Jurnal SPIRIT Vol. 15 No. 1 Mei 2023, hal 42-48 P- ISSN : 2085 – 3092
E- ISSN : 2721 – 057X

DESAIN ARSITEKTUR APLIKASI QR CODE SEBAGAI ANTI PHISHING SERANGAN QR CODE

Slamet[1)]
Prodi S1 Sistem Informasi, Universitas Dinamika, Surabaya, Indonesia[1)]
email: slamet@dinamika.ac.id[1)]

***ABSTRACT**: QR Codes are very vulnerable to falsification because it is difficult to distinguish the original QR Code from a fake QR Code. Because of this vulnerability, the scanning process on fake QR Codes can direct users to dangerous sites with important information or data from the user. To assess QR Code security vulnerabilities and actions using a secure Application-based QR Code Architecture as Anti Phishing against QR Code attacks using hash functions and digital signatures. In experiments simulated attack types to malicious QR codes that redirect users to phishing sites. The real URL is disguised into the QR Code, where the user does not suspect, the URL is redirected to the fake site. As a result, intruders can easily use QR codes as vectors for phishing attacks targeted at smartphone users, even if they are using a browser that has security features.*

***Keywords:** QR code, smartphone security, phishing attack, digital signature*

1. Pendahuluan

QR Code telah menjadi salah satu kode dua dimensi yang lebih popular dibandingkan *barcode* karena kapasitas data yang melekat dan ketahanan kerusakan yang lebih tinggi [1]. Saat ini, keamanan dan *privasi smartphone* menjadi perhatian utama [2], *QR Code* bisa dibajak dan dapat menjadi vektor serangan yang berbahaya bagi pengguna *smartphone*. Dengan jutaan *QR Code* yang disebarkan banyak perusahaan di tempat umum, tidak sulit bagi *intruder* untuk mengganti atau memodifikasinya demi kepentingan pribadi dan niat jahatnya.

Seiring dengan pesatnya perkembangan teknologi, *QR code* telah diadopsi dan digunakan *smartphone*[3]. Hal ini tidak hanya meningkatkan popularitas dan penggunaan *QR code* pada sebagian besar aplikasi[4], namun *QR code* juga mengandung potensi vektor serangan yang dapat dimanfaatkan oleh *intruder*[5]. Dengan demikian, ancaman serius mengancam kepada pengguna *smartphone* yang tidak menaruh curiga kepada banyaknya *QR code* tersebut.

Pada awal 2023, penggunaan *smartphone* di Indonesia mendekati angka hampir 200 juta, melebihi jumlah komputer pribadi [6], karenanya ancaman ini menjadi semakin signifikan. Dibandingkan dengan *morris worm* yang telah menghancurkan 60 ribu komputer [7], data biner maksimum yang dihasilkan *QR code* dengan daya tampung kira-kira 2,9 KB mampu menjadi vektor ancaman dengan muatan berbahaya. *QR code* dapat digunakan dalam beberapa jenis serangan yang berbeda seperti *social engineering* [8] dan serangan proses otomatis. Serangan proses otomatis dijalankan dengan mengeksploitasi kerentanan *SQL injection*, *command injection*, serta serangan *Cross-Site Scripting* (XSS) [9].

Dalam makalah ini, dijelaskan kerentanan *smartphone* berkaitan dengan *QR code*, seluk-beluk *QR code* dan bagaimana hal itu dimanfaatkan sebagai vektor serangan oleh *intruder*. Bentuk serangan disimulasikan menggunakan *QR code* berbahaya, dimana *QR code* yang ada dimanfaatkan sebagai vektor serangan *phishing*, termasuk titik-titik rentan yang teridentifikasi. Selanjutnya, kami menyajikan solusi berupa arsitektur aplikasi *QR code* yang merinci komponen keamanan dan proses yang digunakan untuk mengautentikasi *QR code* dari pengguna.

2. Landasan Teori

Pada bagian ini dijelaskan teori-teori tentang *QR code*, celah dan pengamanan yang akan digunakan pada arsitektur aplikasi yang diusulkan.

2.1. QR Code

QR code adalah kode batang dua dimensi ditemukan oleh perusahaan Jepang Denso Wave [10]. Informasi dikodekan dalam dua dimensi, yaitu arah vertikal dan horizontal, sehingga menyimpan data hingga ratusan kali lebih banyak dari pada *barcode* tradisional, seperti gambar 1. Data diakses dengan cara mengambil foto dari *QR code* menggunakan kamera dan kemudian memproses gambar tersebut.

Gambar 1. *Barcode*

QR Code pada gambar 2, dengan cepat menjadi populer dan teknologinya diadopsi secara luas di seluruh dunia. Terutama di Jepang, dimana kemampuannya untuk menyandikan simbol Kanji secara *default* membuatnya sangat disukai masyarakat jepang. Penggunaan *QR code* sangat populer seperti dapat menyimpan *URL*, alamat dan berbagai bentuk data poster, data rambu lalu lintas, kartu nama, kendaraan angkutan umum, dan lain sebagainya. Mekanisme yang ada di *QR code* mampu menciptakan peluang dan dapat dimanfaatkan sebagian besar aplikasi potensial dan komersil [11, 12, 13, 14, 15].

Gambar 2. *QR code*

2.1. Celah sistem pada *Smartphone*

Phishing adalah masalah besar dan selalu akan terjadi [16] sebagai salah satu bentuk serangan pada *cybercrime*. Di masa yang akan datang, serangan *phishing* akan dilakukan dengan lebih mudah melalui *smartphone* dibandingkan melalui *browser desktop*. Saat ini pengguna mulai dapat melihat lebih banyak serangan *phishing* berupa *malware* pada *smartphone* [17]. *Intruder* membuat berbagai jenis *malware* untuk mengeksploitasi perangkat tersebut demi keuntungan finansial, untuk mengganggu sumber daya *smartphone*, mencuri informasi dan data, mengoptimalkan *search engine*, pesan spam, mengakses jaringan pribadi, atau bahkan merusak perangkat untuk hiburan. Umumnya, *malware* semacam itu dilampirkan ke aplikasi resmi yang popular atau pada aplikasi baru yang memiliki beberapa fungsi untuk mengelabui pengguna.

Smartphone menjadi perangkat yang tidak dapat dipisahkan dalam kehidupan sehari-hari, seperti melakukan panggilan telepon, mengirim pesan teks dan email, transaksi *online*, mengakses media sosial, mengakses data perusahaan, penyimpanan portabel, menyimpan informasi (catatan) dan banyak lagi. Hal ini membuat perangkat *smartphone* menjadi aset berharga untuk dijadikan sasaran serangan berbahaya.

Menurut [18], Pada 10 ribu responden terdapat 67,3% pengguna memiliki tingkat pengamanan yang rendah, sedang 32,7% yang tingkat pengamanannya tinggi terhadap *device*-nya. Dari 10 ribu responden, 91,3% mengimplementasikan upaya pengamanan dengan memasang *password* huruf atau angka atau pola untuk membuka layar *smartphone*, komputer, atau laptopnya. Sedangkan, yang menggunakan fitur kunci *fingerprint authentication* hanya 36,7%, dan fitur *face authentication* sebesar 13,5%. Mereka yang memasang fitur *antivirus* hanya 6,2%, sedangkan 4,3% mengimplementasikan *back-up* data dan 3,4% menggunakan fitur *find my device*. Sementara itu, 15,5% responden tidak melakukan aktifitas apapun dari hal di atas [19]. Selain itu, kurangnya keamanan pada aplikasi smartphone membuatnya rentan terhadap serangan tradisional seperti SQL *injection*, *script* CSS (XSS), dan *Man In The Middle Attack* (MITM). Beberapa aplikasi *antivirus smartphone* tidak menangani *malware* secara memadai karena keterbatasan yang diberlakukan oleh sistem keamanan Android [20]. Karenanya, tidak semua aplikasi *antivirus*, efektif mencegah *malware* dan *spyware* dalam menginfeksi *smartphone* Android [21]. Selain itu, sebagian besar perangkat lunak *antivirus* pada perangkat *smartphone* cerdas berbasis *signature* juga tidak mungkin melindungi perangkat dari serangan *malware* yang canggih (kompleks).

2.2. *QR Code* sebagai vektor serangan

QR code yang dimanipulasi dapat digunakan untuk banyak serangan. Bergantung pada apakah pembaca adalah manusia atau program otomatis, skenario serangan yang berbeda dimungkinkan dan diuraikan dalam bagian ini.

2.2.1 Serangan Proses Otomatis

Karena *QR Code* adalah cara standar untuk menyandikan informasi, sebagian besar *developer software* berusaha memperlakukan informasi yang disandikan sebagai masukan yang aman. Berbagai bagian *QR Code* dapat dimanipulasi untuk mengubah informasi yang disandikan. Bergantung pada program yang memproses informasi yang disandikan, apakah ini dalam bidang perdagangan, transportasi umum, atau yang sepenuhnya otomatis, serangan terhadap *reader* dan *backend* aplikasi secara teori dimungkinkan. Tanpa kehati-hatian dan *awareness* keamanan yang baik, hal ini dapat digunakan oleh *intruder* sebagai daftar serangan berikutnya. Serangan serupa menggunakan *chip* RFID dan SQL *injection* telah terbukti sangat efektif [22]. Berikut adalah bentuk serangan proses otomatis:

-. *SQL Injection*: Banyak sistem otomatis menyimpan dan memproses informasi yang disandikan dalam basis data relasional. Dengan menambahkan titik koma diikuti oleh *query SQL* seperti; *drop table <tablename>* ke informasi yang disandikan, manipulasi ke *database backend* dimungkinkan. Hal ini akan menghapus tabel yang ditentukan dalam perintah, mengakibatkan penolakan serangan layanan. Serangan yang lebih spesifik dapat mencakup penambahan pengguna, menjalankan perintah sistem (misalnya, dengan menggunakan prosedur tersimpan *xp_cmdshell* di Microsoft SQL Server), atau mengubah data seperti harga atau *password* di dalam *database* [23].

-. ***Command Injection***: Jika informasi terenkripsi digunakan sebagai parameter *command line* tanpa disanitasi, hal ini dapat dieksploitasi dengan mudah untuk menjalankan perintah sewenang-wenang atas nama *intruder*, yang dapat menimbulkan konsekuensi bencana bagi keamanan sistem operasi misalnya, memasang rootkit, DoS, atau menghubungkan *shell* ke komputer *remote* di bawah kendali *intruder* [24].

-. ***Fraud***: Perubahan pada sistem otomatis dapat digunakan untuk melakukan penipuan, dengan mengelabui sistem misalnya, dengan mempercayai bahwa ia memproses produk A yang murah sementara memproses produk B yang lebih mahal [25].

2.2.2 Serangan Interaksi Manusia

Manusia tidak dapat membaca kode tanpa aplikasi *reader*, karena informasi yang tersimpan di dalam kode benar-benar dikaburkan. Tetapi dengan membaca *QR Code* yang dimanipulasi, kerentanan pada aplikasi *reader* atau *browser* dapat dipicu karena manusia.

-. *Phishing*: Jika *QR Code* digunakan untuk *link* dalam skenario *augmented reality*, *intruder* dapat membuat *website* palsu dan mengalihkan pengguna dengan mengubah *QR Code*. Hal ini bisa berbahaya jika tidak ada bentuk-bentuk kredensial untuk mengakses *website* tersebut karena pengguna tidak memiliki kemungkinan untuk memverifikasi bahwa *link* yang ada tidak dimodifikasi [26].

-. *Fraud*: *QR Code* sering digunakan dalam *advertising* untuk mengarahkan target ke penawaran khusus atau informasi tambahan tentang produk tertentu. Jika *QR Code* dapat dimanipulasi untuk mengarahkan pengguna ke *website* palsu, *intruder* dapat menjual produk yang diminta tanpa pernah memenuhi kontrak yang disepakati. Korban secara implisit mempercayai perusahaan *advertising* dengan mengikuti *link* yang ada [25].

-. *Menyerang Aplikasi Reader*: Implementasi yang berbeda dari aplikasi *reader* pada komputer atau *smartphone* dapat diserang melalui *command injection* atau *buffer flow* jika informasi yang disandikan tidak dibersihkan. *Intruder* dapat menguasai seluruh isi *smartphone*, termasuk informasi kontak atau konten dan komunikasi calon korban seperti Email atau SMS [27].

-. *Serangan Social Engineering*: Berdasarkan model serangan ini, serangan spesifik seperti *spear phishing* atau varian lain dari *social engineering* diaktifkan, bergantung pada tujuan *intruder*. Meninggalkan *QR Code* di tempat umum yang menawarkan diskon di restoran terdekat adalah contoh vektor serangan baru yang kemungkinan akan berhasil [28].

3. Metode Penelitian

Pada bagian ini diimplementasikan arsitektur keamanan dari sisi *server* dan klien (aplikasi) dalam mempertahankan sistem keamanan *QR Code*. Sisi *server* mengendalikan dan mengautentikasi pengguna yang berusaha mengakses *QR code* perusahaan PT. BM1 Indonesia Bus (dapat diverifikasi oleh pengguna yang memindai kode). Sedangkan sisi klien adalah aplikasi yang tersedia (seperti *QR reader*) untuk penggunaan umum. Pengamanan ini mengimplementasikan fungsi *hash* dan *digital signature* untuk menjamin integritas dan keaslian *QR code*.

Fungsi hash yang aman (SHA-2, SHA-3) untuk menghasilkan pesan inti 256 bit, yang mampu bertahan terhadap serangan *brute force*. Properti ini

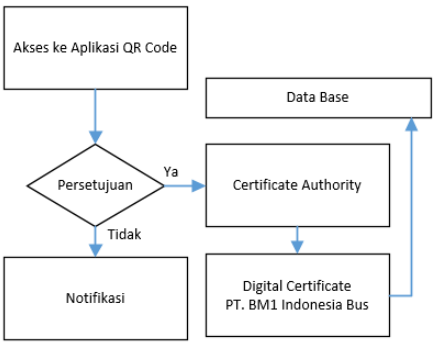
tidak hanya signifikan untuk penggunaan *digital signature* jika teks biasa cukup besar, tetapi juga mengatasi batasan penyimpanan dalam *QR code*.

Digital signature membuktikan bahwa entitas tertentu menghasilkan pesan yang diamati, seperti teks biasa ke dalam *QR code*. Kriptografi dicapai menggunakan *public key* yang memanfaatkan kesulitan untuk memfaktorkan bilangan prima atau masalah algoritma diskrit. Karena *smartphone* memiliki sumber daya yang terbatas untuk melakukan perhitungan berat, diusulkan penggunaan *digital signature* untuk menandatangani *fungsi hash*. Entitas yang menghasilkan teks biasa dan *hash* akan menandatangani *hash* secara digital dengan menggunakan *private key* dan *random key* (kunci sesaat yang berbeda setiap kali ingin menandatangani *QR code* baru).

Solusi yang diimplementasikan melakukan hal-hal di atas untuk menghasilkan *QR code* secara otomatis, sehingga pengguna yang memindai *QR code* kesulitan membajak karena memerlukan *reader* yang mampu mendekode, mendekripsi, dan memverifikasi *signature*.

3.1 Arsitektur sisi *Server*

Pada sisi *server* diimplementasikan *platform* yang dapat menampung otoritas sertifikat (untuk menghasilkan *digital certificate*) dan sistem registrasi. Karena pengguna dapat mengajukan permohonan agar disetujui di platform ini, pengguna dapat memperoleh akses ke *platform* dengan mengajukan dan mengirimkan dokumen yang diminta dengan identitas jelas. Entitas yang disetujui akan mendapatkan akses resmi dan *QR code* yang ditandatangani secara digital.



Gambar 3. Arsitektur *QR Code* sisi Server

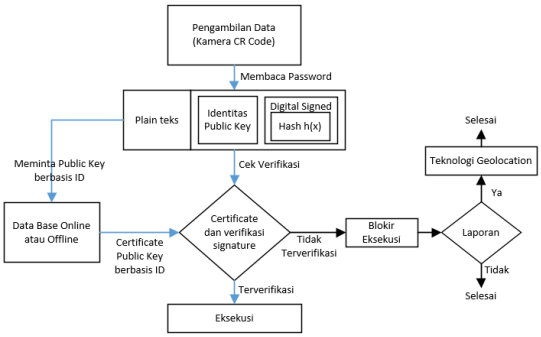
Dalam hal ini (pengguna) perusahaan PT. BM1 Indonesia Bus harus menyertakan teks yang diperlukan untuk dikodekan dan *platform* ini akan menghitung fungsi *hash* dan menandatanganinya secara digital (enkripsi dengan *public key* dan *random key*).

3.1 Arsitektur sisi Klien (Aplikasi)

Setelah memiliki *QR code* yang dibuat oleh *platform* ini, kode ini berisi teks biasa, nomor unik yang menentukan *public key*, dan *hash* yang ditandatangani secara digital. Gambar 4 menampilkan arsitektur *QR code* sisi Aplikasi (klien). Aplikasi dapat mengakses *public key* dari entitas yang disimpan secara *online* ke dalam basis data yang ada, atau dapat diunduh pada aplikasi untuk akses *online* jika koneksi jaringan tidak tersedia.

Saat kode dipindai dan terverifikasi, aplikasi akan memeriksa dengan menggunakan *public key* dan *signature*. Jika sudah sesuai, maka aplikasi akan melakukan tindakan yang dimaksud; jika tidak, *QR code* diklasifikasikan sebagai tidak terverifikasi. Apabila *QR code* tidak terverifikasi, aplikasi akan mengingatkan pengguna dan *QR code* akan diblokir. Selain itu, pengguna dapat mengirimkan laporan *QR code* yang tidak asli.

Opsi ini hanya tersedia untuk pengguna terdaftar yang telah membuktikan identitasnya dengan OTP (*one time password*) yang dikirim ke nomor ponsel saat pertama kali mendaftar. Misalnya, jika *QR code* tersedia di Internet, pengguna dapat mengirimkan URL yang menargetkan kode yang belum diverifikasi. Pengguna juga memiliki opsi untuk menentukan lokasi geografis dari kode tidak asli serta organisasi yang diklaim berasal dari *QR code*. Opsi ini hanya tersedia setelah kode ditandai sebagai tidak terverifikasi. Selanjutnya, pengguna dapat menggunakan aplikasi tanpa mendaftar.



Gambar 4. Arsitektur *QR Code* sisi Aplikasi

4. Hasil Dan Pembahasan

Pada bagian ini, dibahas dan dianalisis bagaimana *QR code* dapat digunakan dalam serangan *phishing*, dengan menjadikannya sebagai vektor serangan yang ampuh.

Eksperimen dimulai dengan membuat *QR code* palsu (berbahaya) dan halaman *website* berisi *phishing*. Selanjutnya, diikuti dengan percobaan serangan melalui *QR code* tersebut dan diujicobakan menggunakan browser mode aman (seperti in-Private pada Microsoft Edge, incognito pada Chrome), untuk menjaga *QR code* berbahaya tetap berfungsi bahkan jika ditandai sebagai berbahaya.

Di bawah ini adalah tahapan-tahapan dalam ekperimen:

4.1 Membuat QR Code

Langkah pertama dalam eksperimen adalah membuat *QR code* seperti pada gambar 5, dengan URL berbahaya yang mengarahkan pengguna ke *situs phishing*.

Persyaratan serangan *phishing* yang berhasil adalah *domain name* yang mirip dengan *website* asli untuk menipu pengguna dan manajemen *link* untuk menyamarkan *domain name* asli.

Gambar 5. *QR Code* Eksperimen

Yang terakhir adalah metode umum untuk menipu pengguna jika *domain name* tidak mirip dengan aslinya agar *QR code* palsu tetap beroperasi. Dalam eksperimen digunakan alamat menggunakan bit.ly, layanan manajemen *link* yang memungkinkan pengguna mempersingkat URL agar lebih menarik dan praktis. *Intruder* sering menggunakan layanan tersebut sebagai metode penyamaran untuk mengelabui orang agar mengunjungi situs jahat. Karena *link* yang dipersingkat menggantikan yang asli, pengguna tidak dapat menemukan tujuan tanpa terlebih dahulu mengunjungi *website*.

Selain itu, beberapa *QR code reader* memungkinkan pengguna untuk melihat konten *QR code* yang dapat dibaca manusia (misalnya URL) sebelum melakukan eksekusi, sementara *code reader* lainnya mengalihkan tanpa tindakan perantara ini. Untuk mengeksekusi serangan tersebut, halaman asli diduplikasi dan kemudian dimodifikasi.

4.2 Membuat *Website* Palsu untuk *Phishing*

Dalam skenario ini, *website resmi* diduplikasi dengan membuat *website* palsu untuk keperluan *phishing*. *Website* ini digunakan untuk mengakses akun resmi korban. *Website* palsu digunakan sebagai *website phishing* dan untuk menyimpan data penting (*database)* yang diambil.

Gambar 6. *Phishing WebPage*

4.3 Melakukan Serangan

Pada tahap ini dilakukan serangan terhadap *database* asli. *Browser* akan membuka dan mengarahkan pengguna ke *website phishing*. Pengguna yang tidak curiga, menambahkan data-data yang diminta dan ketika mengetuk tombol "*next"* akan dialihkan dari halaman resmi tanpa melihat sesuatu yang mencurigakan. Data-data penting yang disimulasikan disimpan di dalam *database* yang telah dibuat.

4.4 Menggunakan Browser Aman

Microsoft Edge memiliki mode *"Private Mode*", demikian juga Chrome memiliki *incognito* sebagai mode *browser* yang dianggap aman. *Browser* mode ini mampu mempersulit pekerjaan *intruder*. Beberapa jam setelahnya atau dengan waktu yang telah ditentukan, *website* tersebut dianggap menipu oleh fitur "*safe browsing*". Dengan menggunakan beberapa baris *source code* serangan, penanggulangan yang dilakukan oleh *browser* tersebut dapat dilewati.

Model *by-pass* yang bisa digunakan oleh *intruder* adalah dengan mengubah nama halaman *phishing*. Namun, pada kenyataannya, itu tidak nyaman bagi *intruder* karena mereka harus

mengubah URL di *QR code* berbahaya tersebut. *Window* notifikasi *pop-up* memperingatkan bahwa "Halaman *website* di (URL) ini telah dilaporkan sebagai penipu dan telah diblokir berdasarkan preferensi keamanan Anda" yang sebenarnya berarti bahwa Microsoft Edge hanya menandai halaman (file) *phishing* yang sebenarnya sebagai berbahaya, dan bukan nama domain.

Praktik ini membantu *web developer* untuk mengidentifikasi *website* yang disusupi; atau *domain name* akan ditandai sebagai *domain* berbahaya. Perlu dicatat bahwa memasukkan *domain name* ke dalam daftar hitam bukanlah *best practice* karena *domain name* berbahaya saat ini mungkin tidak berbahaya setelah jangka waktu tertentu. Dalam praktiknya, halaman *phishing* harus menjadi halaman pertama yang dilihat seseorang saat diarahkan ke *website* tersebut. *Intruder* dapat mengalihkan korban dari halaman utama kosong ke halaman *phishing* dengan satu baris PHP secara otomatis. Oleh karena itu, digunakan halaman utama (misalnya index.php) hanya untuk mengarahkan pengguna ke sub halaman *phishing*. Korban masih akan melihatnya sebagai halaman utama karena *header* kode PHP berikut yang mengalihkan korban secara otomatis ke halaman *phishing* tanpa pesan peringatan muncul lagi.

5. Kesimpulan

Pada makalah ini dijelaskan bahwa *QR Code* dimanfaatkan oleh *intruder* sebagai media *phishing* kepada calon korban dengan memanfaatkan celah seperti membelokkan vektor serangan. Untuk memperkuat keamanan aplikasi *QR Code*, diusulkan Arsitektur keamanan sebagai anti *phishing* agar *QR Code* tidak mudah dieksploitasi.

Arsitektur yang dibuat berbasis server dan aplikasi menggunakan fungsi *hash* dan *digital signature*. Fungsi *hash* sebagai penopang sistem kriptografi dan *digital signature* untuk mengautentikasi pengguna *QR Code*. Target penerapan QR Code adalah pemakai *smartphone*. Percobaan yang disimulasikan menggunakan *QR code* yang telah disusupi *malicious software* dapat mengungkapkan kerentanan yang menyebabkan pengungkapan informasi-informasi pribadi secara tidak disengaja.

QR Code yang sudah menggunakan fungsi *hash* dan *digital signature* berhasil menggagalkan *serangan QR code* berbahaya pada fase *scanning* awal dengan memverifikasi *QR code* asli secara digital.

Analisis keamanan dari model ini menunjukkan bahwa selain mencegah pengguna dari pengalihan ke situs berbahaya oleh QR code berbahaya, juga efektif terhadap serangan MITM (Man In The Middle) Attack. Model ini mudah diimplementasikan karena hanya memerlukan sedikit modifikasi dari perspektif penerapan *QR code*.

Daftar Pustaka

[1] Lin, Pei-Yu & Chen, Yi-Hui. (2017). High payload secret hiding technology for QR codes. EURASIP Journal on Image and Video Processing. 2017. 10.1186/s13640-016-0155-0.

[2] Reinfelder, Lena. (2019). User Interaction with Smartphone Security and Privacy Mechanisms.

[3] Prasetia, Afrizal & Fairuzabadi, Muhammad & Wardani, Setia. (2022). Aplikasi Berbagi Kontak Menggunakan QR Code Untuk Smartphone Android. APPLIED SCIENCE AND TECHNOLOGY REASERCH JOURNAL. 1. 26-31. 10.31316/astro.v1i1.3209.

[4] Senthil, V. & Margam, Madhusudhan. (2019). Application of Quick Response (QR) Code and its Usefulness in Library Services.

[5] Kieseberg, Peter & Schrittwieser, Sebastian & Leithner, Manuel & Mulazzani, Martin & Weippl, Edgar & Munroe, Lindsay & Sinha, Mayank. (2012). Malicious Pixels Using QR Codes as Attack Vector. 10.2991/978-94-91216-71-8_2.

[6] Kompas.com (2023) diakses dari https://tekno.kompas.com/read/2023/02/13/19300087/pengguna-internet-di-indonesia-tembus-212-9-juta-di-awal-2023?page=all, accessed 4 April 2023

[7] Jajoo, Akshay. (2021). A study on the Morris Worm.

[8] Slamet, S. (2022). Pertahanan Serangan Social Engineering Menggunakan Two Factor Authentication (2FA) Berbasis SMS (Short Message System). SPIRIT, 14(2).

[9] Kareem, Fairoz & Ameen, Siddeeq & Ahmed, Awder & Salih, Azar & Ahmed, Dindar & Kak, Shakir & Najat, Zryan & Yasin, Hajar & Mahmood, Ibrahim & Omar, Naaman. (2021). SQL Injection Attacks Prevention System

Technology: Review. Asian Journal of Research in Computer Science. 10.9734/AJRCOS/2021/v10i330242.

[10] Denso Wave (2023) diakses dari https://www.denso-wave.com/en/technology/vol1.html, accessed 2 February 2023

[11] Firmansyah, Guntur & Hariyanto, Didik. (2019). The use of QR code on educational domain: a research and development on teaching material. Jurnal SPORTIF : Jurnal Penelitian Pembelajaran. 5. 265. 10.29407/js_unpgri.v5i2.13467.

[12] Srinounpan, Bamrung & Srinounpan, Chawanrat & Sumethokul, Patcharee & Patwary, Ataul. (2020). The Application of QR Code Technology to Create the Value-Added Products for The Baan Klong Peek Neur Beehive Community Enterprise Group at Tambon Suankhan, Nakhon Si Thammarat Province. Systematic Reviews in Pharmacy. 11. 519-528.

[13] Albastroiu Nastase, Irina & Felea, Mihai. (2015). Exploring the potential of QR codes in higher education considering the attitudes and interests among Romanian students. 10.12753/2066-026X-15-029.

[14] Pasa, Ike & Zamzami, Fuad. (2019). Analisis Pengembangan Fitur Obrolan Baru Berbasis Scan QR Code Pada Aplikasi Paziim. INTEK : Jurnal Informatika dan Teknologi Informasi. 2. 17-25. 10.37729/intek.v2i1.85.

[15] Lerner, Adam & Saxena, Alisha & Ouimet, Kirk & Turley, Ben & Vance, Anthony & Kohno, Tadayoshi & Roesner, Franziska. (2015). Analyzing the Use of Quick Response Codes in the Wild. 359-374. 10.1145/2742647.2742650.

[16] Le-Nguyen, Minh-Khoi & Nguyen, Tri-Chan-Hung & Le, Thuan & Nguyen, Van-Hoa & Phuoc, Ton & Nguyen-An, Khuong. (2022). Phishing Website Detection as a Website Comparing Problem. SN Computer Science. 4. 10.1007/s42979-022-01544-9.

[17] Loxdal, Joakim & Andersson, Måns & Hacks, Simon & Robert, Lagerström. (2021). Why Phishing Works on Smartphones: A Preliminary Study. 10.24251/HICSS.2021.863.

[18] Kata Data Media Network (2023) diakses dari https://databoks.katadata.co.id/datapublish/2022/07/17/mayoritas-warga-ri-tidak-pasang-antivirus-di-gadget, accessed 3 Maret 2023

[19] Guerar, Meriem & Migliardi, Mauro & Palmieri, Francesco & Verderame, Luca & Merlo, Alessio. (2019). Securing PIN-based Authentication in Smartwatches With just Two Gestures. Concurrency and Computation Practice and Experience. 32. 10.1002/cpe.5549.

[20] Sharma, Tejpal & Rattan, Dhavleesh. (2023). Android Malwares with Their Characteristics and Threats. 10.1007/978-981-19-7982-8_1.

[21] Mugisha, David. (2019). Android Application Malware Analysis. International Journal of Mobile Learning and Organisation. 12.

[22] Zhang, Qiujian & Wang, Xiaomei. (2009). SQL Injections through Back-End of RFID System. 1 - 4. 10.1109/CNMT.2009.5374533.

[23] Alghawazi, Maha, Daniyal Alghazzawi, and Suaad Alarifi. 2022. "Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review" Journal of Cybersecurity and Privacy 2, no. 4: 764-777. https://doi.org/10.3390/jcp2040039

[24] Mitropoulos, Dimitris & Spinellis, Diomidis. (2017). Fatal injection: A survey of modern code injection attack countermeasures. PeerJ Computer Science. 3. e136. 10.7717/peerj-cs.136.

[25] Soleymanzadeh, Raha, Mustafa Aljasim, Muhammad Waseem Qadeer, and Rasha Kashef. 2022. "Cyberattack and Fraud Detection Using Ensemble Stacking" AI 3, no. 1: 22-36. https://doi.org/10.3390/ai3010002

[26] Bhavsar, Vaishnavi & Kadlak, Aditya & Sharma, Shabnam. (2018). Study on Phishing Attacks. International Journal of Computer Applications. 182. 27-29. 10.5120/ijca2018918286.

[27] Yao, Huiping & Shin, Dongwan. (2013). Towards preventing QR code based attacks on android phone using security warnings. ASIA CCS 2013 - Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. 341-346. 10.1145/2484313.2484357.

[28] Aldawood, Hussain & Skinner, Geoff. (2020). Analysis and Findings of Social Engineering Industry Experts Explorative Interviews: Perspectives on Measures, Tools and Solutions. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2983280.