CAPE SANDBOX: SISTEM PENGESAN PERISIAN HASAD

Muhamad Akmal Bin Shamsul Hamidi, Masri Binti Ayob

Fakulti Teknologi & Sains Maklumat

43600 Universiti Kebangsaan Malaysia

Abstrak

Ancaman siber yang semakin kompleks menuntut pembangunan penyelesaian yang lebih cekap dan terfokus dalam mengesan perisian hasad. Sehubungan itu, projek ini membangunkan satu sistem pengesanan perisian hasad berasaskan *CAPE Sandbox* yang direka untuk melakukan analisis secara automatik terhadap fail mencurigakan menggunakan pendekatan statik dan dinamik. Melalui antara muka web berasaskan *Django*, penganalisis boleh memuat naik fail dan memperoleh laporan analisis yang terperinci. Sistem ini menggunakan gabungan dua mekanisme penyimpanan data fail rata untuk menyimpan hasil analisis lengkap, dan pangkalan data untuk menyimpan metadata seperti hash, status, masa analisis dan label bagi tujuan paparan pantas dalam antara muka web. Sistem ini menyokong dua peranan utama iaitu Pentadbir yang mengurus konfigurasi dan operasi sistem, serta Penganalisis Perisian Hasad yang melaksanakan pemuatan fail dan semakan laporan. Tambahan pula, sistem turut diintegrasikan dengan *API VirusTotal* bagi mendapatkan maklumat reputasi fail sebagai sokongan tambahan dalam pengesanan ancaman. Secara keseluruhan, sistem ini berfungsi sebagai alat pengesanan perisian hasad yang praktikal, modular, dan sesuai untuk tujuan penyelidikan, ujian makmal, serta pengembangan lanjut dalam bidang keselamatan siber.

Abstract

The increasing complexity of cyber threats demands the development of more efficient and focused solutions for detecting malware. In response, this project develops a malware detection system based on CAPE Sandbox, designed to automatically analyze suspicious files using both static and dynamic techniques. Through a Django-based web interface, analysts can upload files and obtain detailed and easy-to-understand analysis reports. The system employs a hybrid data storage mechanism: flat files are used to store complete analysis results, while a database is used to store metadata such as hash values, status, analysis time, and labels for quick retrieval and display via the web interface. The system supports two main user roles: the Administrator, responsible for system configuration and operation, and the Malware Analyst, who uploads files and reviews analysis reports. Additionally, the system integrates with the VirusTotal API to retrieve file reputation data as supplementary threat intelligence. Overall, this system functions as a practical, modular malware detection tool, suitable for research, lab testing, and further development in the field of cybersecurity.

1.0 PENGENALAN

Dalam Perisian hasad adalah merupakan singkatan untuk *"malicious software",* merujuk kepada pelbagai jenis program yang bertujuan untuk merosakkan sistem komputer. Menurut National Institute of Standards and Technology (NIST), perisian hasad didefinisikan sebagai sebuah program yang dimasukkan ke dalam sistem, biasanya secara tersembunyi, dengan niat untuk mengkompromi kerahsiaan, integriti, atau ketersediaan data, aplikasi, atau sistem operasi milik mangsa atau sebaliknya mengganggu mangsa. Ini menekankan objektif utama perisian hasad iaitu untuk mengganggu keselamatan sistem. Pelbagai sumber lain seperti *TechTarget* dan *Norton* turut menjelaskan bahawa perisian hasad berupaya mencuri, memadam, atau memantau data pengguna tanpa kebenaran, sering kali secara tersembunyi dan dengan niat jahat. Secara kolektif, perisian hasad dikenali dengan dua ciri utama: berniat jahat dan mampu bertindak tanpa pengetahuan pengguna.

Analisis perisian hasad boleh dilakukan secara statik atau dinamik. Analisis statik dilakukan tanpa menjalankan fail, manakala analisis dinamik dijalankan dalam persekitaran terkawal seperti mesin maya. Dengan peningkatan teknik penyamaran seperti polimorfisme, antivirus tradisional sering gagal mengesan perisian hasad. Kaedah pengesanan tradisional yang bergantung kepada tandatangan adalah tidak mencukupi untuk menghadapi ancaman perisian hasad yang semakin kompleks. Perisian hasad moden sering mengubah bentuk polimorfik untuk mengelak daripada dikesan. Tambahan pula, sistem pengesanan sedia ada tidak memberikan analisis mendalam terhadap interaksi perisian hasad dengan sistem dan rangkaian, menyebabkan pemahaman terhadap tingkah laku sebenar perisian tersebut menjadi terhad. Justeru, penggunaan CAPE Sandbox yang menyokong kedua-dua kaedah analisis ini menjadi pendekatan yang lebih komprehensif bagi meningkatkan kadar pengesanan dan pemahaman menyeluruh terhadap tingkah laku perisian hasad.

Untuk menangani isu di atas, projek ini mencadangkan pembangunan sistem pengesanan perisian hasad menggunakan *CAPE Sandbox*, iaitu kesinambungan daripada *Cuckoo Sandbox*. Sistem ini akan melaksanakan analisis statik dan dinamik terhadap fail perisian hasad dalam persekitaran terkawal, serta memantau interaksi rangkaian melalui integrasi *Suricata.* Hasil analisis akan merangkumi aktiviti seperti pemanggilan API, perubahan sistem, sambungan rangkaian, dan maklumat konfigurasi fail. Teknologi ini membolehkan pengesanan ancaman yang sukar dikesan oleh antivirus tradisional. Tambahan pula, penggunaan algoritma pembelajaran mesin berpotensi membantu dalam klasifikasi automatik berdasarkan tingkah laku yang dikesan. Objektif projek ini adalah untuk membangunkan sistem pengesanan perisian hasad menggunakan *CAPE Sandbox* bagi menganalisis tingkah laku secara statik dan dinamik serta mengukur keberkesanan sistem melalui kadar pengesanan dan ketepatan klasifikasi ancaman berdasarkan laporan tingkah laku.

Sistem dibangunkan menggunakan teknologi *Django* sebagai rangka kerja aplikasi web, *MongoDB* untuk pengurusan metadata tugasan, dan fail rata untuk penyimpanan hasil analisis seperti PCAP, laporan JSON, dan fail binari. Antara muka pengguna direka bentuk dengan menekankan aspek mesra pengguna dan ringkas, membolehkan penganalisis memuat naik fail dan menyemak laporan tanpa latihan teknikal khusus. Sistem ini beroperasi secara setempat pada satu peranti sahaja dan menyokong dua peranan pengguna iaitu Pentadbir dan Penganalisis Perisian Hasad.

Sistem turut mengintegrasikan *Suricata* untuk pengesanan ancaman berasaskan rangkaian serta *API VirusTotal* bagi mendapatkan reputasi fail yang dianalisis. Laporan disampaikan dalam bentuk interaktif dan automatik melalui antara muka web.

2.0 KAJIAN LITERATUR

Kesedaran terhadap ancaman siber telah mendorong pembangunan sistem pengesanan yang lebih canggih dan berautomasi. Perisian hasad kini menggunakan teknik penyamaran seperti polimorfisme, metamorfisme dan operasi tanpa fail untuk mengelakkan pengesanan konvensional. Menurut Sikorski dan Honig (2012), pendekatan statik sering gagal dalam mengenal pasti perisian hasad moden, terutama yang tidak mempunyai jejak fail. Oleh itu, terdapat keperluan mendesak terhadap penggunaan sistem seperti Sandbox, yang mampu menganalisis tingkah laku perisian hasad dalam persekitaran maya terkawal. CAPE Sandbox muncul sebagai kesinambungan daripada Cuckoo Sandbox, menawarkan analisis tingkah laku yang lebih mendalam serta integrasi modul rangkaian dan pembelajaran mesin.

Kajian oleh Essien dan Ele (2024) menunjukkan bahawa CAPE dan Cuckoo masing-masing berkesan dalam menganalisis interaksi proses, fail, dan API, dengan CAPE menunjukkan kelebihan dalam pemantauan rangkaian. Joe Sandbox pula menonjol dengan kemampuan *de-obfuscation* kod dan sokongan pelbagai platform, tetapi terhad dari segi kos pelesenan dan keperluan perkakasan yang tinggi (Prasad et al., 2024). Secara keseluruhan, CAPE menyediakan keseimbangan antara kemampuan teknikal dan kebolehcapaian, menjadikannya pilihan sesuai untuk institusi penyelidikan dan keselamatan siber berskala kecil hingga sederhana.

Dari segi metodologi, gabungan analisis statik dan dinamik terbukti memberikan hasil yang lebih komprehensif. Analisis statik memberikan gambaran awal terhadap struktur fail, manakala analisis dinamik membolehkan pemerhatian terhadap tingkah laku sebenar perisian hasad dalam sistem operasi. Penambahan pemantauan log dan integrasi dengan enjin pengesanan seperti *Suricata* dapat memperkayakan konteks analisis (Prasad et al., 2024). Sistem seperti CAPE juga menyokong penggunaan pembelajaran mesin bagi tujuan klasifikasi, membolehkan pengesanan automatik berdasarkan corak tingkah laku yang dipelajari.

Namun begitu, beberapa kekangan dikenal pasti dalam kajian lepas. Rafique et al. (2023) menekankan bahawa kekurangan standardisasi data set menghalang perbandingan adil antara sistem, manakala Prasad et al. (2024) mencadangkan bahawa kebanyakan sistem kurang fleksibiliti dari segi konfigurasi dan tidak responsif terhadap ancaman baru secara masa nyata. Tambahan lagi, kadar positif palsu yang tinggi dan ketiadaan integrasi dengan pangkalan data ancaman global menyukarkan pengesanan tepat terhadap perisian hasad baharu. Kekangan ini menjadi titik tolak kepada pembangunan sistem ini, yang cuba menangani isu-isu tersebut melalui integrasi API VirusTotal dan pengasingan modul analisis.

Sebagai rumusan, literatur terdahulu menegaskan keperluan untuk sistem pengesanan perisian hasad yang menyeluruh, adaptif dan mudah digunakan. CAPE Sandbox menampilkan potensi besar dengan menyatukan pelbagai pendekatan analisis dalam satu platform bersumber terbuka. Gabungan analisis statik, dinamik, log dan pengesanan rangkaian serta visualisasi laporan melalui antara muka web menjadikan sistem ini sebagai jalan penyelesaian yang

berdaya saing dalam persekitaran ancaman siber moden. Projek ini bukan sahaja menyambung dapatan kajian terdahulu, malah mencadangkan pelaksanaan teknikal sebenar sistem CAPE dalam bentuk boleh guna dan mesra pengguna.

3.0 METODOLOGI

Model pembangunan *Incremental Development* membahagikan sistem kepada beberapa modul kecil yang dibangunkan secara berturutan. Setiap modul diuji dan ditambah baik sebelum integrasi, membolehkan pengesanan awal ralat, pengubahsuaian berterusan, serta memastikan sistem akhir lebih stabil dan fleksibel.

3.1 Perancangan

Fasa perancangan melibatkan penetapan objektif dan skop projek, termasuk mengenal pasti keperluan asas sistem pengesanan perisian hasad serta senario pelaksanaan yang sesuai. Dalam projek ini, keputusan awal memfokuskan kepada persekitaran pembangunan, di mana sistem dibangunkan secara setempat menggunakan kaedah *dual boot* dengan Linux Ubuntu sebagai hos utama. Selain itu, kajian awal turut dilakukan terhadap ciri-ciri CAPE Sandbox dan cara ia diintegrasikan dalam persekitaran maya untuk analisis dinamik, di samping penyediaan jadual pelaksanaan dan senarai tugasan modul yang akan dibangunkan, khususnya antara muka pengguna.

3.2 Analisis

Fasa analisis dalam projek CAPE Sandbox ini bertujuan untuk mengenal pasti keperluan sistem berdasarkan proses analisis perisian hasad secara dinamik. Fokus utama adalah memahami aliran kerja penganalisis *malware*, termasuk keperluan untuk memuat naik sampel, menjalankan analisis dalam mesin maya, dan mendapatkan laporan tingkah laku secara automatik. Kajian turut dijalankan terhadap sistem sedia ada seperti *Cuckoo* dan *Joe Sandbox* bagi mengenal pasti ciri penting seperti pemantauan rangkaian, log sistem, serta kebolehan mengekstrak fail dijatuhkan dan *dump* memori. Selain itu, penilaian turut dibuat terhadap kesesuaian antara muka pengguna dalam menyampaikan data teknikal dengan berkesan. Hasil analisis ini dirumuskan dalam bentuk keperluan fungsian dan bukan fungsian sistem, yang menjadi asas kepada reka bentuk modul dalam fasa seterusnya.

3.3 Reka Bentuk

Fasa reka bentuk memberi tumpuan kepada pembangunan antaramuka web sistem CAPE yang mesra pengguna dan jelas dari segi navigasi. Antaramuka dibina supaya pengguna boleh memuat naik fail, melihat status tugasan, dan mengakses laporan dengan mudah. Reka bentuk turut mempertimbangkan susun atur struktur simpanan fail analisis dan integrasi paparan data penting seperti sambungan rangkaian, dump memori, dan skor ancaman. Pendekatan ini memastikan pengguna dapat memahami hasil analisis tanpa perlu kemahiran teknikal mendalam.

3.4 Implementasi

Fasa implementasi melibatkan pembangunan sistem pengesanan perisian hasad menggunakan *CAPE Sandbox* secara setempat pada peranti *Linux Ubuntu*. Dalam fasa ini, pemasangan dan konfigurasi *CAPE Sandbox* dilakukan termasuk penyediaan mesin maya *Windows 10* sebagai

persekitaran analisis. Modul antara muka web dibangunkan menggunakan *Django*, dengan struktur simpanan data menggunakan *MongoDB* bagi metadata tugasan serta fail rata untuk laporan dan artifak analisis. Perkhidmatan *Suricata* turut dikonfigurasi untuk memantau trafik rangkaian semasa analisis, manakala sambungan ke *VirusTotal API* akan dipertimbangkan dalam fasa seterusnya. Setiap komponen diuji secara berperingkat untuk memastikan keserasian dan ketepatan sistem.

3.5 Pengujian

Dalam Fasa ini bertujuan untuk memastikan sistem pengesanan perisian hasad berasaskan *CAPE Sandbox* berfungsi dengan baik serta memenuhi keperluan pengguna akhir. Pengujian dibahagikan kepada dua kategori utama iaitu pengujian berfungsi dan pengujian tidak berfungsi.

Bagi pengujian berfungsi, teknik pengujian kotak hitam (black-box testing) digunakan ke atas semua fungsi utama sistem tanpa menguji kod dalaman. Antara fungsi yang diuji termasuklah proses log masuk pengguna, fungsi muat naik fail perisian hasad, pelaksanaan analisis dalam mesin maya, serta paparan laporan automatik seperti dump memori, fail dijatuhkan, dan skor ancaman. Ujian ini membantu memastikan setiap fungsi utama sistem beroperasi dengan betul dari perspektif pengguna.

Dalam kategori pengujian tidak berfungsi pula yang melibatkan soal selidik kepada pengguna untuk menilai tahap kebolehgunaan sistem. Aspek seperti kemudahan antara muka, kefahaman terhadap paparan laporan, dan keselesaan penggunaan telah diambil kira. Di samping itu, pengujian prestasi dilakukan secara tidak langsung melalui penetapan had masa analisis sekitar 10 minit bagi menyesuaikan dengan kekangan sumber perkakasan. Hasil ujian menunjukkan sistem dapat menghasilkan laporan yang memadai dalam tempoh tersebut.

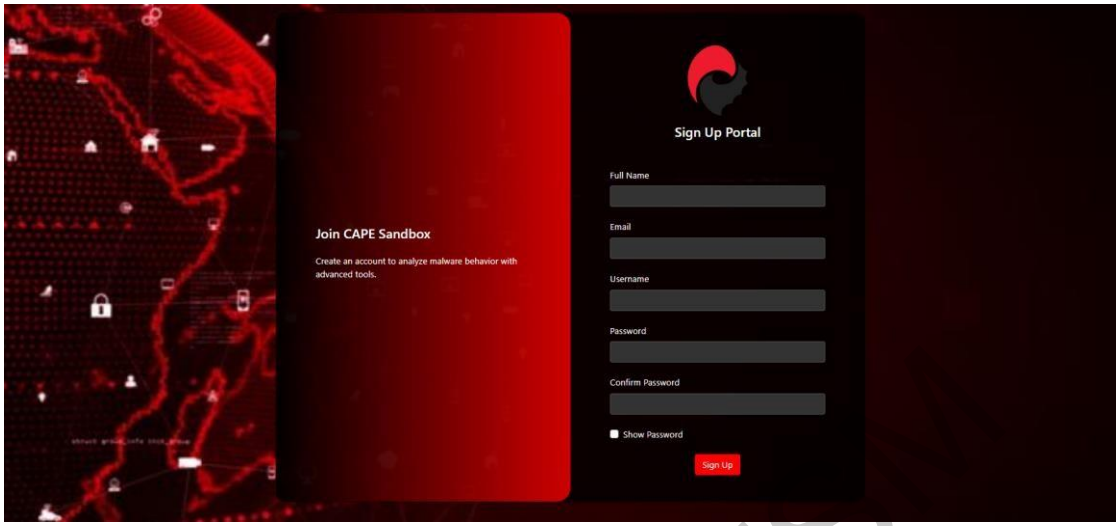
4.0 HASIL

4.1 Pembangunan Aplikasi

Proses pembangunan sistem pengesanan perisian hasad ini dijalankan secara bertahap menggunakan pendekatan *Incremental Development*, di mana setiap modul dibangunkan dan diuji secara berasingan sebelum diintegrasikan. Sistem dibina untuk beroperasi secara setempat pada peranti Linux Ubuntu menggunakan konfigurasi dual-boot. Antara perkhidmatan utama yang disediakan oleh CAPE termasuk *cape.service, cape-web, cape-rooter*, dan *cape-processor,* yang diurus dan dimulakan sebagai perkhidmatan sistem *(systemd service).*

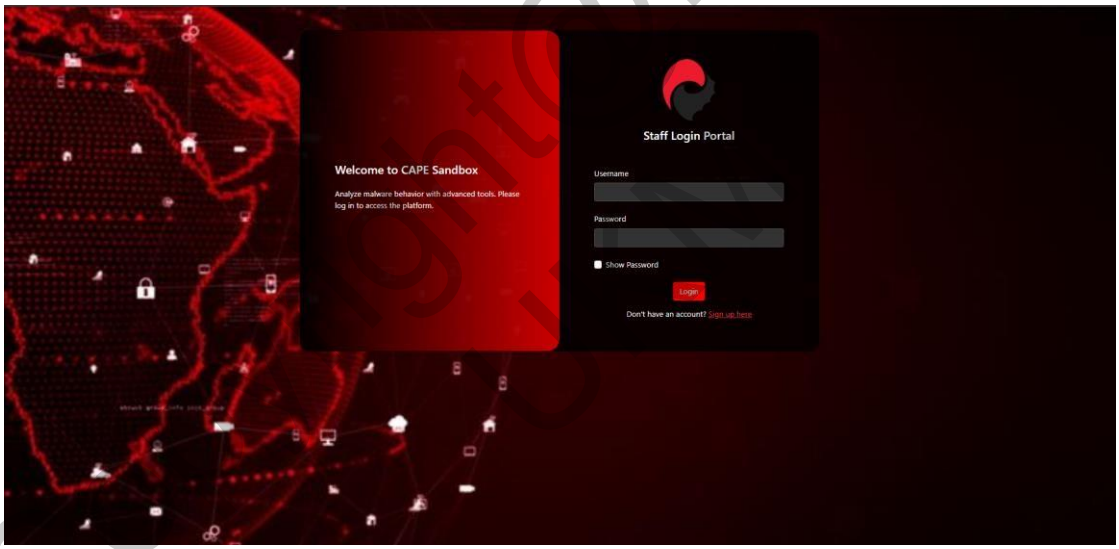
Antara muka web dibangunkan menggunakan *Django* untuk memudahkan pengguna memuat naik fail, memantau status analisis, dan mengakses laporan. Penyimpanan metadata seperti hash, nama fail, dan status analisis diurus oleh *MongoDB*, manakala hasil analisis seperti *report.json, pcap, dump memory,* dan fail dijatuhkan disimpan sebagai fail rata di dalam direktori *storage/.*

Bagi pengguna baharu, pengguna perlulah mendaftar akaun baharu dengan mengisi maklumat yang sepatutnya. Rajah 1 menunjukkan antara muka daftar akaun.



Rajah 1 Antara Muka Daftar Akaun

Seterusnya, pengguna perlu log masuk menggunakan emel dan kata laluan yang telah dibuat. Rajah 2 menunjukkan antara muka log masuk.

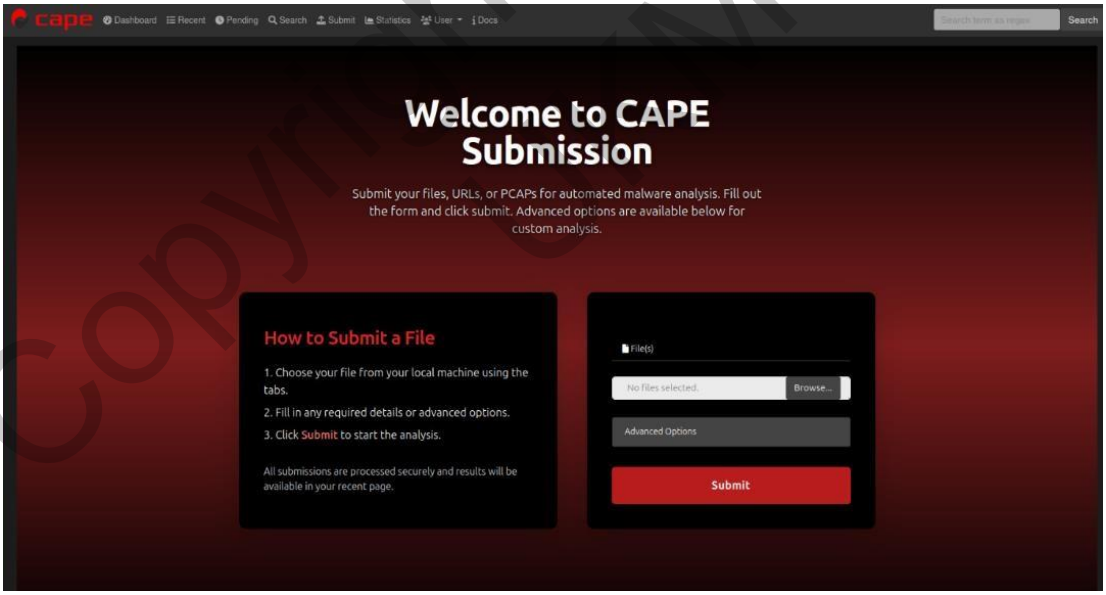


Rajah 2 Antara Muka Log Masuk

Pengguna yang berjaya log masuk akan melihat empat pilihan utama dalam halaman menu, iaitu halaman rumah, modul pembelajaran, kemajuan, dan akaun. Rajah 3 menunjukkan antara muka halaman-halaman yang disebutkan tadi.

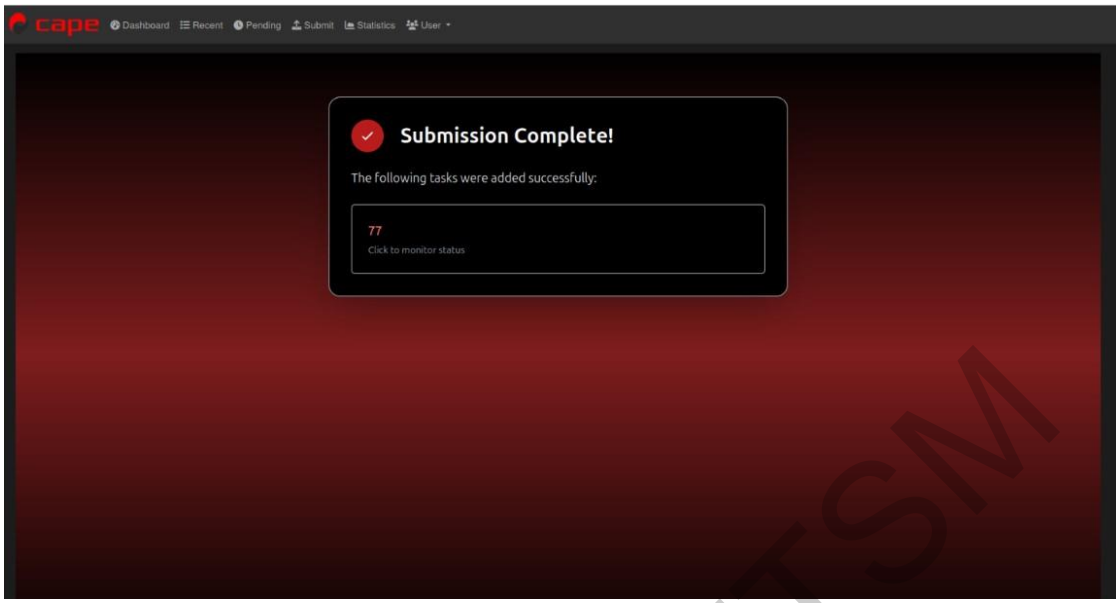
Rajah 3 Antara Muka Halaman utama

Setelah log masuk, pengguna akan dibawa ke halaman utama sistem yang memaparkan senarai tugasan sedia ada beserta status analisis seperti *pending, running,* atau *reported.* Pengguna boleh mengakses butang *submit new sample* untuk memilih dan menghantar sampel perisian hasad ke mesin maya.



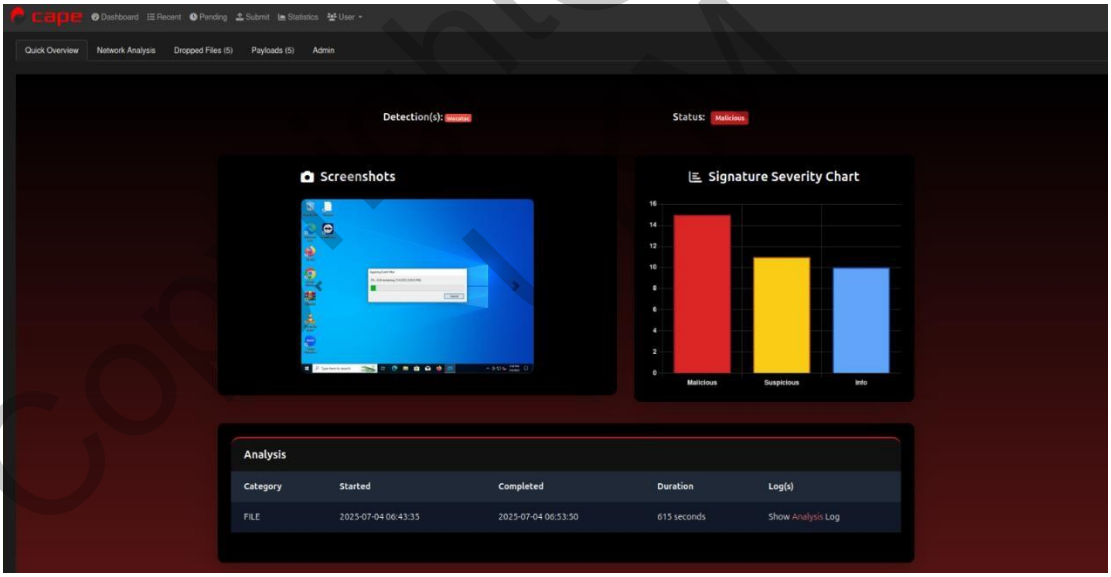
Rajah 4 Antara Muka muat naik fail

Setelah itu, antara muka ini ialah halaman penghantaran fail dalam CAPE Sandbox. Ia mempunyai dua bahagian: sebelah kiri memberi panduan ringkas cara memuat naik fail, manakala sebelah kanan ialah borang sebenar dengan butang *"Browse", "Advanced Options"* dan *"Submit"*. Reka bentuknya ringkas dan mudah digunakan.



Rajah 5 Antara Muka submission complete

Selepas fail dihantar, halaman ini muncul untuk memaklumkan bahawa tugasan telah berjaya didaftarkan ke sistem. ID tugasan akan dipaparkan dan boleh diklik untuk melihat status analisis secara langsung.



Rajah 6 Antara Muka paparan laporan

Setelah analisis selesai, pengguna boleh melihat laporan penuh yang memaparkan tangkapan skrin aktiviti dalam mesin maya, carta tahap keparahan berdasarkan tandatangan *(signature),* serta maklumat seperti masa mula, tamat, tempoh masa dan log analisis. Keputusan seperti "Malicious" dipaparkan untuk menunjukkan status ancaman fail yang dianalisis. Selain itu, pengguna juga boleh menavigasi ke tab-tab khusus seperti *Network Analysis, Dropped Files, Payloads*, dan *Process Memory* untuk mendapatkan maklumat terperinci mengenai tingkah laku perisian hasad sepanjang sesi analisis.

4.2 Penilaian Aplikasi

i. Pengujian Fungsian

Berikut adalah merupakan keputusan bagi Ujian Kotak Hitam. Kes ujian disusun dalam jadual berdasarkan fungsi utama sistem. Setiap kes mengandungi proses, fungsi ujian, input, jangkaan output, anggaran masa dan status. Jadual 1 menunjukkan keputusan bagi Ujian Kotak Hitam.

Jadual 1Keputusan bagi Ujian Kotak Hitam

Bil	Proses	Fungsi Ujian	Input	Jangkaan Ouput	Anggaran Masa(saat)	Status
1	Pendaftaran pengguna	Daftar akaun	Nama, emel, username, kata laluan	Akaun berjaya didaftar.	5–10	Lulus
2	Log masuk pengguna	Log masuk	Emel dan kata laluan	Masuk ke skrin utama pengguna	5–10	Lulus
3	Log keluar	Log keluar	Klik butang “Log Keluar”	Kembali ke skrin log masuk	1–3	Lulus
4	Muat naik fail	Penghantaran fail untuk dianalisis	Fail mencurigakan	Fail dihantar ke sistem dan tugasan dijana	5–10	Lulus
5	Proses Analisis	Jalankan analisis dinamik & statik	Fail yang dimuat naik	VM dilancarkan, aktiviti fail dianalisis, laporan dijana selepas tamat	300–600	Lulus
6	Papar laporan analisis	Semak hasil	Klik pada tugasan yang siap	Laporan dengan maklumat dump, network, suricata, dan VT dipaparkan	5–15	Lulus

7	Reputasi VirusTotal	Semakan reputasi fail secara automatik	Fail dengan hash tertentu	Status reputasi dari vendor VirusTotal ditunjukkan	5–10	Lulus

ii. Pengujian Bukan Fungsian

Ujian Penerimaan Pengguna (UAT) telah dijalankan ke atas sistem CAPE Sandbox bagi menilai aspek bukan fungsian seperti pengalaman pengguna, kemudahan penggunaan antara muka, persembahan visual, dan kecekapan dalam paparan laporan analisis. Seramai 5 orang pengguna sasaran, terdiri daripada pentadbir dan penganalisis, telah mengambil bahagian dalam penilaian ini. Soal selidik yang digunakan merangkumi tiga komponen utama iaitu kebolehgunaan sistem, kejelasan antara muka dan tema, serta kebolehcapaian kandungan analisis yang bersifat dinamik.

A. Ujian Kebolehgunaan dan Kefahaman Sistem

Hasil soal selidik menunjukkan sistem mudah difahami dan digunakan oleh pengguna. 60% peserta memberikan skor 5 untuk kefahaman dan 80% memberikan skor tertinggi untuk kemudahan penggunaan. Ini membuktikan antara muka sistem adalah mesra pengguna dan tidak memerlukan latihan teknikal khusus.

B. Ujian Keberkesanan Fungsi dan Ketepatan Operasi

Soalan berkaitan sama ada sistem berfungsi seperti dijelaskan dan beroperasi dengan baik telah memperoleh penilaian yang konsisten. Majoriti pengguna memberi skor 4 dan 5, menunjukkan sistem dapat melaksanakan analisis serta memaparkan laporan tanpa ralat yang kritikal.

C. Ujian Tahap Kepuasan Pengguna Keseluruhan

80% pengguna memberikan skor 4 dan 5 untuk kepuasan keseluruhan, menunjukkan bahawa pengalaman menggunakan sistem adalah positif dari aspek visual, capaian laporan dan aliran interaksi..

5.0 KESIMPULAN

Sistem ini dibangunkan sebagai satu platform analisis perisian hasad yang berfungsi secara setempat, dengan matlamat untuk membantu pentadbir dan penganalisis keselamatan siber dalam mengenal pasti ancaman melalui pendekatan analisis dinamik dan statik berasaskan CAPE Sandbox. Sepanjang proses pembangunan, pelbagai cabaran teknikal telah dihadapi, termasuk konfigurasi persekitaran mesin maya, integrasi antara modul analisis seperti *Suricata* dan *VirusTotal API*, serta pengurusan paparan laporan dalam antara muka web.

Namun begitu, setiap cabaran berjaya diatasi melalui kaedah pembangunan bertahap dan pelarasan konfigurasi sistem. Ujian telah dilaksanakan secara menyeluruh meliputi ujian fungsian (black-box testing), ujian bukan fungsian (kebolehgunaan, prestasi, ketahanan), serta Ujian Penerimaan Pengguna (UAT) yang melibatkan pengguna sasaran. Keputusan ujian menunjukkan sistem berfungsi dengan baik, mudah digunakan, dan memenuhi keperluan pengguna dari sudut fungsi serta pengalaman interaksi.

Secara keseluruhannya, sistem ini membuktikan potensi untuk digunakan dalam persekitaran penyelidikan atau latihan keselamatan siber, dan boleh dikembangkan lagi dengan integrasi analitik lanjutan, automasi pelaporan, atau pelaksanaan berskala lebih besar.

5.1 Masalah Dan Cabaran

Sepanjang pembangunan sistem pengesanan perisian hasad berasaskan CAPE Sandbox, beberapa masalah dan cabaran teknikal telah dikenal pasti. Antara cabaran utama ialah proses konfigurasi awal CAPE yang kompleks dan memerlukan pemahaman mendalam terhadap sistem virtualisasi seperti KVM/QEMU, serta penyesuaian modul analisis seperti *cape.service, processor,* dan *web interface* agar dapat berfungsi serentak tanpa konflik.

Dari sudut antara muka, cabaran melibatkan penyusunan hasil analisis seperti dump memori, fail dijatuhkan, dan log aktiviti ke dalam paparan web yang mudah difahami pengguna bukan teknikal. Isu juga berlaku dalam penjanaan laporan sekiranya fail analisis dipadam secara manual dari storan tempatan, mengganggu akses kepada laporan terdahulu.

Akhir sekali, sekatan keselamatan menjadikan pengumpulan sampel *malware* sebenar terhad, memerlukan penyelidik mendapatkan sampel daripada pangkalan data terbuka dan menyaringnya secara manual bagi tujuan ujian. Walaupun begitu, kesemua cabaran ini berjaya ditangani secara progresif melalui pendekatan pembangunan bertahap dan ujian sistematik.

5.2 Langkah Yang Diambil Untuk Mengatasi Masalah

Bagi menangani cabaran konfigurasi awal sistem, rujukan dokumentasi rasmi CAPE serta komuniti sumber terbuka seperti *GitHub* dan forum keselamatan siber telah digunakan secara meluas. Panduan langkah demi langkah telah disusun untuk memastikan pemasangan perkhidmatan seperti *cape.service, cape-processor.service*, dan *cape-web.service* dilakukan dalam susunan yang betul, serta untuk mengelakkan konflik semasa sistem diaktifkan.

Untuk isu antara muka dan paparan laporan yang kompleks, pendekatan minimalis telah diterapkan dalam mereka bentuk paparan web. Antara muka telah disusun semula agar laporan seperti tangkapan skrin, maklumat sambungan rangkaian, dan skor ancaman dipaparkan dalam bentuk berstruktur dan mudah difahami walaupun oleh pengguna tanpa latar belakang teknikal.

Masalah berkaitan kehilangan laporan akibat fail storan yang dipadam telah diatasi dengan menetapkan polisi penyimpanan fail agar setiap hasil analisis disimpan dalam struktur folder tetap yang tidak dikacau secara manual. Prosedur pemantauan juga ditambah untuk mengelakkan gangguan kepada fail kritikal.

Bagi menangani kekangan dalam mendapatkan sampel perisian hasad, beberapa repositori terbuka seperti *MalwareBazaar* dan *VirusShare* telah digunakan, di samping prosedur penapisan *SHA256* untuk memastikan kesahan fail. Sampel juga diuji terlebih dahulu dalam mod masa singkat bagi mengurangkan risiko gangguan terhadap sistem.

Secara keseluruhan, pendekatan penyelesaian yang berasaskan dokumentasi, ujian terkawal, serta reka bentuk mesra pengguna telah membolehkan cabaran teknikal diselesaikan secara berperingkat sepanjang pembangunan sistem. sesuai.

6.0 RUJUKAN

Essien, U. E., & Ele, S. I. (2024). Cuckoo Sandbox and Process Monitor (PROCMON) performance evaluation in large-scale malware detection and analysis. *British Journal of Computer Networking and Information Technology, 7*(4), 8–26. https://doi.org/10.52589/bjcnit-fcedoomy

Prasad, R., Kumar, A., & Mehta, P. (2024). Behavioral analysis and evasion techniques in malware detection systems: A review. *Journal of Information Security and Applications, 78*, 103623. https://doi.org/10.1016/j.jisa.2023.103623

Rafique, M., Ali, M., & Latif, K. (2023). Challenges and opportunities in malware sandboxing and classification: A systematic review. *Computers & Security, 125*, 102965. https://doi.org/10.1016/j.cose.2023.102965

National Institute of Standards and Technology. (n.d.). *Glossary of key information security terms*. Retrieved September 10, 2024, from https://csrc.nist.gov/Glossary/?term=5373