soluzioni più grandi e per essere completamente automatizzato.

Per automatizzare l'invio dell'analisi si utilizza l'interfaccia API REST; tuttavia, nel caso in cui si desideri scrivere il proprio script in Python, è possibile utilizzare anche le funzioni `add_path()` e `add_url()`.

Descriviamo i parametri di `add_path()`:

- `file_path (string)`: è il percorso del file da inviare.
- `timeout (integer)`: è la durata massima (in secondi) dell'analisi.
- `package (string or None)`: è il pacchetto di analisi che si desidera utilizzare per analizzare il file.
- `options (string or None)`: è l'elenco di opzioni da passare al pacchetto di analisi (nel formato `key=value, key=value`).
- `priority (integer)`: è la rappresentazione numerica della priorità da assegnare al file da analizzare (1 è basso, 2 medio, 3 alto).
- `custom (string or None)`: è il valore custom da inviare, ed è riutilizzabile durante il processing o il reporting.
- `owner (string or None)`: è il task owner.
- `machine (string or None)`: è l'identificatore della macchina virtuale che si desidera utilizzare; se non viene specificato nessun indicatore, ne verrà selezionato uno automaticamente.
- `platform (string or None)`: è la piattaforma del sistema operativo su cui si desidera eseguire quella specifica analisi (attualmente solo Windows).
- `tags (string or None)`: è il tag per la selezione della macchina.
- `memory (True or False)`: serve per generare un dump di memoria completo della macchina utilizzata per l'analisi.
- `enforce_timeout (True or False)`: è impostato a `true` per forzare l'analisi del malware per tutta la durata del timeout.
- `clock (string or None)`: fornisce un orario personalizzato da impostare nella macchina utilizzata per l'analisi.

Il return type è `integer`.

Nel Listato 6.3 si mostra un esempio di utilizzo di `add_path()`.

```

1 >>> from cuckoo.core.database import Database
2 >>> db = Database()
3 >>> db.add_path("/tmp/malware.exe")
4 1

```

Listato 6.3. Esempio di utilizzo `add_path()`

Descriviamo anche i parametri di `add_url()`:

- `url (string)`: è l'URL da analizzare.
- `timeout (integer)`: è la quantità massima di secondi per cui eseguire l'analisi.
- `package (string or None)`: è il pacchetto di analisi che si desidera utilizzare per l'URL.
- `options (string or None)`: è l'elenco di opzioni da passare al pacchetto di analisi (nel formato `key=value, key=value`).
- `priority (integer)`: è la rappresentazione numerica della priorità da assegnare all'URL (1 è basso, 2 medio, 3 alto).
- `custom (string or None)`: è il valore custom da inviare, ed è riutilizzabile durante il processing o il reporting.

- `owner` (`string` or `None`): è il task owner.
- `machine` (`string` or `None`): è l'identificatore della macchina virtuale che si desidera utilizzare; se non viene specificato nessun indicatore, ne verrà selezionato uno automaticamente.
- `platform` (`string` or `None`): è la piattaforma del sistema operativo su cui si desidera eseguire quella specifica analisi (attualmente solo Windows).
- `tags` (`string` or `None`): è il tag per la selezione della macchina.
- `memory` (`True` or `False`): serve per generare un dump di memoria completo della macchina utilizzata per l'analisi.
- `enforce_timeout` (`True` or `False`): è impostato su `True` per forzare l'analisi del malware per tutta la durata del timeout.
- `clock` (`string` or `None`): fornisce un orario personalizzato da impostare nella macchina utilizzata per l'analisi.

Il return type è `integer`.

Nel Listato 6.4 si mostra un esempio di utilizzo di `add_url()`.

```

1 >>> from cuckoo.core.database import Database
2 >>> db = Database()
3 >>> db.connect()
4 >>> db.add_url("http://www.cuckoosandbox.org")
5 2

```

Listato 6.4. Esempio di utilizzo `add_url()`

6.2 Web interface

Cuckoo ha un'interfaccia grafica completa sotto forma di un'applicazione Django; attraverso questa interfaccia, si riesce ad inviare file e URL in analisi, vedere la history delle analisi fatte, e molto altro ancora.

L'interfaccia web estrae i dati da un database MongoDB; per fare ciò si deve abilitare nel file `reporting.conf` l'opzione relativa ad esso; altrimenti, si genererà un'eccezione durante l'avvio.

Nel path `CWD/web/local_settings.py` ci sono altre opzioni di configurazione; in particolar modo si può abilitare la notifica degli errori tramite e-mail e configurare gli admin.

Nella Figura 6.3 viene mostrato un esempio del file `local_settings.py`.

Dalla Versione 2.0.0, la dimensione massima di caricamento è stata aumentata da 25MB a 10GB. Si tratta di un miglioramento notevole, che permette a quasi tutti i file di essere analizzati.

Nel Listato 6.5 vengono mostrati i comandi per poter avviare l'interfaccia web. Il primo comando avvia il servizio sulla porta 8000 (default) all'indirizzo `localhost`; il secondo comando, invece, permette di configurare l'interfaccia web definendo l'indirizzo e la porta da utilizzare.

La Figura 6.4 mostra l'avvio da terminale dell'interfaccia web.

```

1 $ cuckoo web runserver
2 $ cuckoo web runserver 0.0.0.0:PORT

```

```

# Copyright (C) 2013 Claudio Guarnieri.
# Copyright (C) 2014-2017 Cuckoo Foundation.
# This file is part of Cuckoo Sandbox - http://www.cuckoosandbox.org
# See the file 'docs/LICENSE' for copying permission.

import web.errors

# Maximum upload size (10GB, so there's basically no limit).
MAX_UPLOAD_SIZE = 10 * 1024 * 1024 * 1024

# Override default secret key stored in $PWD/web/.secret_key
# Make this unique, and don't share it with anybody.
# SECRET_KEY = "YOUR_RANDOM_KEY"

# Language code for this installation. ALL choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

# Allow verbose debug error message in case of application fault.
# It's strongly suggested to set it to False if you are serving the
# web application from a web server front-end (i.e. Apache).
DEBUG = False
DEBUG404 = False

```

Figura 6.3. File local_settings.py

Listato 6.5. Comandi disponibili per avviare il servizio di interfaccia web

```

fengge@babel:~$ cd /usr/local/lib/python2.7/dist-packages/sflock/decode/office.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
from cryptography.hazmat.backends import default_backend
/usr/local/lib/python2.7/dist-packages/sflock/decode/office.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
from cryptography.hazmat.backends import default_backend
Performing system checks...

System check identified no issues (0 silenced).
January 25, 2022 - 17:25:01
Django version 1.8.4, using settings 'cuckoo.web.web.settings'
Starting development server at http://0.0.0.0:8081/
Quit the server with CONTROL-C.

```

Figura 6.4. Avvio da terminale dell'interfaccia web

6.3 Cuckoo malware analysis

Aperto un browser all'indirizzo <http://0.0.0.0:8081/> si aprirà una pagina con una dashboard di Cuckoo.

Come mostrato nella Figura 6.5, possiamo mettere in analisi dei file presenti sulla macchina host, degli URL o degli Hash. Si ha, anche, una panoramica generica delle ultime analisi fatte, delle informazioni sul sistema, e della versione di Cuckoo.

Facciamo un esempio di analisi relativo ad un URL. Si inserisce, nel relativo spazio, l'indirizzo da testare; nel nostro caso, testeremo il link <https://roja.directa.live/> come mostrato in Figura 6.6. Nella Figura 6.7 possiamo vedere, invece, come cuckoo alla fine dell'analisi lo classifica dando ad esso uno score pari a 5.2.

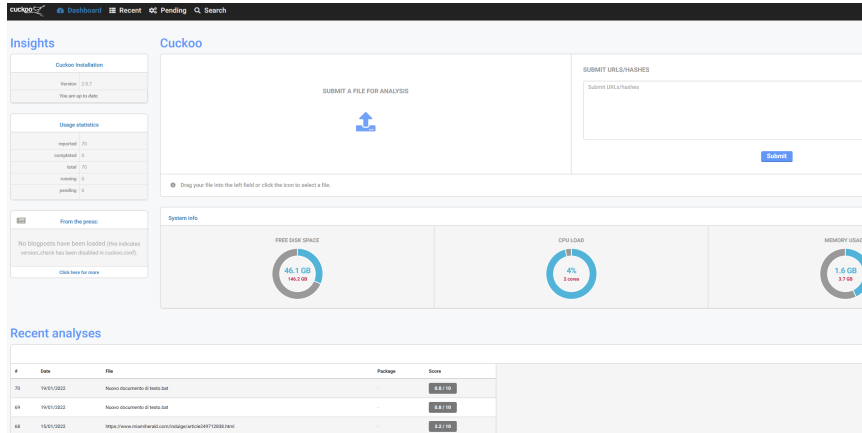


Figura 6.5. Pagina iniziale di Cuckoo

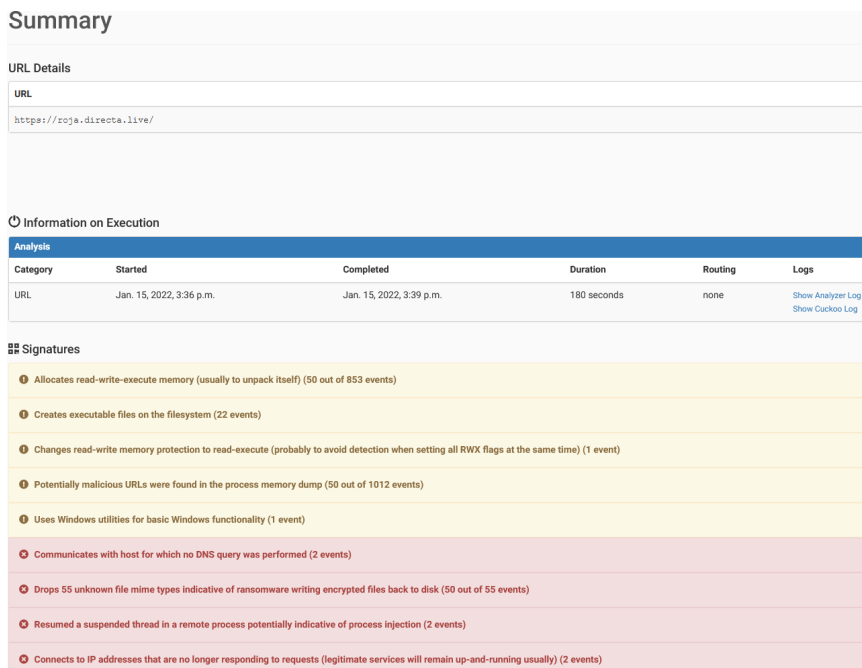


Figura 6.6. Pagina web di Cuckoo con i risultati dell'analisi URL

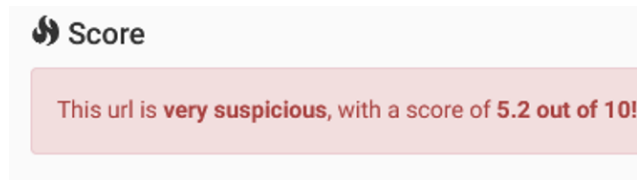


Figura 6.7. Score dell'analisi URL

Ci sono varie opzioni da inserire prima di avviare la submit. Queste sono:

- **free**: se abilitato, non verrà prodotto alcun log sui comportamenti del malware ed esso verrà eseguito liberamente.
- **procmemdump**: se abilitato, prende i dump di memoria di tutti i processi monitorati attivamente.
- **human**: se abilitato, simulerà anche l'interazione umana (i movimenti del mouse, etc.).

Si mostra, di seguito, l'elenco dei pacchetti di Cuckoo esistenti e utilizzabili per l'analisi:

- **applet**: utilizzato per analizzare gli applet Java; esso richiede di specificare il nome della classe da eseguire.
- **bin**: utilizzato per analizzare dati binari generici, come le shellcode.
- **cpl**: utilizzato per analizzare gli applet del pannello di controllo.
- **dll**: utilizzato per eseguire e analizzare le librerie collegate dinamicamente. Per questo comando è necessario specificare:
 1. La funzione da eseguire; se non ne viene specificata nessuna, Cuckoo proverà ad eseguire `DllMain`.
 2. Gli argomenti da passare alla DLL tramite riga di comando.
 3. Un nome di processo da utilizzare per falsificare il nome del programma di avvio della DLL in modo tale da non far vedere il vero nome, ovvero `rundll32.exe` (utilizzato per ingannare possibili trucchi anti-sandboxing di determinati malware).
- **doc**: utilizzato per eseguire e analizzare documenti Microsoft Word.
- **exe**: pacchetto di analisi predefinito utilizzato per analizzare eseguibili Windows generici.
- **generic**: utilizzato per eseguire e analizzare campioni generici tramite `cmd.exe`.
- **ie**: utilizzato per analizzare il comportamento di Internet Explorer all'apertura dell'URL o del file HTML specificato.
- **jar**: utilizzato per analizzare i contenitori JAR di Java; è necessario specificare il percorso della classe da eseguire; se non ne viene specificato nessuno, Cuckoo tenterà di eseguire la funzione principale specificata nel file `MANIFEST` di JAR.
- **js**: utilizzato per eseguire e analizzare file Javascript (ad esempio, i file che si trovano negli allegati delle e-mail).
- **hta**: utilizzato per eseguire e analizzare i file dell'applicazione HTML.
- **msi**: utilizzato per eseguire e analizzare il programma di installazione di Windows MSI.
- **pdf**: utilizzato per eseguire e analizzare documenti PDF.
- **ppt**: utilizzato per eseguire e analizzare documenti Microsoft PowerPoint.
- **ps1**: utilizzato per eseguire e analizzare gli script di PowerShell.
- **python**: utilizzato per eseguire e analizzare gli script Python.
- **vbs**: utilizzato per eseguire e analizzare i file Script.
- **wsf**: utilizzato per eseguire e analizzare i file Windows Script Host.
- **xls**: utilizzato per eseguire e analizzare documenti Microsoft Excel.
- **zip**: utilizzato per eseguire e analizzare archivi Zip. Per questo comando è necessario specificare:

1. Il nome del file contenuto nell'archivio da eseguire; se non ne viene specificato nessuno, Cuckoo proverà ad eseguire `sample.exe`.
2. Gli argomenti della riga di comando al processo iniziale del malware inviato.
3. La password dell'archivio; se non ne viene specificato nessuna, Cuckoo cercherà di estrarre l'archivio senza password.

Cuckoo, essendo del tutto personalizzabile, può anche creare dei nuovi pacchetti di analisi.

Per specificare il pacchetto dell'analisi bisogna utilizzare il comando nel Listato 6.6. Se non ne viene specificato nessuno, Cuckoo proverà a rilevare il tipo di file e a selezionare il pacchetto di analisi corretto. Se il tipo di file non è supportato, per impostazione predefinita, l'analisi verrà interrotta.

```
1 $ cuckoo submit --package <package name> /path/to/malware
```

Listato 6.6. Comando per avviare l'analisi con un pacchetto specifico

6.3.1 Risultati dell'analisi

Quando l'analisi viene completata, i file relativi ad essa vengono archiviati nella sottodirectory al path `CWD/storage/analyses/`. Tale sottodirectory viene denominata in base all'ID numerico di tipo incrementale; questo rappresenta il task di una specifica analisi nel database.

Nella Figura 6.8 viene mostrato un esempio di struttura della directory.

```
.
|-- analysis.log
|-- binary
|-- dump.pcap
|-- memory.dmp
|-- files
|   |-- 1234567890_dropped.exe
|-- logs
|   |-- 1232.bson
|   |-- 1540.bson
|   `-- 1118.bson
|-- reports
|   |-- report.html
|   |-- report.json
`-- shots
    |-- 0001.jpg
    |-- 0002.jpg
    |-- 0003.jpg
    `-- 0004.jpg
```

Figura 6.8. Struttura delle directory delle analisi

Descriviamo i vari file presenti:

- **analysis.log**: è un file di registro generato dall'analizzatore che contiene una traccia dell'esecuzione dell'analisi all'interno dell'ambiente Guest. Esso riporterà la creazione di processi, file ed eventuali errori occorsi durante l'esecuzione.

- `dump.pcap`: è il file dump di rete generato da `tcpdump`.
- `dump.sorted.pcap`: è una versione ordinata di `dump.pcap`, che permette all'interfaccia Web di cercare rapidamente il flusso TCP.
- `memory.dmp`: se abilitato, contiene il dump della memoria completa della macchina di analisi.
- `files/`: in questa directory ci sono tutti i file su cui operava il malware durante l'esecuzione e che Cuckoo ha scaricato.
- `files.json`: è un file contenente una voce con codifica JSON per ogni file eliminato disponibile (ad esempio, tutti i file in `files/`, `shots/`, e così via); esso, inoltre, contiene meta informazioni, ove disponibili.
- `logs/`: è la directory contenente tutti i registri grezzi generati dal monitoraggio del processo di Cuckoo.
- `reports/`: è una directory contenente tutti i report generati da Cuckoo.
- `shots/`: è una directory contenente tutti gli screenshot del desktop della macchina Guest, presi durante l'esecuzione del malware. Un esempio della generazione degli screenshot da parte di Cuckoo è rappresentato nella Figura 6.9.
- `tlsmaster.txt`: è un file contenente le informazioni segrete su TLS che sono state acquisite durante l'analisi. Le informazioni segrete in TLS possono essere utilizzate per decriptare il traffico SSL/TLS e, quindi, anche i flussi HTTPS.

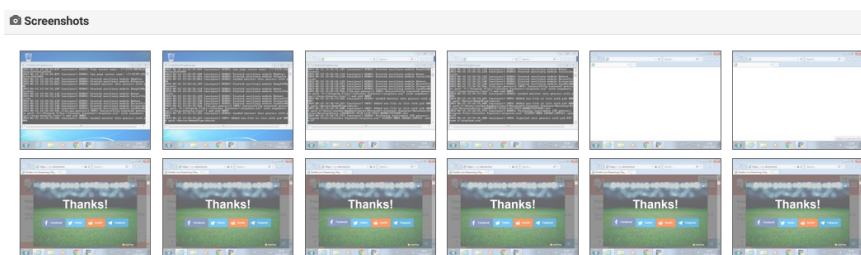


Figura 6.9. Screenshot generati dall'analisi fatta da Cuckoo

Nella Figura 6.10 viene mostrato un primo piano di screenshot fatto da Cuckoo durante l'analisi dell'URL.

La classificazione dei malware è fatta attraverso diversi strumenti:

- *Yara*: è uno strumento utilizzato per classificare i malware; esso permette di creare delle descrizioni di famiglie di malware, basate su testo o pattern binari, dette regole. *Yara* è stato originariamente sviluppato da *Victor Alvarez* di *Virustotal* ed è utilizzato principalmente nella ricerca e nel rilevamento di malware.
- *Mitre Attack*: è un database di tattiche di attacco, realizzato mediante osservazione di scenari di attacchi reali. Esso è una rappresentazione abbastanza completa dei comportamenti che gli utenti malintenzionati utilizzerebbero per compromettere le reti.
- *Virus Total*: è un sito online che permette di fare analisi sui file, URL, etc.

Attualmente il modulo di classificazione di Cuckoo è ancora in fase Alpha; perciò, molte volte, la classificazione potrebbe restituire uno score non adeguato, sottostimando o sovrastimando il punteggio reale.

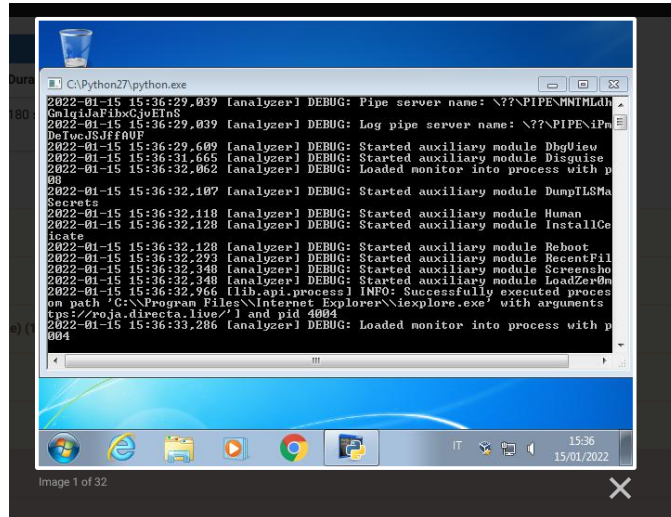


Figura 6.10. Screenshot di esempio dell’analisi URL generato da Cuckoo

Cuckoo contiene un bug che permette di classificare un malware con uno score superiore a 10.

Nelle Figure 6.11 - 6.19 si mostrano le sezioni che descrivono in maniera generale i risultati dell’analisi URL.

Signatures

Allocates read-write-execute memory (usually to unpack itself) (50 out of 853 events)

Time & API	Arguments
NtProtectVirtualMemory Jan. 15, 2022, 3:36 p.m.	process_identifier: 4004 stack_dep_bypass: 0 stack_pivoted: 0 heap_dep_bypass: 0 length: 4096 protection: 64 (PAGE_EXECUTE_READWRITE) base_address: 0x7724e000 process_handle: 0xffffffff
NtProtectVirtualMemory Jan. 15, 2022, 3:36 p.m.	process_identifier: 4004 stack_dep_bypass: 0 stack_pivoted: 0 heap_dep_bypass: 0 length: 4096 protection: 64 (PAGE_EXECUTE_READWRITE) base_address: 0x7724e000 process_handle: 0xffffffff
NtProtectVirtualMemory Jan. 15, 2022, 3:36 p.m.	process_identifier: 4004 stack_dep_bypass: 0 stack_pivoted: 0 heap_dep_bypass: 0 length: 4096 protection: 64 (PAGE_EXECUTE_READWRITE) base_address: 0x7724e000 process_handle: 0xffffffff

Figura 6.11. Report dell’analisi URL (Parte 1)

❗ Creates executable files on the filesystem (22 events)	
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\VRM4MFTT\custom-messages.5799ddf75a30812a3d49[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\461D7XGS\js[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\0RHMBIHD>tag[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\2S3LXBUR\bootstrap.min[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\2S3LXBUR\jquery-ui.min[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\461D7XGS\sdk[2].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\2S3LXBUR\analytics[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\0RHMBIHD\shares[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\2S3LXBUR\addthis_widget[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\0RHMBIHD\sdk[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\461D7XGS\layers.fa6cd1947ce26e890d3d[1].js
file	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\461D7XGS\jquery.scrolling-tabs[1].js

Figura 6.12. Report dell'analisi URL (Parte 2)

❗ Changes read-write memory protection to read-execute (probably to avoid detection when setting all RWX flags at the same time) (1 event)	
Time & API	Arguments
NtProtectVirtualMemory Jan. 15, 2022, 3:36 p.m. 🌐	process_identifier: 892 stack_dep_bypass: 0 stack_pivoted: 0 heap_dep_bypass: 1 length: 4096 protection: 16 (PAGE_EXECUTE) base_address: 0x045b0000 process_handle: 0xffffffff

Figura 6.13. Report dell'analisi URL (Parte 3)

❗ Potentially malicious URLs were found in the process memory dump (50 out of 1012 events)	
url	http://certificates.starfieldtech.com/repository/1604
url	http://www.expedia.com/favicon.ico
url	http://uk.ask.com/favicon.ico
url	http://www.priceminister.com/
url	http://cr13.digicert.com/ssca-sha2-g6-1.cr101
url	http://cr14.digicert.com/DigiCertTLRSASHA2562020CA1-4.cr10
url	https://mc.yandex.ru/metrika/tag.js
url	http://www.iask.com/favicon.ico
url	https://www.reddit.com/
url	http://weheartit.com/
url	http://www.merlin.com.pl/favicon.ico

Figura 6.14. Report dell'analisi URL (Parte 4)

🚩 Uses Windows utilities for basic Windows functionality (1 event)	
cmdline	"C:\Program Files\Internet Explorer\iexplore.exe" SCODEF:4004 CREDAT:209921 /prefetch:2

Figura 6.15. Report dell'analisi URL (Parte 5)

🚩 Communicates with host for which no DNS query was performed (2 events)	
host	23.220.255.17
host	23.220.255.6

Figura 6.16. Report dell'analisi URL (Parte 6)

🚩 Drops 55 unknown file mime types indicative of ransomware writing encrypted files back to disk (50 out of 55 events)	
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\Content\F2DDC02B5F37625B82E81F4976CEE400_A01EFC9EF87B331821A80D893F4D7FE8
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\620BEF1064BD8E252C59957B3C91896
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\Content\620BEF1064BD8E252C59957B3C91896
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\Content\75CA58072B9926F763A91F0CC2798706_93E4B2BA79A897B3100CCB272F03BF4F
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\Content\68FAF71AF355126BCA00CE2E73CC7374_77B682CF3AAC7B00161DFFF7DEA4CC8C
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\A3E6546D43CF34D85B14CC51DAFA332
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\Content\F07644E38ED7C9F37D11EEC6D4335E02_1160E11B9377D569BC114C731E94B72F
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\Content\265C0DEB29181DD1891051371C5F963A_B0CC424E58EE93AF3ACDE89D8BDEDD
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\35DDED268117918D1D277A171D8DF7B_CE500F4904CEE254834ABDBE94442DC2
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\Content\DA67B12F4C0184D0A1ED54E3B43E7FCA
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\Content\70DAE932E3BC3C00656A27B544BA9CA
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\A16C6C16D94F76E0808C087DFC657D99_E5B132B41B26E2FD23A912C0CB5FBCBA
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\000F7F8FAB2D96E6F8CB5C9A3B4EC90
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\A16C6C16D94F76E0808C087DFC657D99_E0990A7CF057A22E5C656F7713BE4EB4
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\6A2279C2CA42EBEE26F14589F0738E50
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\F07644E38ED7C9F37D11EEC6D4335E02_1160E11B9377D569BC114C731E94B72F
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\1B1F4BA66CDBFEC85A20E11BF729AF23_AA85F8F9DAFF33153B5AEC2E983B94B6
file	C:\Users\cuckoo\AppData\LocalLow\Microsoft\CryptnetUrlCache\MetaData\E887E036775F4159E2816B7B9E527E5F_ADE58CF9589C2D8854882E94BDA0990

Figura 6.17. Report dell'analisi URL (Parte 7)

🚩 Resumed a suspended thread in a remote process potentially indicative of process injection (2 events)	
Time & API	Arguments
Process injection	Process 4004 resumed a thread in remote process 892
NtResumeThread	thread_handle: 0x00000468 suspend_count: 1
Jan. 15, 2022, 3:36 p.m. 🕒	process_identifier: 892

Figura 6.18. Report dell'analisi URL (Parte 8)

🚩 Connects to IP addresses that are no longer responding to requests (legitimate services will remain up-and-running usually) (2 events)	
dead_host	23.220.255.6:80
dead_host	23.220.255.17:80

Figura 6.19. Report dell'analisi URL (Parte 9)

Per vedere in maniera più specifica quello che è successo durante l'analisi, possiamo andare nelle varie sezioni, a sinistra, come mostrato nella Figura 6.20. I report completi sono mostrati nelle Figure 6.21, 6.22 e 6.23.

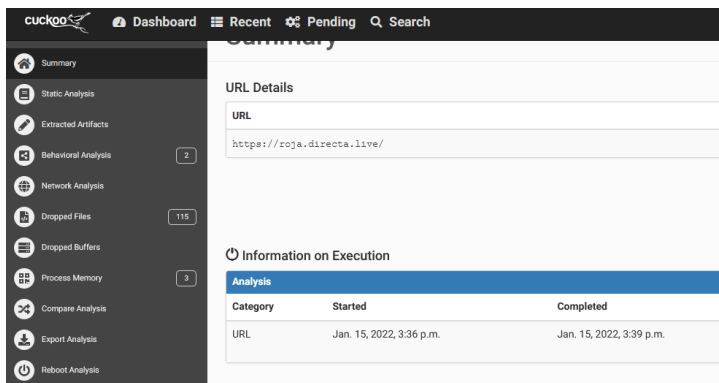


Figura 6.20. Elenco delle sezioni con le informazioni complete dell'analisi

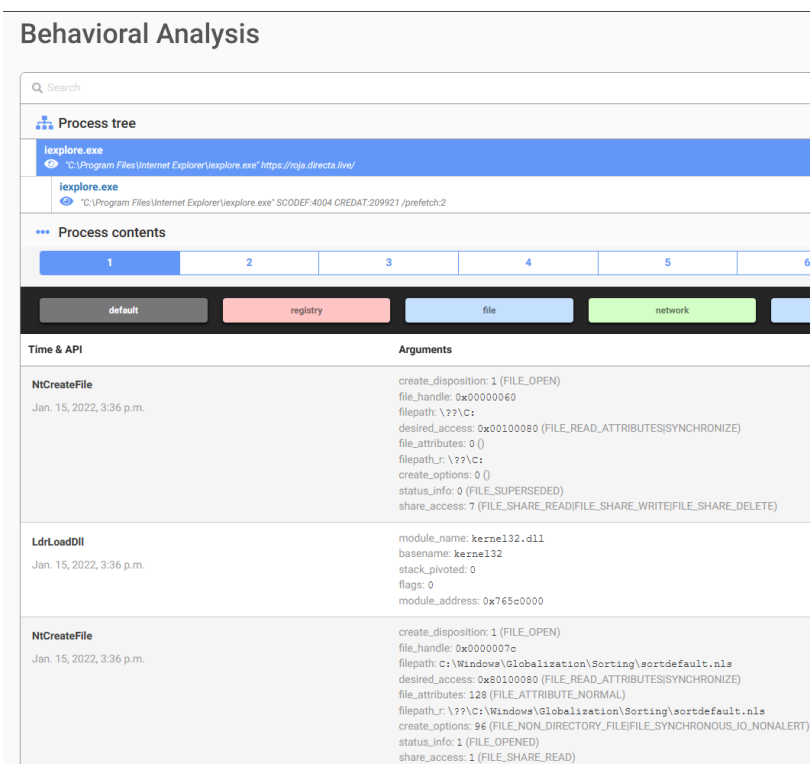


Figura 6.21. Report specifico dell'analisi URL (Parte 1)

Dropped Files	
Name	7b6bfa13f0778c40_sh.f48a1a04fe8dbf021b4cda1d[1].htm Download Submit file
Filepath	C:\Users\cuckoo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\VRM4MFTT\sh.f48a1a04fe8dbf021b4cda1d[1].htm
Size	70.7KB
Processes	892 (iexplore.exe)
Type	HTML document, ASCII text, with very long lines
MDS	d1e54f0011efab67f5d299e62ce41c69
SHA1	b518bb4fef90af133240c8e6efc08f4b3b74c5ad
SHA256	7b6bfa13f0778c40bb2a00af9819bea2f07aEcb4d071e7e4f436196953a50b4d
CRC32	36b69cef
ssdeep	None
Yara	None matched
VirusTotal	Search for analysis
Name	9162b15f464c24cb_e03mh9fb.htm Download Submit file

Figura 6.22. Report specifico dell'analisi URL (Parte 2)

Process Memory

Process memory dump for iexplore.exe (PID 892, dump 1)

Extracted/injected images (may contain unpacked executables)
[Download #1](#)

Process memory dump for iexplore.exe (PID 892, dump 2)

Extracted/injected images (may contain unpacked executables)
[Download #1](#)

URLs found in process memory

```

http://certificates.starfieldtech.com/repository/1604
http://www.expedia.com/favicon.ico
http://uk.ask.com/favicon.ico
http://www.priceminister.com/
http://cr13.digicert.com/ssca-sha2-g6-1.cr101
http://cr14.digicert.com/DigicertLRSASHA2562020CA1-4.cr10
https://mc.yandex.ru/metrika/tag.js
http://www.iask.com/favicon.ico
https://www.reddit.com/
http://weheartit.com/
http://www.merlin.com.pl/favicon.ico
https://adservice.google.com/bh/adsid/integrator.
http://www.cnet.com/favicon.ico

```

Figura 6.23. Report specifico dell'analisi URL (Parte 3)

Nella Figura [6.24](#) viene mostrata la sezione recent, cioè l'history delle analisi passate.





cuckoo  Dashboard  Recent  Pending  Search						
Files	URLs	Score 0 - 4	Score 4 - 7	Score 7 - 10		
70	2022-01-19 16:23	63966d9ce3af0fd41d2f0f393f8b9ef	Nuovo documento di testo.bat	reported	score: 0.8	
69	2022-01-19 16:20	63966d9ce3af0fd41d2f0f393f8b9ef	Nuovo documento di testo.bat	reported	score: 0.8	
68	2022-01-15 16:51	-	https://www.miamiherald.com/indulge/article249712838.html	reported	score: 3.2	
67	2022-01-15 16:47	-	https://www.animesatum.it/watch?file=eaLUwYuNwtXXs&server=0	reported	score: 3.2	
66	2022-01-15 16:44	-	http://www.rojdirect.org	reported	score: 3.2	
65	2022-01-15 16:41	-	http://www.cricfree.sx	reported	score: 3.2	
64	2022-01-15 16:38	-	http://www.ifeed2all.eu	reported	score: 3.2	
63	2022-01-15 16:35	-	http://www.allp2ptv.org	reported	score: 2.8	
62	2022-01-15 16:32	-	http://www.hesgoal.com	reported	score: 3.4	
61	2022-01-15 16:29	c11e25278417f985cc968c1e361a0fb0	f659b269fbe4128588f7a2fa4d6022cc74e508d28eee05c5aff26cc23b7bd1a5	reported	score: 1.4	

Figura 6.24. Sezione recent delle analisi fatte

Integrazione con la piattaforma SOAR

In questo capitolo verrà mostrata l'integrazione di Cuckoo su Cortex XSOAR.

7.1 Cortex XSOAR

Cortex XSOAR è una piattaforma di Security Orchestration, Automation e Response che consente agli esperti della cybersecurity di contrastare velocemente le minacce che colpiscono l'azienda. Cortex XSOAR è l'evoluzione della piattaforma Demisto, acquisita da Palo Alto Networks nel marzo 2019. Tutti i clienti Demisto sono stati migrati automaticamente e gratuitamente su Cortex XSOAR.

Palo Alto Networks sta ridefinendo la categoria della security orchestration, automation e response mettendo al centro della strategia il Threat Intel Management, cioè la gestione dell'intelligence sulle minacce. Quest'ultima consente alle organizzazioni di comprendere meglio il panorama globale delle minacce, anticipare le mosse degli attaccanti e agire tempestivamente per la gestione degli attacchi.

Una buona gestione dell'intelligence sulle minacce offre meccanismi di difesa proattivi contro qualsiasi minaccia. Ciò può essere ottenuto solo se i dati sulle minacce sono rilevanti, vasti, affidabili e utilizzabili.

Cortex XSOAR è la piattaforma SOAR più completa sul mercato di oggi, che orchestra centinaia di prodotti di sicurezza per aiutare i clienti SOC a standardizzare e automatizzare i loro processi, ottenendo tempi di risposta più rapidi e una maggiore produttività del team.

Cortex XSOAR permette di essere:

- *Completo*, in quanto ha oltre 750 integrazioni e oltre 680 pacchetti da utilizzare in molti casi d'uso che riguardano la security. Esso semplifica l'orchestrazione e l'automazione dei flussi di lavoro e dei processi di risposta agli incidenti.
- *Innovativo*, in quanto permette di scoprire, utilizzare e condividere le integrazioni di orchestrazione attraverso il Marketplace di Cortex.
- *Scalabile*, in quanto permette di avere tante integrazioni integrabili e configurabili facilmente. Nella Figura [7.1](#) viene mostrato un esempio di diverse integrazioni su Cortex XSOAR.

- *Centralizzato*, in quanto riesce a gestire tutti gli incidenti di sicurezza da un'unica posizione, consentendo così anche di automatizzare le attività di ticketing direttamente dal XSOAR.
- *Azionabile*, in quanto la gestione integrata delle informazioni sulle minacce, permette di collegare le informazioni relative alle minacce esterne agli incidenti in tempo reale, riducendo, così, del 90% il tempo dedicato alla gestione delle informazioni sulle minacce.
- *Smart*, in quanto è un alleato perfetto per gli analisti della sicurezza. La piattaforma è basata sull'apprendimento automatico.

Nella Figura [7.1](#) viene mostrata la sezione relativa alle integrazioni su Cortex XSOAR

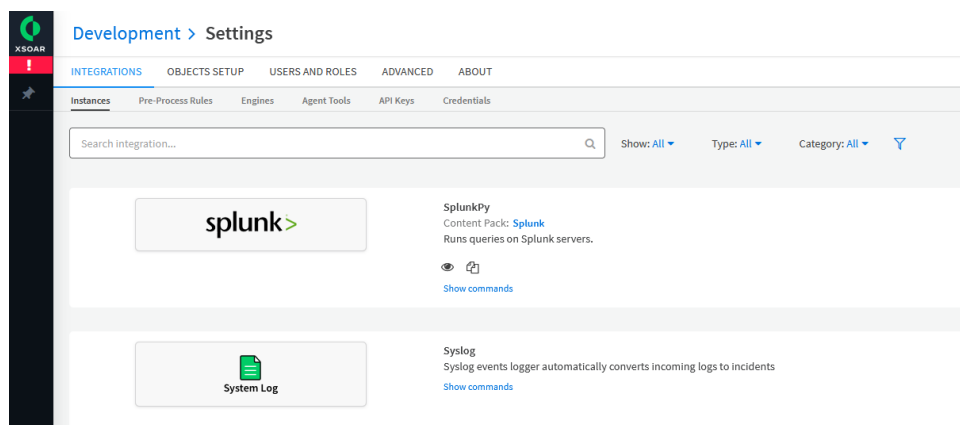


Figura 7.1. Sezione integrazioni su Cortex XSOAR

7.1.1 Integrazione di Cuckoo su Cortex XSOAR

Per integrare Cuckoo su Cortex XSOAR bisogna avviare il server API dal terminale della macchina host. Nella Figura [7.2](#) si mostra uno screenshot con il comando `cuckoo api`. Per impostazione predefinita, Cuckoo collegherà il servizio su `localhost:8090`.

Possiamo decidere, attraverso il secondo comando del Listato 7.1, di modificare i valori di default.

```
1 $ cuckoo api
2 $ cuckoo api --host 0.0.0.0 --port 1337
```

Listato 7.1. Comando per avviare il server API

Per motivi di privacy aziendale non è permesso mostrare l'indirizzo reale utilizzato.

Dopo aver avviato il server API (Figura [7.2](#)), bisogna accedere al portale di Cortex XSOAR ed installare direttamente dal Marketplace l'integrazione di Cuckoo.


```

x-fenggebelducci-virtual-machine:~$ cuckoo api
/usr/local/lib/python2.7/dist-packages/sflock/decode/office.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.hazmat.backends import default_backend
* Serving Flask app "cuckoo.apps.api" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
2022-01-25 17:36:06,053 [werkzeug] INFO: * Running on http://localhost:8090/ (Press CTRL+C to quit)

```

Figura 7.2. Avvio del server API

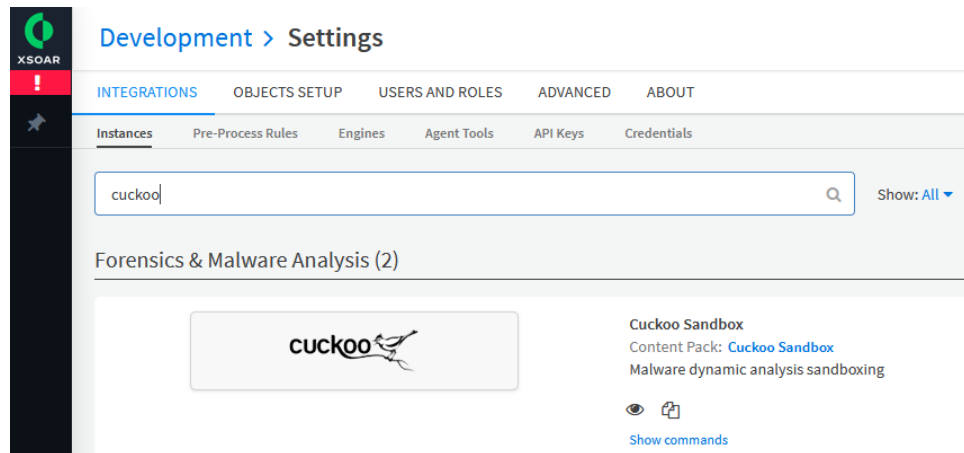


Figura 7.3. Integrazione di Cuckoo installata su Cortex XSOAR

Nella Figura 7.3 si mostra l'integrazione di Cuckoo installata su Cortex.

Il codice per l'integrazione di Cuckoo nella Figura 7.4 va modificato. In particolare, vanno inseriti il return con l'indirizzo della macchina di Cuckoo e la relativa porta.

Dopo aver avviato il server API ed installato l'integrazione di Cuckoo su Cortex XSOAR, bisognerà accedere alle impostazioni di Cuckoo Sandbox ed inserire la password presente nel file `cuckoo.conf` per effettuare l'autenticazione attraverso API token. La Figura 7.5 mostra il form delle impostazioni di Cuckoo Sandbox.

A questo punto, se l'integrazione è andata a buon fine, apparirà un messaggio di successo, come mostrato nella Figura 7.6.

Nella Figura 7.7 si possono vedere alcuni comandi di Cuckoo, applicabili direttamente su Cortex XSOAR.

Infine, si esegue il comando del Listato 7.1 per visualizzare tutti i task di Cuckoo nella War Room.

I risultati del comando vengono mostrati nella Figura 7.8.

```
1 !cuckoo-list-tasks
```

Listato 7.2. Comando per visualizzare tutti i task di Cuckoo

```
60
61 var undrscrToCamelCase = function(string){
62   string = '_' + string;
63   return string.replace(/_([a-z])/g, function (g) { return g[1].toUpperCase(); });
64 };
65
66 //returns single object withing entity (i.e. File[0])
67 var jsonToEntityObject = function(origObj, newKeys){
68   var ret = {};
69   var path;
70   var newField;
71   for(var key in newKeys){
72     if(newKeys[key]){
73       ret[newKeys[key]] = dq(origObj, '.'+key);
74     }
75   }
76   return ret;
77 };
78
79 //returns entire entity array (i.e. File)
80 var jsonToEntity = function(origObj, newKeys){
81   var j;
82   var ret;
83   if(!Array.isArray(origObj)){
84     ret = [jsonToEntityObject(origObj, newKeys)];
85     return ret;
86   }
87   else if(origObj.length > 0){ //makes sure no empty arrays are pushed
88     ret = [];
89     for(j=0; j<origObj.length; j++){
90       ret.push(jsonToEntityObject(origObj[j], newKeys));
91     }
92     return ret;
93   }
94 };
95
96 var fixUrl = function(base) {
97   res = base;
98   if (base && base[base.length - 1] != '/') {
99     res = res + '/';
100  }
101  return "http://[REDACTED]:8090/";
102 };
103
```

Figura 7.4. Codice dell'integrazione di Cuckoo

Cuckoo Sandbox

Instance Settings

Name *
Cuckoo Sandbox_instance_1

Server URL (e.g. https://192.168.0.1)
http://[REDACTED]

[Switch to credentials](#)

Username (Only if your cuckoo service requires HTTP auth)
_token

Password

Trust any certificate (not secure)

Use system proxy settings

Do not use by default

Log Level: Off ▾

Run on
Single engine: No engine ▾

Delete

Authenticate with an API token

[Help](#) [Test results](#)

In new Cuckoo installations, a random API token is automatically generated for you, and located in the cuckoo.conf file. In order to authenticate with an API token, insert '___token' in the "Username" textbox, and the token itself in the "Password" textbox.

[View Integration Documentation](#)

Figura 7.5. Impostazioni di Cuckoo Sandbox su Cortex XSOAR

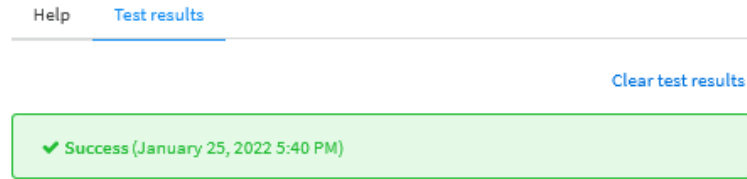


Figura 7.6. Messaggio di successo per l'avvenuta integrazione di Cuckoo su Cortex XSOAR

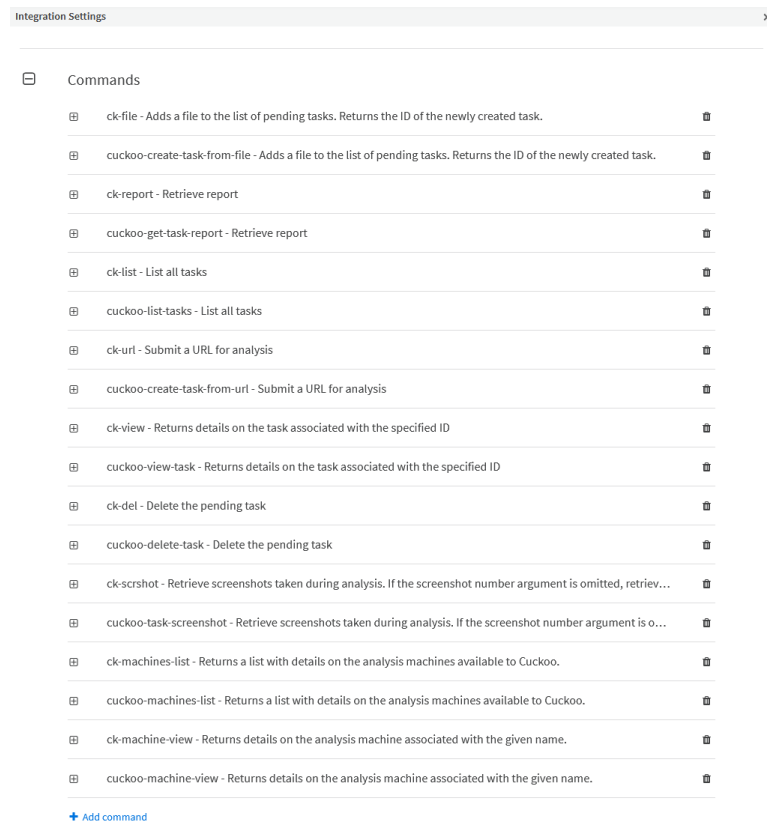


Figura 7.7. Comandi eseguibili su Cortex XSOAR

ID	Category	Date	Completion	Status	Duration	Action	Error	Host
2020-01-01 12:02:46	url	2020-01-01 12:02:46	2020-01-01 12:03:37	100	1:01	None		D:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\...
2020-01-01 12:08:13	url	2020-01-01 12:08:13	2020-01-01 12:08:19	100	0:06	None		D:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\...
2020-01-01 12:11:19	file	2020-01-01 12:11:19	2020-01-01 12:11:33	95	0:14	None		D:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\...
2020-01-01 12:18:00	file	2020-01-01 12:18:00	2020-01-01 12:18:41	100	0:41	None		D:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\... C:\Program Files\Cuckoo\Tasking\...

Figura 7.8. Lista dei task di Cuckoo

Conclusioni

In questo lavoro di tesi sono state presentate la progettazione e la realizzazione di una sandbox open source per l'analisi automatica dei malware e l'integrazione di essi sulla piattaforma SOAR di Cybertech.

Nella prima parte della tesi abbiamo descritto che cosa sia la cybersecurity e abbiamo analizzato i principali attacchi cyber noti a livello globale, descrivendone la loro evoluzione negli ultimi anni.

Abbiamo, poi, fornito una panoramica generale dell'azienda Cybertech, del concetto di SOC e di SOAR, oltre che del progetto di sandbox proposto per la presente tesi.

In seguito abbiamo descritto l'installazione di Cuckoo sui sistemi operativi Ubuntu e CentOS.

Successivamente abbiamo testato il funzionamento del sistema facendo delle analisi per individuare la presenza di malware sia sui file che sugli URL.

In tale attività, abbiamo visto che il modulo di classificazione non riesce ancora a definire bene lo score di un malware; infatti, a volte, esso assegna ad un determinato malware anche uno score superiore a 10, il suo limite massimo.

Infine, abbiamo integrato il sistema Cuckoo Sandbox su Cortex XSOAR, cioè la piattaforma SOAR di Cybertech.

Cuckoo, essendo personalizzabile in tutti i suoi aspetti, ha sicuramente un futuro prospero. Esso, essendo composto da moduli, permette di effettuare qualsiasi cosa, ad esempio creare dei nuovi package per le analisi, oppure sviluppare delle signature, che permettono di identificare dei pattern predefiniti sui risultati delle analisi.

Insomma, Cuckoo è integrabile in qualsiasi ambiente e modificabile in ogni suo aspetto.

Gli sviluppatori, nell'ultimo post datato il 19 giugno 2019, hanno comunicato la progettazione di un nuovo server per i risultati delle analisi che minimizza il consumo della CPU; oltre a questo, hanno comunicato che in futuro lavoreranno sicuramente sulla compatibilità di Cuckoo con Python 3.

Un altro miglioramento che gli sviluppatori faranno riguarderà il modulo di classificazione di Cuckoo, che attualmente, è ancora nella versione alpha.

Riferimenti bibliografici

1. Cortex XSOAR. <https://xsoar.pan.dev/>, 2022.
2. Cuckoo latest. <https://cuckoo.readthedocs.io/en/latest/>, 2022.
3. Cuckoo latest faq. <https://cuckoo.readthedocs.io/en/latest/faq/>, 2022.
4. Cuckoo latest installation. <https://cuckoo.readthedocs.io/en/latest/installation/>, 2022.
5. Cuckoo latest introduction. <https://cuckoo.readthedocs.io/en/latest/introduction/>, 2022.
6. Cuckoo latest usage. <https://cuckoo.readthedocs.io/en/latest/usage/>, 2022.
7. Cuckoo Sandbox. <https://cuckoosandbox.org/>, 2022.
8. Install mongodb on red hat. <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/>, 2022.
9. Object Relational Mappers. <https://www.fullstackpython.com/object-relational-mappers-orms.html>, 2022.
10. Palo Alto Networks. <https://www.paloaltonetworks.com/cortex/cortex-xsoar>, 2022.
11. Perché è importante un SOAR. <https://www.zerounoweb.it/techtarget/searchsecurity/i-vantaggi-degli-strumenti-soar/>, 2022.
12. Perché è importante un SOC. <https://www.matika.it/perche-soc-fondamentale-garantire-la-sicurezza-dellinfrastruttura-it/>, 2022.
13. Threats. <https://www.kaspersky.it/resource-center/threats>, 2022.
14. VirtualBox host only network. <https://precisionsec.com/virtualbox-host-only-network-cuckoo-sandbox-0-4-2/>, 2022.
15. VirtualBox manual. <https://www.virtualbox.org/manual/>, 2022.
16. What is threat intelligence management. <https://www.paloaltonetworks.com/cyberpedia/what-is-threat-intelligence-management>, 2022.
17. YARA rules. <https://blog.malwarebytes.com/security-world/technology/2017/09/explained-yara-rules/>, 2022.
18. Monnappa K A. *Learning Malware Analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware*. Packt Publishing, 2018.
19. D. Barker. *Malware Analysis Techniques: Tricks for the triage of adversarial software*. Packt Publishing, 2021.
20. G. Blokdyyk. *Security Orchestration Automation And Response A Complete Guide - 2021 Edition*. 5STARCOOKS, 2021.
21. G. Butti, D. Caivano, D. V. Cassinerio, N. Ciardi, L. Colucci, M. D'Elia, A. Di Mattia, A. Ercoli, G. Faggioli, G. Gamberi, P. Giudice, A. Ieranò, M. Lestingi, F. M. R. Livelli, G. Nencini, M. Pericò, D. Raguseo, M. Raimondi, S. Rinauro, P. L. Rotondo,

- S. Scozzari, G. M. Sgro, F. Talone, G. Tesoriere, R. Tordi, A. Vallega, A. Z. Manzoni, V. S. Barletta, and S. Boccaredelli. Rapporto Clusit Ottobre 2021. <https://clusit.it/rapporto-clusit/>, 2021.
22. D. Clinton. *Ubuntu Linux Bible*. Wiley, 2020.
 23. A. Handa, R. Negi, and S. K. Shukla. *Implementing Enterprise Cybersecurity with Open-source Software and Standard Architecture*. River Publishers, 2021.
 24. M. Helmke. *Ubuntu Linux Unleashed*. Addison-Wesley Professional, 2020.
 25. A. Honig and M. Sikorski. *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press, 2012.
 26. P. Jacobs, A. Arnab, and B. Irwin. Classification of Security Operation Centers. https://www.researchgate.net/publication/261164882_Classification_of_Security_Operation_Centers, 2021.
 27. M. Ligh, A. Case, J. Levy, and A. Walters. *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory*. Wiley, 2014.
 28. Publicancy Ltd. *Managing Modern Security Operations Center Building Perfect Career as SOC Analyst*. Independently published, 2021.
 29. A. Mohanta and A. Saldanha. *Malware Analysis and Detection Engineering*. Apress, 2020.
 30. J. Muniz. *The Modern Security Operations Center*. Addison-Wesley Professional, 2021.
 31. J. Muniz, G. McIntyre, and Dr. N. AlFardan. *Security Operations Center: Building, Operating, and Maintaining your SOC 1st Edition*. Cisco Press, 2015.
 32. D. Murdoch. *Blue Team Handbook: SOC, SIEM, and Threat Hunting (V1.02)*. Independently published, 2019.
 33. Nathans and David. *Designing and Building Security Operations Center*. Syngress, 2014.
 34. The Art of Service Security Operations Center Publishing. *Security Operations Center A Complete Guide*. The Art of Service - Security Operations Center Publishing, 2020.
 35. Digit Orc Tabianto and I. Mujardianotto. *Malicious code analysis using Cuckoo sandbox*. Acon Publishing, 2014.
 36. A. Thomas. *Security Operations Center - Analyst Guide: SIEM Technology, Use Cases and Practices*. Arun E Thomas, 2017.

Ringraziamenti

A conclusione di questo elaborato, desidero menzionare tutte le persone, senza le quali questo lavoro di tesi non esisterebbe nemmeno.

Innanzitutto, ringrazio il mio professore Ursino che mi ha seguito, con la sua infinita disponibilità, in ogni step della realizzazione dell'elaborato.

Ringrazio di cuore i miei genitori. Grazie per avermi permesso di portare a termine gli studi universitari.

Ringrazio mio fratello Alessio, che mi ha sostenuto moralmente in tutto il percorso di studio universitario.

Ringrazio la mia fidanzata Valeria, per avermi trasmesso la sua immensa forza e il suo coraggio. Grazie per tutto il tempo che mi hai dedicato. Grazie perché ci sei sempre stata.

Ringrazio il mio Tutor e correlatore Balducci per i suoi preziosi consigli e per avermi guidato nel mondo della Cybersecurity e del SOC.

Grazie anche a tutti i colleghi di corso con cui ho affrontato esami e progetti.

Infine, vorrei dedicare questo piccolo traguardo anche a me stesso, che possa essere l'inizio di una lunga e brillante carriera professionale.