



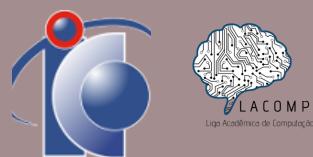
# Info Hacker

RESUMO QUINZENAL OFICIAL DA ACADEMIA HACKER



## 01: Intrusion Detection System (IDS)

## 02: Como funciona: QR Code



# Intrusion Detection System (IDS)

por Everton Borges

É um dispositivo de rede dedicado, ou uma das várias ferramentas em um servidor ou firewall que varre os dados em um banco de dados de regras ou de assinaturas de ataques, à procura de tráfego mal-intencionado. Se uma correspondência for detectada, o IDS registrará a detecção de registro e criará um alerta para um administrador de rede. O sistema de detecção de invasão não age quando uma correspondência é detectada, então não impede os ataques. O trabalho do IDS é apenas detectar, registrar e relatar.

A varredura realizada pelo IDS deixa a rede mais lenta (conhecido como latência). Para se prevenir contra o atraso de rede, um IDS é geralmente colocado off-line, separado do tráfego de rede regular. Dados são copiados ou espelhados por um switch e então encaminhados para o IDS para a detecção off-line. Também existem ferramentas de IDS que podem ser instaladas sobre o sistema operacional do computador, como o Linux ou Windows.



## Referências:

- <https://www.netacad.com/courses/cybersecurity/introduction-cybersecurity>

# Como funciona: QR Code

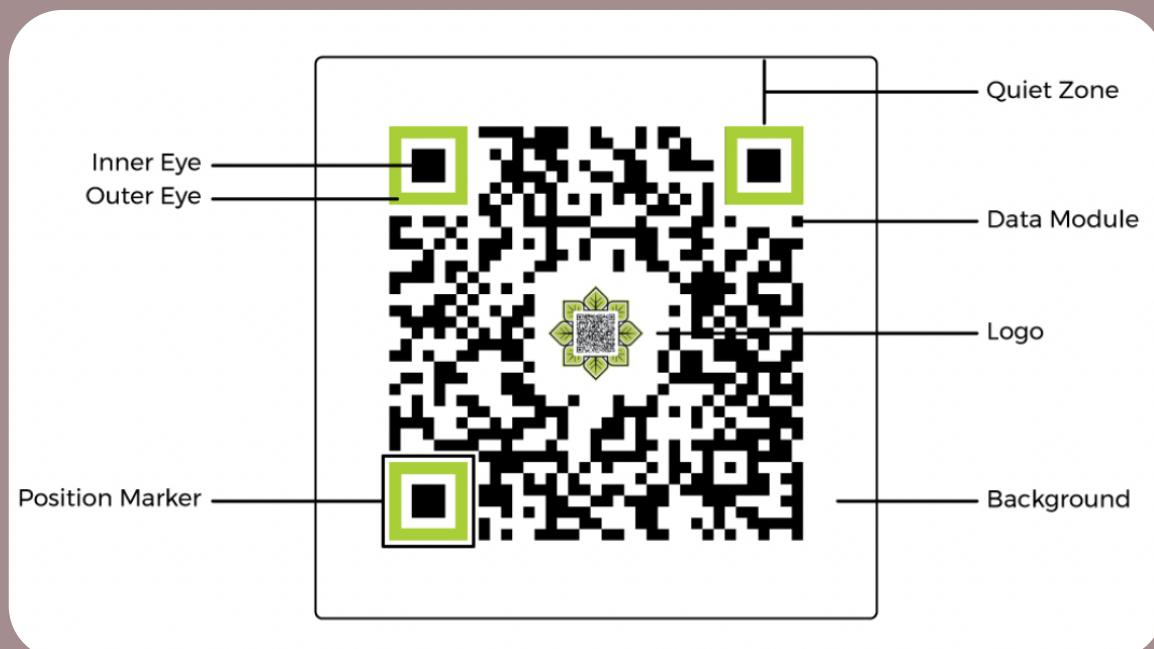
por Yanka Ribeiro

Eles estão em todos os lugares e são usados para (quase) qualquer coisa. Mas, se um parente seu perguntasse como eles funcionam você saberia explicar? Para os que responderam “vish”, ou algo similar, vocês aprenderão nesta leitura o que é e como funciona um QR Code. Segue o fio.

Um **Quick Response Code**, ou Código de Resposta Rápida, é um tipo de código de barras 2D criado por desenvolvedores japoneses em 1994, que permite acesso à informação codificada depois que é lido por um dispositivo óptico. Naquela época, percebeu-se um gargalo no uso de códigos de barras convencionais: o limite de 20 caracteres alfanuméricos. Nesse tipo de código, os dados são codificados levando em consideração a largura e a distância entre as linhas verticais e a medida que o mercado japonês crescia, com novos produtos chegando às prateleiras, 20 caracteres já não era suficiente para armazenar todas as informações necessárias. Além disso, a posição única de leitura frequentemente resultava em erros. Foi nesse contexto, que a *Denso Wave* decidiu agir e criar um novo tipo de código de barras que pudesse guardar mais dados e ser lido em qualquer orientação.

Agora, os dados originais serão convertidos em uma combinação de quadrados. E os quadrados mais importantes são:

- **Módulo de informação (data module)**: A unidade básica de um QR Code. Embora customizável, a construção ótima de um QR Code, e mais utilizada, é um quadrado preto em um fundo branco.



- **Marcador de posição (position marker)**: Permite que os scanners e câmeras percebam a orientação de leitura e localizem os módulos de informação mais rápida e precisamente. Há 3 marcadores de posição em um QR Code e cada um é constituído pelos chamados *inner* e *outer eyes*.
- **Quiet zone**: Área em branco que circunda a matriz de módulos. Permite que os dispositivos ópticos identifiquem onde começa e onde termina o QR Code.

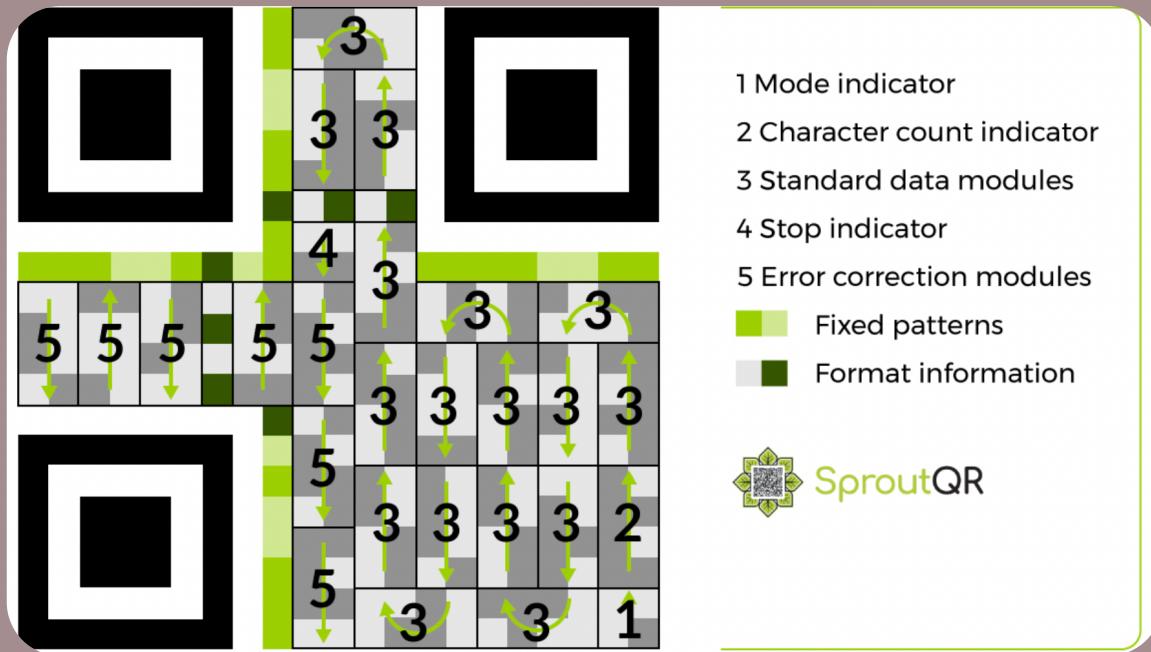
O menor tamanho de um QR code é uma matriz de 21 linhas e 21 colunas, ou seja, é a versão 1. Uma matriz 25x25 é versão 2 e assim por diante. O maior tamanho possível, versão 40, contém 177 linhas e 177 colunas e nesta versão é possível armazenar 7.089 caracteres numéricos ou 2.953 alfanuméricos. Também é possível armazenar bytes e kanji, mas são menos frequentes. Na prática, um QR Code pode armazenar qualquer coisa que use números, letras, pontuação e símbolos, o que explica o uso crescente em variados nichos de mercado, produtos e até mesmo transações bancárias.

Importante ressaltar que o tamanho mínimo geralmente refere-se à distância de leitura e não ao tamanho do arquivo. Então quanto mais caracteres codificados, maior e mais complexo um QR code se torna, certo? Sim e não. Essa é a diferença de um código estático e um dinâmico, que é explicada com mais detalhes no terceiro link de referência deste artigo.

Por fim, chegamos ao processo de leitura. O leitor de QR code começa sempre pelo canto inferior direito, depois movendo-se dois módulos de informação para cima até atingir o primeiro marcador de posição. Então, ele lê mais dois módulos à esquerda e vai para baixo. Esse zig-zag é repetido até todos os módulos de informação serem lidos. Para facilitar o entendimento, a leitura foi descrita em 6 passos básicos explicados a seguir.

1. Celular é apontado para o QR code.
2. O leitor de QR code existente na câmera do celular reconhece os 3 marcadores de posição e a zona quieta. Agora, o leitor está ciente dos limites do QR code.
3. O leitor começa a ler pelo canto inferior direito, onde está o indicador de modo. Esses 4 módulos de informação indicam o tipo de dado codificado (numérico, alfanumérico, byte, ou kanji).
4. Em seguida, o indicador de quantidade de caracteres é lido. Um agrupamento de 8 módulos que estão logo após o indicador de modo. E como você pode deduzir facilmente, este indicador informa quantos caracteres no total foram codificados.
5. Sabendo o tipo e tamanho do dado final, o leitor de QR code continua a processo de leitura fazendo um zig-zag na matriz até atingir o *end indicator*, significando que toda informação foi lida.

6.. Após a leitura do caracter final, o leitor segue para os módulos de informação responsáveis pela correção de erros. Nesses módulos estão 1 de 4 níveis de correção ou o quanto foi codificado para servir de backup em caso de erro.



E assim funciona um QR Code. Certamente há mais detalhes na codificação que não foram cobertos nesse texto, principalmente questões de segurança envolvendo geradores online de QR code. Mas, como o objetivo aqui é despertar a curiosidade, deixo a cargo do digníssimo leitor buscar esta informação.

### **Referências:**

How do QR Codes work, por QR Code Tiger

How do QR Codes work, por Sprout QR

What is a dynamic QR Code, por Scanova

## **Info Hacker - Edição 012**

### **26/07/2022**



#### **Autores:**

Everton Borges  
Yanka Ribeiro

#### **Revisão**

Everton Borges  
Bruno Severo



acha.ufal

## **Edições Anteriores**

<https://github.com/Academia-Hacker/Info-Hacker>