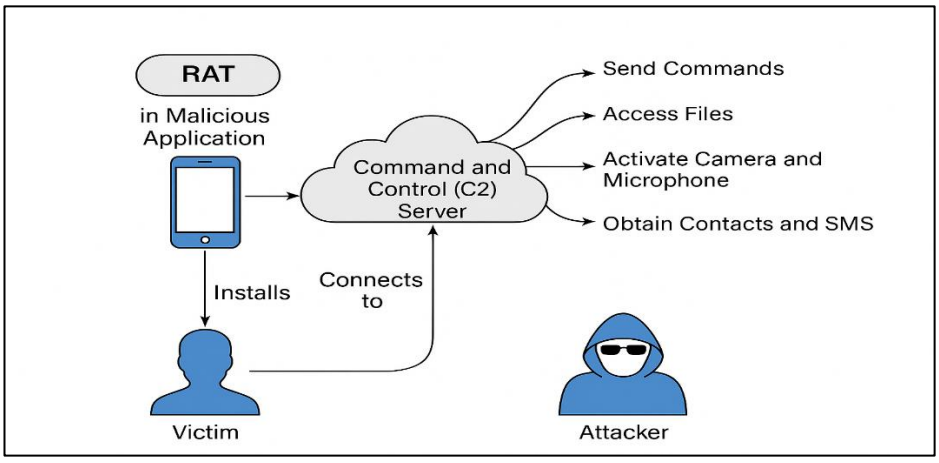
BAB II

LANDASAN TEORI

2.1 *Remote Access Trojan*

Remote Access Trojan (RAT) dikategorikan sebagai *malware* berbahaya yang dirancang untuk memberikan akses kendali jarak jauh kepada pelaku terhadap perangkat korban. Salah satu cara RAT didistribusikan adalah melalui *Trojan Downloader* yang menggunakan pendekatan *reverse engineering* untuk menembus mekanisme pertahanan sistem operasi dan aplikasi. Teknik ini memungkinkan penyerang untuk menanamkan *payload* secara tersembunyi ke dalam aplikasi, tanpa terdeteksi oleh sistem keamanan konvensional (Putra & Santoso, 2022). Gambar 2.1 merupakan arsitektur cara kerja *Remote Access Trojan* (RAT)



Gambar 2.1 Arsitektur RAT*(El-Metwaly & Abdelfattah, 2024)*

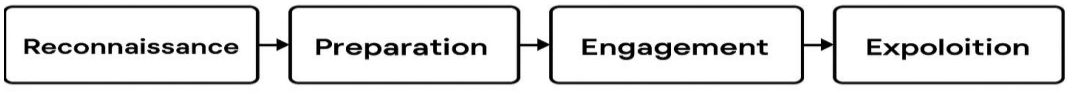
Penggunaan *reverse TCP payload* merupakan teknik umum dalam skenario serangan RAT. *Payload* ini berfungsi untuk membuka koneksi balik secara otomatis dari perangkat korban ke *server* penyerang, sehingga memfasilitasi jalannya komunikasi *command and control* (C2) secara diam-diam. Melalui kanal tersebut,

pelaku dapat menjalankan perintah dari jarak jauh, mencuri data, dan memanipulasi sistem target tanpa interaksi langsung (Moegiono, 2024).

Remote Access Trojan (RAT) dapat menyebabkan berbagai dampak serius terhadap perangkat dan pengguna. RAT memungkinkan penyerang mengakses data pribadi seperti kontak, SMS, dan file penting sehingga menimbulkan kebocoran informasi. Selain itu, perangkat dapat dikendalikan dari jarak jauh, termasuk mengaktifkan kamera dan mikrofon tanpa sepengetahuan pengguna. Data yang diperoleh juga berpotensi disalahgunakan untuk serangan lanjutan seperti phishing. Dalam beberapa kasus, RAT dapat menimbulkan kerugian finansial akibat akses ke aplikasi perbankan atau kode OTP. Secara umum, RAT mengancam privasi, keamanan data, dan stabilitas sistem perangkat.

2.2 *Social Engineering*

Social engineering atau rekayasa sosial merupakan pendekatan manipulatif yang memanfaatkan teknik psikologis guna mempengaruhi individu agar menyerahkan informasi atau akses sistem secara sukarela. Manusia merupakan komponen paling lemah dalam rantai keamanan siber, karena keputusan yang diambil pengguna sering kali didasari oleh kepercayaan atau kelengahan, bukan pemahaman teknis (G. S. Mahendra dkk., 2022). Mekanisme manipulasi sosial (*social engineering*) yang digunakan untuk menyebarkan RAT dapat dilihat pada Gambar 2.2.



Gambar 2.2 Mekanisme *social engineering (Salahdine & Kaabouch, 2019)*

Seiring berkembangnya teknologi pertahanan *digital*, teknik manipulasi sosial masih sangat relevan dan terbukti efektif dalam menembus sistem keamanan, terutama melalui serangan berbasis interaksi manusia (Gunawan dkk., 2025). Serangan berbasis *social engineering* kerap kali lebih berbahaya dibandingkan serangan teknis murni, karena mampu memanfaatkan kepercayaan korban secara langsung melalui teknik *phishing*, aplikasi palsu, atau pesan penipuan yang tampak meyakinkan (Melik, 2025.).

2.3 Eksploitasi *Android*

Eksploitasi terhadap sistem operasi *Android* merujuk pada rangkaian tindakan yang secara sengaja memanfaatkan celah keamanan pada sistem, aplikasi, maupun konfigurasi perangkat untuk memperoleh akses tanpa otorisasi. *Android*, sebagai *platform* berbasis *open source*, memang dirancang dengan fleksibilitas tinggi yang memberi kebebasan kepada pengembang dan pengguna dalam melakukan modifikasi. Namun demikian, karakteristik tersebut dapat menjadi celah yang berpotensi dimanfaatkan oleh aktor berbahaya untuk melakukan penyalahgunaan (P. Mahendra dkk., 2022). Metode eksploitasi dapat dilakukan melalui berbagai jalur, diantaranya dengan memanipulasi aplikasi dari pihak ketiga, melakukan *privilege escalation*, menerapkan teknik *reverse engineering*, atau menyisipkan *malicious payload* pada *file* APK yang tampak sah (Kusnadi & Ramadhan, 2023).

Eksploitasi pada perangkat *Android* umumnya terjadi karena adanya celah pada aplikasi sistem atau layanan bawaan yang belum terlindungi dengan baik. Penggunaan izin aplikasi yang terlalu luas (*over-permission*) merupakan titik lemah

yang paling sering dieksploitasi oleh pelaku (R. Putra dkk., 2021). Sementara itu, serangan RAT berbasis *Android* kerap kali mengandalkan mekanisme teknis seperti *code injection*, *buffer overflow*, dan koneksi *reverse TCP* sebagai jalur komunikasi balik dengan *server* penyerang (Suryanto & Nugroho, 2024). Kombinasi antara eksploitasi teknis dan manipulasi sosial sering kali digunakan secara bersamaan guna meningkatkan efektivitas serangan, menunjukkan bahwa dimensi teknis dan psikologis dalam eksploitasi saling melengkapi.

2.4 Metasploit

Metasploit Framework adalah platform uji penetrasi berbasis *open-source* yang dikembangkan untuk membantu proses identifikasi serta eksploitasi kerentanan sistem secara sistematis. *Framework* ini menyediakan berbagai modul utama, seperti *exploit, payload, auxiliary*, dan *post-exploitation*, yang memungkinkan setiap tahapan pengujian keamanan dijalankan secara terstruktur. Di samping itu, *Metasploit* juga berperan sebagai *command and control* (C2) *framework* yang mengelola komunikasi antara pihak penyerang dan perangkat target, sehingga aktivitas pengendalian jarak jauh dapat dilakukan secara terorganisasi serta terdokumentasi dengan baik. (Routray & Satapathy, 2020).

Relevansinya terhadap penelitian ini tampak pada fungsinya sebagai instrumen inti dalam merancang dan menganalisis integrasi *Remote Access Trojan* (RAT) dengan teknik *social engineering* pada *platform Android*. Melalui pemanfaatan modul-modul yang tersedia, *Metasploit* digunakan untuk menyusun skenario eksploitasi yang runtut, dimulai dari inisiasi koneksi hingga tahap kontrol penuh terhadap perangkat sasaran. Lebih lanjut, fleksibilitas yang dimiliki

framework ini memungkinkan penerapan mekanisme *auto exploitation,* yakni proses eksploitasi dan aktivitas *pasca*-koneksi yang dapat berjalan secara otomatis tanpa keterlibatan manual secara langsung. Dengan karakteristik tersebut, Metasploit memberikan dukungan signifikan terhadap pencapaian tujuan penelitian, khususnya dalam pengembangan model eksploitasi *Android* yang terintegrasi, terotomatisasi, dan merepresentasikan dinamika serangan dalam kondisi yang mendekati situasi nyata. (Sharma dkk., 2023).

2.5 MSFVenom

Msfvenom merupakan komponen utama dalam *Metasploit Framework* yang berfungsi sebagai pembangkit *payload* sekaligus encoder untuk menghasilkan kode berbahaya dalam berbagai format, termasuk *file* APK *Android. Tool* ini menyatukan fungsi *msfpayload* dan msfencode sehingga proses pembuatan *payload* menjadi lebih praktis dan efisien. Dalam konteks eksploitasi *Android, msfvenom* umum digunakan untuk membangun aplikasi APK yang mampu membentuk koneksi *reverse TCP*, sehingga perangkat target dapat dikendalikan dari jarak jauh melalui mekanisme *command* and *control.* (Kaur & Kaur, 2021).

Keterkaitan *msfvenom* dengan tujuan penelitian ini terletak pada perannya dalam tahap perancangan dan implementasi integrasi *Remote Access Trojan* dan teknik *social engineering*. Pada tujuan pertama, *msfvenom* digunakan untuk merancang *payload* yang disisipkan ke dalam aplikasi palsu sebagai media distribusi berbasis rekayasa sosial, sehingga memungkinkan terbentuknya skema eksploitasi yang terintegrasi. Sementara itu, pada tujuan kedua, *payload* yang dihasilkan dikonfigurasi agar mendukung mekanisme *auto exploitation*, yaitu

kemampuan untuk membangun koneksi dan menjalankan proses eksploitasi secara otomatis tanpa intervensi manual setelah aplikasi dijalankan. Dengan demikian, penggunaan msfvenom tidak hanya mendukung tahap teknis pembuatan RAT, tetapi juga menjadi pondasi dalam mewujudkan sistem eksploitasi *Android* yang terotomatisasi dan terstruktur sesuai dengan tujuan penelitian.

2.6 *State of The Art*

Tabel 2.1 *State-of-The-art*

No	Judul	Penulis	Metodologi	Hasil
1	*CANDY: A Social Engineering Attack to Leak Information from Infotainment System*	Costantino et al. (2018)	Eksperimen *Trojan-horse App di Android Infotainment,* distribusi via *social engineering*	Penelitian berhasil menunjukkan bagaimana aplikasi *Trojan* dapat disisipkan dalam aplikasi sah *(Gas Station Finder)* untuk mencuri data pengemudi dan membuka *backdoor* pada sistem *infotainment* kendaraan. **Kelebihan:** memberikan bukti nyata ancaman social engineering pada sistem otomotif. **Kekurangan:** Hanya terbatas pada skenario kendaraan dan belum membahas solusi mitigasi komprehensif.
2	*CyberCop: A Remote Surveillance Solution*	Patil et al. (2022)	Perancangan custom RAT untuk *Linux* khusus penegakan hukum	Hasil menunjukkan bahwa RAT dapat membantu investigasi digital dengan kemampuan stealth dan *bypass antivirus.* **Kelebihan:** Memberikan solusi legal untuk investigasi forensik digital.

No	Judul	Penulis	Metodologi	Hasil
				Kekurangan: Fokus hanya pada Linux, bukan Android; isu etika dan privasi masih besar.
3	*Android Spy Agent - Remote Access Trojan*	Barapatre & Parkhi (2020)	Implementasi RAT *Android* berbasis *reverse TCP, bypass antivirus*	Hasil penelitian menunjukkan RAT dapat berjalan *stealth* dengan *bypass firewall*, AV, dan *Google Play Protect.* **Kelebihan:** Menunjukkan kelemahan nyata sistem *Android* terhadap RAT. Namun **Kekurangan:** Masih *prototype* penelitian, rentan disalahgunakan bila tidak dikontrol.
4	*Analysis of Remote Access Trojan Attack using Android Debug Bridge*	Aprilliansyah, Riadi & Sunardi (2022)	Eksploitasi *Android* via ADB & *Ghost Framework*	Penelitian ini mengungkap kerentanan *Android* 9 terhadap serangan jarak jauh dengan *Ghost Framework* dan ADB, yang memungkinkan akses dan kontrol perangkat tanpa membuka kunci layar saat *debugging* aktif. **Kelebihan:** penelitian ini adalah kemampuannya membuktikan secara praktis kerentanan *Android* terhadap serangan RAT, memberikan gambaran lengkap proses eksploitasi serta rekomendasi *patching*. **Kekurangan:** terbatas pada perangkat tertentu, minim solusi mitigasi, bergantung pada konfigurasi khusus, serta

No	Judul	Penulis	Metodologi	Hasil
				kurang membahas aspek hukum dan etika.
5	*Analisis Malicious Software Trojan Downloader pada Android (Kamus Kesehatan v2.apk)*	Putra, Santoso & Ardiansyah (2022)	Teknik *reverse engineering* untuk menganalisis *malicious software trojan downloader* pada aplikasi *Android*	Penelitian ini menunjukkan bahwa infeksi *trojan downloader* pada aplikasi kamus kesehatan meningkatkan ukuran file dari 10,05 MB menjadi 10,17 MB serta menambah *permissions* dari 9 menjadi 27. **Kelebihan:** adalah penggunaan metode *reverse engineering* yang efektif dan relevan untuk meningkatkan kesadaran keamanan. **Kekurangan:** terletak pada tidak adanya detail pembuatan *payload* dengan *Metasploit* serta pengujian yang masih terbatas dan bersifat *pre-experimental.*
6	*Detecting Remote Access Trojan (RAT) Attacks based on Different LAN Analysis Methods*	Rashid et al. (2024)	Simulasi RAT di LAN dengan *Mega RAT,* analisis statis & dinamis	Penelitian menunjukkan RAT dapat dideteksi dengan analisis statis dan dinamis menggunakan tools gratis seperti *Wireshark.* **Kelebihan:** menawarkan metode deteksi yang murah dan mudah digunakan. **Kekurangan:** Uji coba hanya terbatas pada lingkungan VM, belum diuji di jaringan nyata skala besar.
7	*Analisa Metasploit "msfvenom"*	Mursalim, Darmawan & Aprilia (2024)	Pembuatan *trojan* via *msfvenom* &	Penelitian ini menunjukkan bahwa deteksi antivirus

No	Judul	Penulis	Metodologi	Hasil
	Backdoor Trojan dan FUD Trojan		*Python,* uji *antivirus*	terhadap *malware* FUD Trojan sangat rendah, dengan *payload chrome.exe* hanya terdeteksi 15% (11/71) dan *windows.dll* hingga 80% (56/70), menandakan *malware* dirancang agar sulit terdeteksi. **Kelebihan:** analisis mendalam tentang pembuatan *malware* dengan *Metasploit Framework* dan *Python* serta sorotan pada lemahnya deteksi antivirus dan pentingnya pembaruan rutin. **Kekurangan:** terletak pada pengujian yang hanya berbasis eksperimen dan fokus pada *antivirus* konvensional tanpa membahas solusi modern seperti kecerdasan buatan.
8	*Exploitation with Reverse_tcp Method on Android Device Using Metasploit*	Rizky Dwiananda Lukita Putra & Is Mardianto (2020)	Eksperimen menggunakan *framework Metasploit*	Penelitian ini membuktikan bahwa metode *reverse_tcp* dapat mengeksploitasi Android 7.1.2 non-*rooted* melalui bug *buffer overflow* dan *binder*, dengan *payload* APK yang dikonfigurasi lewat *bash script* dalam 27 ms. **Kelebihan:** efisiensi dan analisis teknis mendalam.

No	Judul	Penulis	Metodologi	Hasil
				Kekurangan: terbatas pada satu *versi Android* dan minim pembahasan mitigasi keamanan.
9	Analisis Keamanan dan Eksploitasi Kernel Android 13 Menggunakan Metasploit Reverse_TCP	alman Alhidamkara, Somantri, Ivana Lucia Kharisma (2024)	Eksploitasi *Android* 13 dengan *Metasploit* (*backdoor, Meterpreter,* data SMS & log panggilan)	Hasil penelitian membuktikan bahwa *Google Play Protect* mampu mendeteksi aplikasi berbahaya, namun aplikasi RAT masih dapat diinstal dan dijalankan sehingga memungkinkan pencurian data. **Kelebihan:** Menunjukkan kelemahan sistem keamanan *Android* terbaru. **Kekurangan:** Penelitian tersebut masih terbatas pada pengujian laboratorium dan solusi pencegahan yang diusulkan masih bersifat umum.
10	*Methods of stealing personal data on Android using a remote administration tool with social engineering techniques*	Hadikusuma & Lukas (2023)	Eksperimen pemasangan RAT melalui *phishing*	Penelitian berhasil membuktikan pencurian data (kontak, kamera, mikrofon) melalui RAT yang dipasang via *social engineering*. **Kelebihan:** Studi aplikatif dengan demonstrasi nyata. **Kekurangan:** Fokus eksploitasi, belum membahas strategi pencegahan secara detail.

Berdasarkan kajian pada Tabel 2.1, dapat diidentifikasi bahwa sebagian besar penelitian terdahulu telah berhasil membuktikan efektivitas *Remote Access Trojan* (RAT) dalam mengeksploitasi perangkat *Android*, baik melalui pendekatan

reverse TCP, pemanfaatan *framework* seperti *Metasploit*, maupun teknik *reverse engineering*. Penelitian seperti CANDY: A *Social Engineering Attack to Leak Information from Infotainment System* menekankan keberhasilan teknik *social engineering* dalam mendistribusikan aplikasi *Trojan*, sementara studi lain lebih berfokus pada aspek teknis eksploitasi, *bypass antivirus,* atau analisis kerentanan sistem. Namun demikian, sebagian besar penelitian tersebut masih memisahkan tahapan serangan secara parsial, yakni hanya pada tahap distribusi, eksploitasi, atau analisis *malware*, tanpa membangun model serangan yang terintegrasi.

Selain itu, proses eksfiltrasi data pada banyak studi sebelumnya umumnya masih dilakukan secara manual melalui perintah interaktif setelah sesi koneksi terbentuk. Belum banyak penelitian yang mengembangkan mekanisme otomatisasi pasca-eksploitasi yang mampu mengeksekusi perintah secara langsung dan mengirimkan laporan hasil pengambilan data secara tersembunyi kepada penyerang. Di sisi lain, pengujian juga masih sering terbatas pada lingkungan laboratorium atau *virtual machine*, sehingga belum sepenuhnya merepresentasikan skenario serangan yang realistis pada perangkat fisik dan jaringan berbeda.

Dengan demikian, *research gap* yang dapat diidentifikasi terletak pada belum adanya penelitian yang mengintegrasikan teknik *social engineering* dengan implementasi *Remote Access Trojan* berbasis *Android*. Oleh karena itu, penelitian berjudul *Integrasi Remote Access Trojan dan Social Engineering* untuk Eksploitasi *Android* hadir untuk mengisi celah tersebut dengan mengembangkan model serangan terintegrasi yang lebih komprehensif dan realistis, sehingga dapat

memberikan kontribusi baru dalam pemodelan ancaman siber berbasis *Android* serta memperkaya kajian eksperimental di bidang keamanan informasi.

Penelitian ini tidak berfokus pada penemuan celah baru pada protokol keamanan sistem operasi *Android*, melainkan pada pengembangan model eksploitasi yang terintegrasi dan terotomatisasi. Keterbaruan penelitian ini terletak pada integrasi antara *Remote Access Trojan* (RAT) dan teknik *social engineering* dalam satu skema serangan yang utuh, mulai dari tahap distribusi, eksploitasi, hingga eksfiltrasi data. Selain itu, penelitian ini memperkenalkan mekanisme *auto exploitation* yang memungkinkan eksekusi perintah pasca-eksploitasi secara otomatis tanpa intervensi manual, serta *automatic reporting* yang mampu mengirimkan hasil ekstraksi data secara *real-time*. Dengan pendekatan tersebut, penelitian ini menghasilkan simulasi serangan yang lebih sistematis, efisien, dan mendekati kondisi nyata dibandingkan dengan penelitian terdahulu yang umumnya masih bersifat parsial dan bergantung pada interaksi manual.

Tabel 2.2 Matriks Penelitian

No	**Judul**	**Peneliti**	**RAT**	***Social Engineering***	***Payload***	***Android***	**Automatic exploitation**	**Reporting**
1	*CANDY: A Social Engineering Attack to Leak Information from Infotainment System*	Costantino et al. (2018)	✓	✓	✓	✓	-	-
2	*CyberCop: A Remote Surveillance Solution*	Patil et al. (2022)	✓	✓	✓	✓	-	-

No	Judul	Peneliti	RAT	*Social Engineering*	*Payload*	*Android*	Automatic exploitation	Reporting
3	*Android Spy Agent – Remote Access Trojan*	Barapatre & Parkhi (2020)	✓	-	✓	✓	-	-
4	*Analysis of Remote Access Trojan Attack using Android Debug Bridge*	Aprilliansyah, Riadi & Sunardi (2022)	✓	✓	✓	✓	-	-
5	*Analisis Malicious Software Trojan Downloader pada Android (Kamus Kesehatan v2.apk)*	Putra, Santoso & Ardiansyah (2022)	✓	✓	-	✓	-	-
6	*Detecting Remote Access Trojan (RAT) Attacks based on Different LAN Analysis Methods*	Rashid et al. (2024)	✓	-	-	✓	-	-
7	*Analisa Metasploit "msfvenom" Backdoor Trojan dan FUD Trojan*	Mursalim, Darmawan & Aprilia (2024)	✓	✓	-	✓	-	-
8	*Exploitation with Reverse_tcp Method on Android Device Using Metasploit*	Rizky Dwiananda Lukita Putra & Is Mardianto (2019)	✓	-	✓	✓	-	-
9	Analisis Keamanan dan Eksploitasi Kernel Android 13 Menggunakan Metasploit Reverse TCP	alman Alhidamkara, Somantri, Ivana Lucia Kharisma (2024)	✓	-	✓	✓	-	-
10	*Methods of stealing personal data on Android using a remote administration tool with social*	Hadikusuma & Lukas (2023)	✓	✓	✓	✓	-	-

No	Judul	Peneliti	RAT	*Social Engineering*	*Payload*	*Android*	Automatic exploitation	Reporting
	engineering techniques							
11	***Penelitian sekarang***		✓	✓	✓	✓	✓	✓

Berdasarkan Tabel 2.2, dapat dilihat bahwa sebagian besar penelitian terdahulu memiliki kesamaan pendekatan dalam mengimplementasikan *Remote Access Trojan* (RAT) dan memanfaatkan teknik *social engineering* sebagai metode penyebaran *malware*. Namun demikian, sebagian besar penelitian tersebut masih bergantung pada proses eksploitasi yang dilakukan secara manual. Belum adanya studi yang menyoroti atau mengimplementasikan proses eksfiltrasi data secara otomatis, yang merupakan salah satu celah kritis dalam simulasi serangan dunia nyata.

Sebagai keterbaruan dari penelitian terdahulu, penelitian ini memperkenalkan mekanisme otomatisasi dalam proses *pasca-eksploitasi*, khususnya pada tahap pengambilan serta pengiriman *reporting* ke *email* penyerang.

Penelitian ini memberikan kontribusi dalam pengembangan model eksploitasi perangkat Android yang terintegrasi dengan menggabungkan *Remote Access Trojan* (RAT), teknik *social engineering*, serta mekanisme otomatisasi dalam satu alur sistematis. Selain itu, penelitian ini mengimplementasikan mekanisme *auto exploitation* yang memungkinkan proses *pasca-eksploitasi* berjalan secara otomatis tanpa intervensi manual, sehingga meningkatkan efisiensi dan konsistensi dalam pengambilan data. Penelitian ini juga memperkenalkan *automatic reporting* yang mampu mengirimkan hasil ekstraksi data secara *real-*

time, sehingga mempercepat proses dokumentasi dan analisis hasil eksploitasi. Dari sisi empiris, penelitian ini melakukan pengujian langsung pada perangkat *Android* nyata dengan menggunakan parameter kuantitatif seperti tingkat keberhasilan instalasi, pembentukan sesi, ekstraksi data, dan pelaporan, sehingga menghasilkan pendekatan yang lebih realistis dan komprehensif dibandingkan penelitian sebelumnya. Dengan demikian, penelitian ini tidak hanya berkontribusi secara konseptual, tetapi juga secara teknis dan empiris dalam bidang keamanan siber.