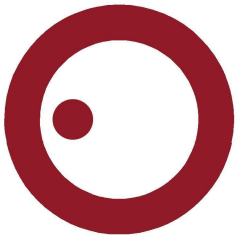


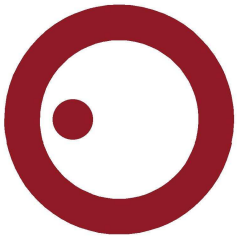
escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Análise Forense

TESTE E VALIDAÇÃO DE AMEAÇAS INFORMÁTICAS EM CONTEXTO DE ENSINO SUPERIOR

ESTUDANTE LUÍS MIGUEL BABO COSTA

Leiria, março de 2025



escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Análise Forense

TESTE E VALIDAÇÃO DE AMEAÇAS INFORMÁTICAS EM CONTEXTO DE ENSINO SUPERIOR

ESTUDANTE LUÍS MIGUEL BABO COSTA
Número: 2222886

Estágio realizado sob orientação do Professor Doutor Paulo Jorge Ferreira Batista Pinheiro Cordeiro (paulo.cordeiro@ipleiria.pt).

Leiria, março de 2025

AGRADECIMENTOS

Gostaria de expressar os meus agradecimentos a todos os que contribuíram para a realização do meu estágio e elaboração deste documento.

Em primeiro lugar, gostaria de agradecer ao Instituto Superior Politécnico Gaya pelo acolhimento e pela oportunidade de realizar o estágio, que me proporcionou o ambiente ideal para a minha aprendizagem e experiência.

Ao meu orientador por parte do ISPGaya, o Professor Doutor Justino Lourenço, por toda a ajuda e disponibilidade na superação de desafios e nos projetos em que estive envolvido.

Gostaria de agradecer também ao Instituto Politécnico de Leiria e ao meu coordenador de estágio, o Professor Doutor Paulo Jorge Ferreira Batista Pinheiro Cordeiro, por toda a ajuda e disponibilidade em momentos de dificuldade e dúvidas.

Por fim, gostaria de agradecer à minha família e amigos pelo apoio constante e incansável. Tornam toda esta caminhada mais fácil.

Muito obrigado a todos.

RESUMO

Ao longo dos últimos anos as tecnologias *Wi-Fi* e *QR Codes* têm vindo a aumentar a sua popularidade e utilização pelos utilizadores em todo o mundo, devido à sua facilidade de implementação e utilização, baixo custo e velocidades elevadas, bem como a possibilidade de tecnologias com capacidade de leitura de *QR Codes* por parte dos utilizadores. Esta facilidade de acesso e utilização destas tecnologias naturalmente apresenta riscos de segurança para os utilizadores.

Este relatório explora as tecnologias *Wi-Fi* com utilização do protocolo de segurança *WPA-Enterprise*, amplamente utilizado em ambientes corporativos e educacionais e os seus possíveis ataques. Foram realizados ataques do tipo *Evil Twin* e *deauthentication*. O segundo projeto explora vulnerabilidades resultantes da utilização de *QR Codes* como ataques de *phishing*, execução de *scripts* e execução remota de código. Ambos os projetos foram realizados em contexto académico, nas instalações do ISPGAYA.

ABSTRACT

Over the last few years, Wi-Fi and QR Code technologies have been increasing in their popularity and use by users all over the world, due to their ease of implementation and use, low cost and high speeds, as well as the the possibility of technologies capable of reading QR Codes by users. This ease of access and use of these technologies naturally raises security risks for users.

This report explores Wi-Fi technologies using the WPA-Enterprise security protocol, which is widely used in corporate and educational environments, and their possible attacks. Evil Twin and deauthentication attacks were carried out. The second project explores vulnerabilities resulting from the use of QR Codes such as phishing attacks, script execution and remote code execution. Both projects were carried out in an academic context at the ISPGAYA facilities.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

api	Application Programming Interface.
bit	Digito binário.
Byte	Unidade de informação digital composta por oito bits.
CODEC	COmpression/DECompression.
CPU	Central Processing Unit.
DHCP	Dynamic Host Configuration Protocol.
DOS	Denial of Service.
EAP	Extensible Authentication Protocol.
GB	Gigabyte.
IDS	Intrusion Detection System.
IP	Internet Protocol.
IPLeiria	Instituto Politécnico de Leiria.
ISP	Internet Service Provider.
ISPGAYA	Instituto Superior Politécnico Gaya.
MAC	Media Access Control.
MD5	Message Digest 5.
MITM	Man in the Middle.
PEAP	Protected Extended Authentication Protocol.
QR	Quick Response.

RADIUS	Remote Authentication Dial-In User Service.
sdn	Software Defined Network.
SO	Sistema Operativo.
SSID	Service Set Identifier.
TLS	Transport Layer Security.
TTLS	Tunneled Transport Layer Security.
USB	Universal Serial Bus.
Wi-Fi	Wireless Fidelity.
WPA	Wi-Fi Protected Access.
XSS	Cross-Site Scripting.

1

INTRODUÇÃO

O presente documento apresenta os resultados obtidos durante o estágio na instituição ISPGAYA no período de 27 de novembro de 2023 até 26 de julho de 2024. Este estágio foi realizado como cadeira integrante do curso de Mestrado em Cibersegurança e Informática Forense pelo IPLeiria.

O principal objetivo deste relatório é descrever as atividades desenvolvidas durante este período, destacando os conhecimentos e experiência adquiridos, os desafios enfrentados e as suas soluções, com as suas contribuições para o desenvolvimento profissional e pessoal.

Este estágio teve como objetivo atacar a rede Wi-Fi do ISPGAYA, bem como realizar atividades de *social engineering* através da utilização de QRCodes, tendo sido dividida em diversas etapas.

A rede Wi-Fi do ISPGAYA utiliza autenticação Extensible Authentication Protocol (EAP) através de servidor Radius, com introdução de nome de utilizador e password, permitindo assim a centralização dos serviços de autenticação, autorização e contabilização. Ao aceder o utilizador necessita de aceitar o certificado emitido pelo servidor Remote Authentication Dial-In User Service (RADIUS) da instituição. Utilizadores que não possuam conhecimento ou sendo realizada uma má configuração podem expor inconscientemente as suas credenciais a atacantes que configurem *Rogue Access Points*, também designados por *Evil Twins*, ou utilizem algum mecanismo man-in-the-middle (Palamà et al., 2023). Sendo bem sucedido, um atacante poderia ter acesso à rede Wi-Fi da instituição, credenciais da vítima bem como acesso à plataforma utilizada pelos estudantes (inforestudante) aos seus dados pessoais ou, tendo acesso à plataforma utilizada pelos docentes da instituição (infordocente) poderia ter acesso a notas de alunos bem como os seus dados pessoais.

Quick Response (QR) Codes surgiram na indústria automóvel e têm vindo a aumentar a sua popularidade ao longo dos anos fora desta indústria devido à sua rapidez de leitura e capacidade de armazenamento de informação (Sharma, 2012). Estes podem ser utilizados para armazenar informações numéricas, alfanuméricas, byte ou binárias e kanji - sistema de escrita da língua japonesa. Desta forma, é

possível identificar algumas possíveis ameaças, tais como campanhas de *phishing* e execução de código malicioso no dispositivo que realiza a leitura do código.

1.0.1 *Objetivos*

Os objetivos do estágio foram explorar estas duas tecnologias e também as suas vulnerabilidades e alguns possíveis vetores de ataque, bem como analisar a resposta a este tipo de ataques em ambiente académico. Assim, este estágio foi dividido em dois projetos: o primeiro a explorar vulnerabilidades Wi-Fi, mais concretamente no mecanismo de segurança *WPA-Enterprise*, através da criação de *Evil Twins* e ataques de *deauthentication*, na tentativa de obter as credenciais dos alunos do ISPGAYA. O segundo projeto tinha como intuito tirar proveito dos *QR Codes* para abrir um *website* com dados fictícios que executasse *scripts*, como campanha de *phishing* ou até mesmo de execução remota de código.

1.0.2 *Metodologia*

A metodologia utilizada começou por uma pesquisa bibliográfica nas bases de dados científicas acerca dos temas de ambos os projetos seguida de uma abordagem experimental nas instalações do ISPGAYA: primeiro projeto voltado para Wi-Fi *WPA-EAP* e utilização de *Evil Twins* para tentar obter credenciais e o segundo projeto voltado para a utilização maliciosa de *QR Codes*, direta aos utilizadores do ISPGAYA (docentes e alunos) - com suporte de várias ferramentas de análise de risco e vulnerabilidade presentes na rede do ISPGAYA que serão detalhadas mais adiante.

1.0.3 *Contribuições*

Estes trabalhos contribuiram para uma compreensão mais aprofundada de algumas fragilidades com a utilização do protocolo *WPA-Enterprise*, especialmente o fator humano, pois apesar de poder ser um bom método de segurança, através da utilização bem sucedida de ataques do tipo *Evil Twin* é possível ter acesso às credenciais dos utilizadores e assim conseguir acesso à rede e escalar o ataque.

No que toca ao projeto com os *QR Codes* também é possível reparar na falha do fator humano, podendo existir uma iliteracia digital no que toca a certos perigos que

possam envolver os *QR Codes*, como execução automática de código no dispositivo da vítima. Estes trabalhos contribuíram também para oferecer a outras pessoas um ponto de análise e estudo acerca do trabalho adotado neste projeto:

- Processo de criação de *Evil Twins* e ataques de *deauthentication*;
- Utilização de *QR Codes* como forma de *phishing* e vetor de ataque para potenciais vítimas;
- Alertar utilizadores dos perigos destas possíveis ameaças para assim poderem mitiga-las;
- Documento em formato de relatório com as informações relativas a este trabalho realizado;

1.0.4 *Estrutura do documento*

Este documento encontra-se dividido em 4 partes. Introdução, que dá uma breve contextualização do trabalho realizado, enquadramento/contexto que permite dar informação útil acerca do trabalho realizado e/ou trabalho semelhante já realizado. Desenvolvimento que detalha os projetos realizados e por fim a conclusão que termina este documento com os resultados obtidos e as suas reflexões.

2

TRABALHO RELACIONADO

Existe algum trabalho realizado na área, especialmente com redes Wi-Fi protegidas por WPA2, ou seja numa perspetiva residencial/pessoal. Em contexto empresarial/institucional WPA-Enterprise destaco o trabalho realizado (Palamà et al., 2023), onde foram realizados ataques do tipo Evil Twin para tentar obter credenciais de estudantes, em ambiente de Wi-Fi Eduroam, algo semelhante ao explorado neste trabalho. Também o trabalho de (Cassola et al., 2013) realiza uma abordagem a este tema em que é utilizada uma antena para interferir com o sinal de Wi-Fi, é criado um Evil Twin e é estrategicamente colocado para ser mais apelativo aos utilizadores para se conectarem ao mesmo, que serão analisados com mais detalhe mais à frente. No que toca à área dos *QR Codes*, destaco o trabalho Kharraz et al. (2014), em que os autores analisam um grande número de *websites* que contêm códigos *QR* e verificam se estes contêm conteúdo malicioso e em que cenários se verificavam. Também destaco o trabalho realizado por Vidas et al. (2013) em que os autores criam posters com *QR Codes* por locais públicos de Pittsburgh e pelo campus da universidade de Carnegie Mellon, explorando os comportamentos dos utilizadores.

Wi-Fi

Wi-Fi é a tecnologia de redes sem fio (WLAN) que permite a comunicação de dispositivos e que se conectem à internet, pertencendo e estando certificados pela marca *Wi-Fi Alliance* e que utilizam o padrão ou os protocolos IEEE 802.11. Estas redes têm vindo a ser mais desenvolvidas e de forma mais rápida devido ao seu baixo custo, velocidades elevadas e mobilidade (Catur Bhakti et al., 2007), sendo também uma grande alternativa a redes fixas (ligadas por cabo), em que uma pessoa estaria limitada a um determinado espaço (Khasanova, 2021). É possível encontrar dispositivos que contenham *chipsets* Wi-Fi em *laptops*, *tablets*, *smartphones*, entre outros. Contêm os protocolos que são implementados em Wi-Fi, que asseguram qualidade de serviço e segurança (AlQahtani e AlOraini, 2012), sendo o de segurança WPA. Este pode ser dividido em duas categorias: WPA *Personal* ou PSK (Pre-Shared Key) e WPA *Enterprise*. O primeiro utiliza um *secret* partilhado para

autenticação e um dispositivo para gestão de credenciais de utilizador, o que pode ser algo inseguro, pois não se trata de um *secret* distinto para cada utilizador, ou seja, um atacante conseguindo acesso à chave, conseguiria obter acesso à rede (AlQahtani e AlOraini, 2012). O WPA *Enterprise* utiliza o protocolo e servidor RADIUS para autenticação e distribuição de chaves, tendo cada utilizador uma chave diferente e nome de utilizador, fora da camada de enlace de dados. (AlQahtani e AlOraini, 2012; Rubens et al., 2000)

Com este crescimento, vantagens e maior utilização, aumentou também a exploração de vulnerabilidades dos dispositivos Wi-Fi. Atacantes procuram-nas continuamente, podendo conceder-lhes acesso à rede (Khasanova, 2021), com o objetivo de comprometer a privacidade, integridade e confidencialidade de dados (A. e Kshirsagar, 2023). Os ataques mais comuns a redes Wi-Fi são *Man-in-the-middle*, *Denial of Service*, *Evil Twin*, uma vez que não necessitam de muito *hardware* e *software* e, em alguns casos, o atacante não necessita de estar conectado à rede Wi-Fi (Khasanova, 2021). Os atacantes também decidem focar-se, grande parte das vezes, nos utilizadores, uma vez que são o fator mais vulnerável, não tendo conhecimento acerca das vulnerabilidades e ameaças existentes (Esser, 2017).

O ataque DOS é considerado um dos ataques mais perigosos, pois causa grandes impactos financeiros nas empresas. Este ataque tem como objetivo, como o nome indica, interromper as comunicações da rede através de uma sobrecarga de pedidos, que pode ser conseguido através do envio de pedidos de *Deauthentication* e *Disassociation* (Khasanova, 2021)

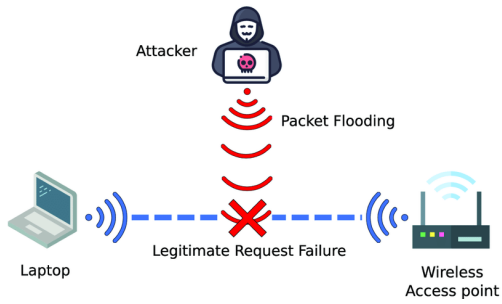


Figura 2.1: *Denial Of Service* - ilustração (Staddon et al., 2021)

Os ataques MITM acontecem quando um atacante atua como intermediário nas comunicações entre dois pontos na rede, alterando-as, onde consegue ter acesso às comunicações dos mesmos, pode também alterá-las. Grande parte dos ataques MITM são acompanhados de outros ataques como DOS, onde o Acess Point legítimo é afetado e faz com que as vítimas se liguem ao *Evil Twin* (Khasanova, 2021).

Os métodos mais comuns de ataques MITM são: ARP-spoofing, MAC-spoofing, DHCP-spoofing e DNS-spoofing (Khasanova, 2021).

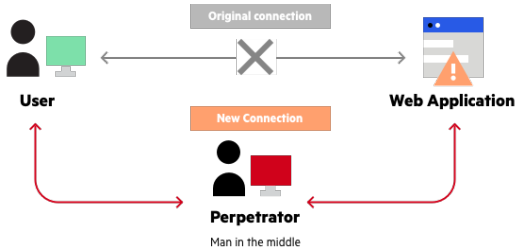


Figura 2.2: *Man-in-the-Middle* - ilustração (Maharjan, 2020)

O ataque *Evil Twin* é uma ameaça às redes Wi-Fi bastante conhecida (Palamà et al., 2023) em que um atacante simula um *Access Point* legítimo ao replicar o *BSSID*, *SSID*, *Channel*, endereço IP do *access point* original (Modi e Parekh, 2017; Shrivastava et al., 2020) com o objetivo de obter os dados de autenticação das vítimas (Asaduzzaman et al., 2020; Briones et al., 2013; Syahrial et al., 2024) bem como outras informações pessoais (Kuo et al., 2018). Isso ocorre pois todos os *access points* enviam *beacons* que contêm estas informações (Modi e Parekh, 2017) e o atacante replica-as no seu *access point* falso. Para tornar este ataque ainda mais eficiente, um atacante pode colocar-se mais próximo das suas vítimas para que este sinal seja mais forte e/ou ao desativar mecanismos encriptação, incentivando as potenciais vítimas a ligarem-se ao seu *AP*. Os utilizadores não conseguem distinguir os dois *access points* (legítimo e *evil twin*) e acabam por se ligar ao que tenha melhor sinal e com menor nível de encriptação, conectando-se ao *access point* malicioso (Palamà et al., 2023), podendo resultar em perdas de informação significativas (Kumar e Paul, 2016).

Este ataque pode ser realizado em conjunto com ataques de *de-auth* (DOS) no sentido de obrigar os utilizadores a conectarem-se ao *AP* do atacante. A falta de conhecimento de muitos utilizadores em relação a redes Wi-Fi inseguras auxilia bastante a que este tipo de ataque seja bem sucedido, com o atacante a conseguir adquirir os dados de acesso das suas vítimas, observar ou manipular o seu tráfego ou até mesmo obter controlo dos seus dispositivos (Palamà et al., 2023). O atacante tendo acesso à rede legítima pode também fornecer acesso à internet às suas vítimas realizando *bridge* entre o *access point* legítimo ou outro provedor de serviço e o seu *evil twin*, conseguindo assim realizar ataques do tipo MITM e intermediar o tráfego entre a vítima e o *AP* legítimo (Modi e Parekh, 2017). (Kuo et al., 2018)

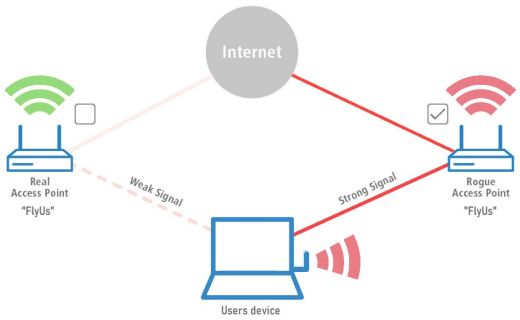


Figura 2.3: Exemplo de ataque Evil Twin/Rogue Access Point

WPA-EAP

WPA-EAP ou *WPA Enterprise* é um mecanismo de segurança para redes Wi-Fi em ambientes empresariais ou de ensino de acordo com o padrão 802.11X, utilizado em protocolos como *IEEE 802.11i* e *IEEE 802.16e* (Catur Bhakti et al., 2007), para proteger redes contra acesso não autorizado (Cassola et al., 2013). Existem dois modelos de *EAP*: *pass through behaviour model* e *multiplexing model*. O primeiro conta com 3 intervenientes distintos - *Supplicant, Authenticator* e Servidor de autenticação, sendo o *supplicant* o dispositivo cliente, o *Authenticator* o *AP* e o servidor de autenticação um servidor *AAA* (*Authentication, Authorization and Accounting*) (Catur Bhakti et al., 2007), que serve como ponto central (FreeRadius, 2024), como *RADIUS* ou *DIAMETER*, sendo o *authenticator* um intermediário na comunicação (Catur Bhakti et al., 2007).

Para se conectar o cliente/utilizador, através de um dispositivo que contenha uma interface de rede que esteja abrangida pelo protocolo 802.11 envia um pedido de conexão para o *Access Point* para se ligar à rede, em que este já recebeu previamente um certificado através do protocolo *RADIUS*. Este responde com um pedido das credenciais de acesso para o utilizador, este insere e são enviadas para o servidor de autenticação. É então criado um canal de comunicação seguro entre o utilizador e o servidor de autenticação (*TLS*) após validação do certificado enviado ao utilizador, como forma de autenticar a rede e o servidor ao cliente (Cassola et al., 2013). São enviadas as credenciais para o servidor *RADIUS* que realiza a autenticação. Conforme estejam corretas ou não, é cedido o acesso à rede ao utilizador ou negado. As mensagens *EAP* transmitidas entre cliente e autenticador/*access point* são do tipo *EAP over LAN (EAPOL)* enquanto que as mensagens entre o *access point* e servidor de autenticação são encapsuladas no formato *RADIUS* (Abo-Soliman e Azer, 2017).

No modelo *multiplexing* apenas existem dois dispositivos distintos, onde o *authenticator* e o servidor de autenticação são um e realizam todos os serviços do servidor de autenticação (Catur Bhakti et al., 2007).

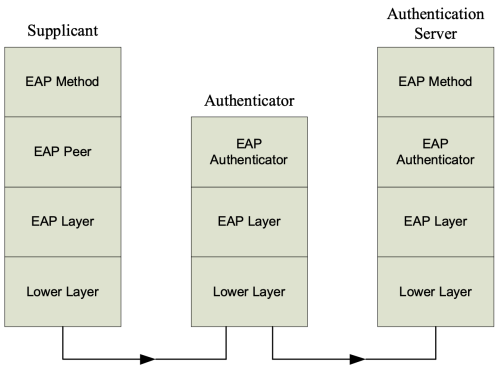


Figura 2.4: *EAP pass-through* (Catur Bhakti et al., 2007)

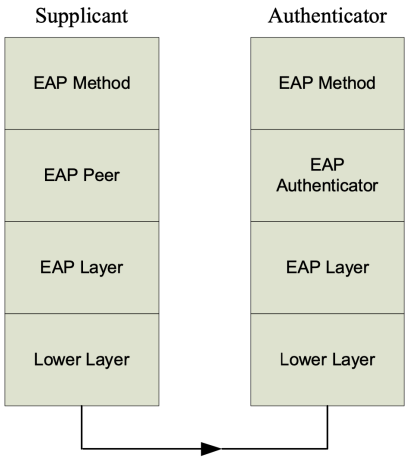


Figura 2.5: *EAP multiplexing* (Catur Bhakti et al., 2007)

Existem também diferentes métodos de autenticação dos utilizadores (Catur Bhakti et al., 2007; Palamà et al., 2023), em que cada utilizador tem o seu nome de utilizador e *password*, estabelecendo sempre um canal de comunicação seguro entre o utilizador e o servidor de autenticação como o *MD5, TLS, TTLS e PEAP* (Abo-Soliman e Azer, 2017).

De acordo com Catur Bhakti et al. (2007):

- *EAP-TLS*, que utiliza o protocolo *TLS* e necessita certificados do servidor e do cliente *PKI* (*Public Key Infrastructure*) com o objetivo de autenticar ambas as partes. Considerado o método mais seguro.

- *EAP with tunneled TLS* (*EAP-TTLS*) apenas necessita de certificado por parte do servidor e pode ser utilizado outro método de autenticação do utilizador como *ID* e *password*. É realizada a autenticação do servidor de acordo com o certificado e de seguida é criado um canal protegido de comunicação para a autenticação do utilizador. Oferece uma forte segurança e evita a complexidade de estabelecer o *PKI*.
- *Protected EAP* (*PEAP*) é semelhante ao anterior, em que necessita apenas do certificado do servidor, realiza a encriptação dos dados de autenticação do utilizador, cria um canal de comunicação seguro (*TLS Tunnel*), oferecendo assim uma forte segurança. A diferença para o protocolo anterior é a menor compatibilidade com métodos e plataformas de autenticação mais antigas.
- *Lightweight EAP* (*LEAP*) que suporta autenticação de ambas as partes e troca dinâmica de chaves entre os mesmos em cada reautenticação - com o objetivo das chaves serem sempre diferentes e não serem intercetadas por algum atacante.
- *EAP* com *MD5* (*EAP-MD5*) utiliza a função de *hashing Message-Digest algorithm 5* (*MD5*) para autenticação dos utilizadores. Considerado um mecanismo fraco devido à autenticação unilateral - apenas o cliente é autenticado perante o servidor. Não utiliza chaves para as sessões. (Catur Bhakti et al., 2007)

Normalmente também é utilizado o *MAC Authentication Bypass (MAB)* com o servidor *RADIUS* para dispositivos que não suportam o padrão 802.11X, como telefones fixos, impressoras, câmeras e *access points*, permitindo conceder acesso à rede através de autenticação de dispositivos (Networks, 2024). Neste caso, o servidor *RADIUS* usa como utilizador e *password* o endereço *MAC* dos dispositivos para autenticação. Um atacante pode alterar o seu endereço *MAC* (*MAC Spoofing*) e assim ter acesso à rede sem necessitar de credenciais de acesso (Kyungroul et al., 2016). Um atacante sabendo algum desses endereços *MAC*, consegue alterar o seu endereço *MAC* (*MAC Spoofing*) para um dos capturados, conseguindo assim ter acesso à rede sem necessitar de se autenticar.

Attacks	EAP-TTLS	EAP-PEAP	EAP-TLS
Dictionary attack	Vulnerable	Vulnerable	Resistant
Evil Twin	Vulnerable	Vulnerable	Resistant
Denial of Service	Vulnerable	Vulnerable	Vulnerable

Tabela 2.1: Comparison of EAP Methods Against Attacks (Abo-Soliman e Azer, 2018)

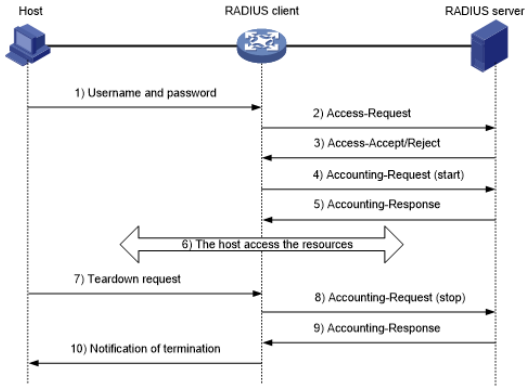


Figura 2.6: *WPA-EAP* com servidor *RADIUS*

Existem vulnerabilidades que podem ser exploradas para este protocolo, tais como ataques *brute force* e de dicionário, que têm como objetivo a obtenção de acesso à rede, sendo o primeiro voltado para autenticações curtas e o segundo que tenta aceder com utilização credenciais previsíveis ou capturadas (Abo-Soliman e Azer, 2018). *Evil Twins/Rogue AP*, *sniffing* e *DoS* (Jung et al., 2022), em que o último afeta todos os métodos de autenticação do protocolo *EAP* (Abo-Soliman e Azer, 2018).

Evil Twins e WPA-EAP

No trabalho abordado por Palamà et al. (2023) é feita a configuração de um *Evil Twin* com o *SSID* da instituição (Universidade de Roma Tor Vergata) e é configurado um servidor *RADIUS* falso para a autenticação dos utilizadores - grupo controlado. Deslocavam-se dentro do raio do *access point* legítimo para tentar que os utilizadores da rede se conectassem ao *evil twin* e a partir daí, pudessem ser colecionadas as suas credenciais. Isto aconteceria porque teriam um maior sinal de conexão, pois estariam mais próximos dos utilizadores e ao tempo de exposição a esta rede maliciosa (deslocavam-se cerca de 300 metros). Uma vez que os utilizadores estariam conectados à rede da instituição, estes também esperavam que se conectassem automaticamente à rede maliciosa, de forma passiva, tendo conseguido alcançar 37 dispositivos móveis diferentes. Foi possível analisar que os dispositivos *Apple/iOS*

não se conectavam automaticamente ao *evil twin* nem enviavam as suas credenciais, uma vez que é necessário o utilizador aceitar o certificado fornecido pelo AP e quando estes se conectam ao *evil twin* é-lhes apresentado o certificado falso para aceitar ou não e estes podem suspeitar e cancelar a conexão ao mesmo. Isto não se verificou para os utilizadores com *android*, até nas versões mais atualizadas dos mesmos. Isto acontece pois o *Android OS* permite aos utilizadores realizar uma configuração mais detalhada da rede podendo especificar protocolos de autenticação e controlo de certificados. O que acontece em alguns casos é a não escolha de nenhuma configuração e aceitar certificados de redes inseguras, tornando estes dispositivos vulneráveis. É de notar que esta primeira experiência foi realizada de forma passiva, não estando a interagir com os seus dispositivos.

Num segundo projeto realizaram as mesmas configurações na Universidade de Brescia mas analisaram mais alguns parâmetros como *authentication runs*, autenticação bem-sucedida (*successful run*), *phase 1 e 2 authentication*. Neste projeto, já tinham como objetivo uma autenticação passiva, bem como ativa, tendo configurado um *captive portal* para informar os utilizadores caso se tivessem ligado e autenticado ao *evil twin*. Também analisaram dispositivos *Apple*, pois este segundo projeto já se tratava de um ataque ativo e não passivo, estando estes dispositivos vulneráveis apenas caso os utilizadores forcem a ligação, devido a más práticas.

Os autores concluíram assim que este método de autenticação para os utilizadores (maioritariamente alunos) não era o mais adequado para a sua segurança e podiam ser vulneráveis a este tipo de ataques. Como analisado anteriormente, no caso dos dispositivos *Apple* mesmo não se conectando automaticamente, quando os utilizadores forçam a sua conexão tornam-se vítimas do ataque. Os autores também notaram que de forma geral existe um certo desinteresse por parte dos utilizadores por terem sido vítimas deste ataque, pois não obtiveram praticamente respostas para as informações que apresentaram no seu *captive portal*(página web que obriga a visualização e interação por parte de um utilizador numa rede pública - normalmente é utilizado como forma de *login* através do preenchimento de um formulário que depois analisa as credenciais do utilizador e fornece acesso à rede ou não. Neste caso os autores criaram este *captive portal* para redirecionar certos pedidos para esta página e assim alertar os utilizadores que se teriam conectado a um *Evil Twin ou Rogue Access Point*), o que ainda piora a situação e consideração por este tipo de ameaças.

Statistic	Medical School		Dept. of Economics		Engineering faculty	
	Value (abs)	Value (%)	Value (abs)	Value (%)	Value (abs)	Value (%)
Total runs	1806		516		1453	
Successful runs	708/1806	39.2%	96/516	18.6%	363/1453	25%
Non-random MACs	752/1806	41.6%	98/516	18.9%	323/1453	22.2%
Anonymous identities	0/1806	0%	0/516	0%	0/1453	0%
Phase 1 - PEAP	708/708	100%	98/98	100%	361/363	99.5%
Phase 1 - TTLS	0/708	0%	0/98	0%	2/363	0.5%
Phase 2 - GTC	345/708	48.7%	13/96	13.5%	54/363	14.9%
Phase 2 - MS-CHAPv2	225/708	31.8%	31/96	32.3%	127/363	35%
Phase 2 - PWD	138/708	19.5%	52/96	54.2%	182/363	50.1%
Apple devices	103/708	14.6%	0/96	0%	4/363	1.1%

Tabela 2.2: Statistics for three departments (Palamà et al., 2023)

QR Codes

Em relação a trabalho relacionado com *QR Codes* no contexto de ameaças à segurança dos utilizadores, destaco o trabalho elaborado por Sharma (2012) que retrata alguns possíveis vetores de ataque para os utilizadores que realizam a leitura dos códigos nos seus dispositivos e também o trabalho realizado por Kharraz et al. (2014), em que os autores analisaram um grande número de *websites* que continham *QR Codes* e verificaram se estes eram maliciosos ou não, se redirecionavam os utilizadores para *websites* intermediários ou se os levavam diretamente para *download* de conteúdo malicioso/*phishing* (profundidade) e também os organizaram por categorias (negócio, *downloads* gratuitos, música, conteúdo adulto e notícias *online*).

QR Codes são símbolos de duas dimensões criados inicialmente para a indústria automóvel nos processos de controlo de produção de peças automóveis, tendo-se tornado num meio de partilha e transmissão de informação (Goel et al., 2017), começando a ser vastamente utilizados em diferentes áreas, desde monitorização comercial a entretenimento, etiquetagem de produtos em lojas e aplicações em

que utilizadores utilizem *smartphones* (Tiwari, 2016), utilizado para codificar um *link* ou informação textual, tornando-os instantaneamente disponíveis, evitando a necessidade de escrita do *URL* por parte do utilizador, até transações monetárias em que são enviados dados sensíveis (Krombholz et al., 2015), tendo-se tornado *standard* de transferência de dados (Pasala e Mukherjee, 2024) pelas seguintes razões:

- Maior capacidade de armazenamento de informação que os códigos de barras.
- Pode ser utilizado sem qualquer custo associado por qualquer utilizador.
- A maior parte dos dispositivos móveis dos utilizadores possuem câmaras fotográficas com capacidade de leitura de códigos *QR* e podem aceder a *URLs* ou aplicações que estejam codificados no mesmo.

(Soon, 2008)

O processo de criação de códigos *QR* pode ser dividido em sete etapas:

1. ANÁLISE DE DADOS: acontece previamente à codificação das *strings* para uma *string* de bits. Como este método é diferente para cada tipo de texto (numérico, alfanumérico, byte ou kanji), é feita a análise para determinar o melhor método de codificação.
2. CODIFICAÇÃO DE DADOS: É feita a codificação dos dados que resulta numa *string* de *bits* dividida em códigos de 8 *bits* de comprimento. O método de conversão é identificado pelo *Mode Indicator*, que tem 4 *bits* de comprimento, que está inserido no início dos dados codificados.
3. CORREÇÃO DE ERROS: Nesta etapa é feita a correção de erros e a sua codificação através do método Reed-Solomon, onde é utilizada a *string* de texto codificado e acrescenta os erros gerados. Os leitores de códigos *QR* realizam a leitura tanto do texto codificado como dos erros gerados que foram codificados.
4. CRIAÇÃO DA MENSAGEM FINAL: O texto codificado e os erros gerados codificados são organizados para a mensagem final na sua ordem correta. Em textos de dimensão grande são gerados por blocos.
5. COLOCAÇÃO DOS MÓDULOS NA MATRIZ: Após a sua organização, os blocos de *bits* são colocados na matriz do código *QR*.
6. *DATA MASKING*: Feito um *data masking* de oito padrões para facilitar a leitura do código. Os padrões iniciais podem não ser facilmente lidos por *scanners* de códigos *QR*.

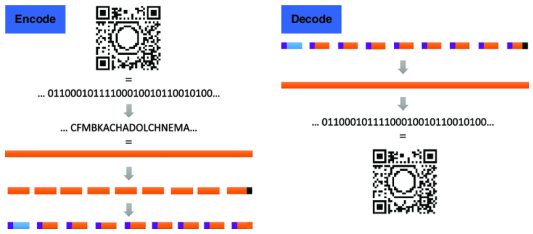


Figura 2.7: Encodificação e descodificação de código *QR* (Martens et al., 2018)

7. FORMATAÇÃO E INFORMAÇÃO DE VERSÃO: Neste último passo é feita a formatação e se necessário é dada a versão do código *QR*, adicionando pixéis em áreas específicas do mesmo.

(Tiwari, 2016)

Para ser feita a descodificação de um código *QR*, é feito quase o processo inverso, também dividido em sete etapas:

1. RECONHECIMENTO DE MÓDULOS: reconhecimentos de módulos claros e escuros, criando um *array* de *bits*.
2. EXTRAÇÃO DE INFORMAÇÃO DE FORMATAÇÃO: Descodificação da formatação utilizada e é também extraído o padrão de *masking*. Também é aplicada correção de erros na informação de formatação.
3. INFORMAÇÃO DE VERSÃO: Se aplicável, é feita a descodificação da versão do código, que está contida na área de informação da versão.
4. EXTRAÇÃO DA *MASKING*: Utilizada a operação *XOR* na região codificada de bits com o padrão de *mask*, que já foi extraído da informação de formatação
5. EXTRAÇÃO DE DADOS E CÓDIGOS DE CORREÇÃO DE ERROS: Leitura dos símbulos de caracteres, extraindo os dados e códigos de correção de erros da mensagem.
6. DETEÇÃO DE ERROS E CORREÇÃO: Utilização dos códigos de correção de erros para identificar erros e corrigi-los.
7. DESCODIFICAR A MENSAGEM: Divisão dos blocos em segmentos, de acordo com os indicadores de modo e dos indicadores de contagem de caracteres. São descodificados os dados de caracteres de acordo com o modo de codificação e a mensagem é descodificada em texto.

(Tiwari, 2016)

Apesar dos seus benefícios, existem bastantes riscos associados à utilização dos códigos *QR*. Execução de código malicioso, roubo de informação bem como violação de privacidade e roubo de identidade são riscos típicos a que um utilizador pode estar sujeito em segundo plano enquanto está apenas a ler o código QR em primeiro plano (Bani-Hani et al., 2014). Os riscos associados por parte de códigos *QR* podem ser divididos em duas categorias: processos automáticos e interação humana. (Sharma, 2012)

No que toca a processos automáticos, envolvem todos os ataques que sejam do tipo de injeção de código no browser/dispositivo da vítima (*shell*) ao fazer um sistema de forma diferente à suposta de quem o criou (Krombholz et al., 2015) tais como *URLs* com comandos/código incorporados - através de *cookies*, formulários ou *HTTP headers* (Averin e Zyulyarkina, 2020), *SQL Injection* - caso o *URL* da vítima não tenha nenhum tipo de controlo contra estes ataques, permitindo ao atacante elevar privilégios, alterar ou eliminar dados da base de dados (Averin e Zyulyarkina, 2020), ataques *XSS* - através do *URL* de um *website* fictício ou em alguma vulnerabilidade encontrada que executa *JavaScript* no dispositivo da vítima (Averin e Zyulyarkina, 2020), e *Payloads* maliciosas no *URL* que fazem *download* quando abertos e os executam automaticamente - também designados por *drive-by downloads* que permitem ao atacante escalar o seu ataque através de ataques como execução de código remoto e roubo de informação pessoal (Kharraz et al., 2014). Existem também casos em que um atacante altera o *QR Code* legítimo (Pasala e Mukherjee, 2024) para um visualmente semelhante, mas que direciona o utilizador para um *website* malicioso (Yong et al., 2019). Modificando os módulos do código, a aplicação que faz o *scanning* interpreta os dados de forma diferente (Krombholz et al., 2015).

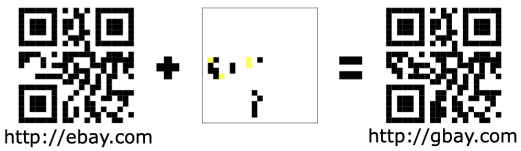


Figura 2.8: Manipulação de *QR Code* (Yong et al., 2019)

Nos riscos de ataque através da interação humana são os ataques que exploram a fraqueza da interação humana ao realizar o scan do *QR Code*, uma vez que um humano não tem a capacidade de descodificar um código *QR* visualmente, torna-o o fator mais vulnerável (Krombholz et al., 2015). Alguns destes ataques são: *Phishing*, onde existe um clone ou até mesmo um serviço/website fictício que requisite dados do utilizador e este insere-os - estes ataques baseiam-se tanto no

engano a nível técnico, bem como técnicas de *social engineering* (Kharraz et al., 2014). *Social engineering*, com técnicas de persuasão para manipular as vítimas ao fazer os utilizadores abrir o *URL* através do código *QR* e a partir daí realizar a propagação de *malware*, ou injetar código no dispositivo do mesmo, como também obter informações pessoais dos mesmos (Sharma, 2012). O sucesso deste tipo de ataques é devido ao facto de não ser possível ser feita a leitura por parte do olho humano, apenas por dispositivos que consigam realizar essa leitura (Rafsanjani et al., 2023) aliado a uma fácil e rápida implementação por parte dos atacantes, tornando também este ataque bastante eficiente (Kharraz et al., 2014). Também se sabe que a maioria dos códigos *QR* maliciosos detetados realiza um redirecionamento do utilizador para estes *websites* de *phishing* e *exploits* comparativamente a *websites* que realizam *downloads* de *malware* de forma direta (Yong et al., 2019). No trabalho realizado por (Vidas et al., 2013) onde os autores espalham diversos posters e realizam depois inquéritos às pessoas que efetuaram o *scan* ao código *QR*. Estes concluíram que mais de 75% dos utilizadores realizaram o *scan* por curiosidade(64%) ou diversão(14%). 54% dos utilizadores leram o conteúdo do *URL* antes de visitar o *website*, algo que os autores referem como mais seguro do que os restantes 36% que não leram mas visitaram na mesma um domínio que não era reconhecido/conhecido - algo que pode ser aproveitado e explorado por potenciais atacantes, uma vez que as vítimas não se abstêm de abrir o *website* por não conhecerem o domínio.

QR Codes e seus ataques

No trabalho realizado por Kharraz et al. (2014) os autores realizaram uma análise a 14 milhões de páginas *web* através de *web crawlers* com o intuito de averiguar a quantidade de *websites* maliciosos que existiriam através do scan de *QR Codes*. Já nesta altura os autores concluíram que este mecanismo estava a ser utilizado para distribuir *malware* ou realizar ataques do tipo *phishing* a potenciais vítimas, mas também concluíram que o número era reduzido, o que significava um risco reduzido à exposição deste tipo de ataque, algo que potenciais utilizadores maliciosos poderiam usar como vantagem, uma vez que as potenciais vítimas poderiam confiar na página *web* e que o código *QR* não os vá direcionar para uma página maliciosa. Este tipo de ataques é bastante simples e eficiente, baseiam-se em técnicas de embuste técnico e técnicas de *social engineering* em que os atacantes abusam da confiança imposta no *website*, e em caso de criação de clone malicioso do *website*, na confiança imposta do utilizador a acreditar que o *website* é legítimo. Os atacantes exploram esta abordagem dos *QR Codes* pelo facto de serem facilmente gerados, o *URL* que está presente no *QR Code* não precisa de ser inserido manualmente no *browser* das

potenciais vítimas e assim podem não reparar no *URL* que está a ser aberto (Vidas et al., 2013). Em grande parte dos casos, os *smartphones* também encurtam o *URL* e não permitem aos utilizadores visualizarem o mesmo completo, outro fator que poderia ser utilizado a favor dos atacantes, bem como utilizar *URL shorteners* para também esconder o *URL* completo das vítimas.

Os autores concluíram que 38% dos códigos *QR* analisados redirecionavam os utilizadores para páginas diferentes, e que alguns redirecionavam utilizadores para páginas que realizavam *download* direto de artefactos executáveis, tanto executáveis do tipo *Windows Exe* como ficheiros *Android APK*, em que 67,1% dos casos se tratavam de versões gratuitas de aplicações pagas. Os autores também analisaram como se distribuíam estes ataques de *QR Codes* e verificaram que em 53% dos códigos *QR* maliciosos a categoria do *website* era diferente do *website* do qual eram extraídos. Concluíram que os códigos *QR* maliciosos podem ser usados para distribuição de *malware*, tanto para *Windows* como para dispositivos móveis, via *downloads* diretos ou levando as vítimas a websites de *phishing*, sítios intermediários - *websites* vulneráveis que são explorados por atacantes através de injeção de *scripts* para as suas vítimas ou até mesmo *exploit sites*, que são *websites* que contêm ferramentas para detetar e comprometer aplicações vulneráveis nos dispositivos das vítimas. Neste projeto, os autores verificaram que os atacantes utilizavam os *QR Codes* para facilitar o processo de interação com as vítimas e abusavam da confiança que os utilizadores depositam em marcas e *websites* conhecidos. No caso dos ataques do tipo *phishing*, os códigos *QR* redirecionavam os utilizadores para *websites* clones maliciosos de sítios conhecidos, para de seguida obter dados de autenticação ou pagamento. Nos casos em que existia distribuição de *malware* por *download* direto, os autores concluíram que o tempo de vida do mesmo era curto e que o *URL* correspondente deixava de funcionar num curto espaço de tempo, sendo raramente bem sucedido este ataque - apenas 5 *QR Codes* foram bem sucedidos nesta tarefa.

Possíveis formas de mitigar estes ataques

Evil Twin

Sugestões para possíveis mitigações destes ataques passam por uma maior literacia digital e alerta dos utilizadores, não só a nível de prevenção em dispositivos como *laptops*, onde já existe alguma preocupação por parte dos utilizadores que já têm o cuidado de utilizar proteção contra vírus, atualizar os dispositivos e utilização de

firewalls (Tick, 2018) mas no que toca a *smartphones* já não acontece o mesmo (Tick, 2018). Uma vez que são bastante utilizados, é necessário um reforço e aumento de ações de sensibilização aos utilizadores acerca deste tema como também de como funcionam, no caso desta instituição e muitas de ensino superior, ações de sensibilização acerca de como *WPA-Enterprise* funciona, como por exemplo acerca das trocas de certificados e protocolos de autenticação (Palamà et al., 2023). No caso de um utilizador se conectar a um *Evil Twin* e não reparar, seja numa rede pública desconhecida ou a uma rede privada que esteja habituado/protegido, uma boa solução é a utilização de *Virtual Private Networks (VPNs)* pois cria uma camada de encriptação adicional (uma vez que a expectada do *AP* já não existe) que um atacante teria de superar - o atacante apenas consegue observar tráfego encriptado por parte da *VPN* (Almjamai, 2022).

Uma das soluções apresentadas por Bauer et al. (2008) para a deteção de *Evil Twins* passa pela avaliação de contexto. Um utilizador numa determinada localização tem informação dos diversos *APs* ativos. Ao mudar de localização e visualizar um determinado *AP* com o mesmo *SSID*, poderá não estar suscetível a ligar-se. Os autores exemplificam com um *CoffeeAP* numa determinada localização, com o utilizador conectado ao mesmo, estando rodeado de diversos e distintos *APs*. Ao mudar a sua localização vai ser exposto a mais *APs* a emitir *probes* para conexão, inclusivamente o utilizador com *probes* à procura de um dos *APs* já previamente conectados, incluindo o *CoffeeAP* - Um atacante escuta estes *probe requests* e cria um *Evil Twin* com o SSID de *CoffeeAP*. Ao utilizar esta informação contextual, o utilizador sabe que não se encontra na localização onde o *AP* legítimo estaria e decide não se conectar ao *Evil Twin*.

Outras soluções num ponto de vista proativo, são a implementação de formas de controlo destes ataques como por exemplo, desenvolver um mecanismo de controlo nas *probe response frames* que implementa uma contagem do número de vezes que um *AP* estabeleceu uma ligação à lista desta resposta e do lado do *AP* guarda o número de vezes que um dispositivo se conectou ao mesmo. Quando um cliente se tenta conectar ao *AP*, este procura na sua tabela se já existiram conexões e o seu número de conexões, através do seu endereço *MAC*. Do lado do sistema operativo, este vai procurar os *SSID*s dos *AP* e procura a sua contagem para comparação, autenticação e associação. Após o envio de associação, o *AP* incrementa o seu valor de contagem e o sistema operativo ao receber o *response frame* de associação, aumenta o valor de contagem para aquele *AP*. Tendo valores diferentes de contagem gera um alerta de *Evil Twin* detetado (Kumar e Paul, 2016).

Outra forma possível para mitigar seria utilizar algo como *EvilScout*, que implementa *Wi-fi SDN* para deteção, pois possibilita um controlo centralizado de todos os *AP*s, bem como facilita o processo de monitorização da rede - através de *software*. Este está configurado para realizar a monitorização da rede e utiliza a distribuição de *IP-Prefix* para detetar o *Evil Twin*. Analiza os *IP*s dos pacotes recebidos no gestor de *IP-Prefix* no controlador (uma vez que o *evil twin* pode estar a comunicar através de um *AP* legítimo) e se o endereço *IP* do cliente não for consistente com o gestor no controlador, é detetado o *Evil Twin* (Shrivastava et al., 2020).

O segundo método apresentado por Bauer et al. (2008) trata-se de um método de autenticação com a *framework EAP* , denominado *EAP-SWAT*. Este método é semelhante ao *EAP-TTLS* com autenticação unilateral por parte do servidor, mas com a utilização do princípio *trust-on-first-use* com o certificado do *AP* - sendo vulnerável na primeira autenticação, mas que os autores consideram que não seja uma limitação relativamente grave uma vez que o utilizador teria de selecionar o *AP* a que se quer conectar. Também difere pelo facto do cliente decidir se aceita o certificado do servidor, enquanto que no protocolo *EAP-TTLS* é exigido que a troca de credenciais seja efetuada fora de banda antes que a autenticação ocorra. Os *AP*s utilizariam certificados de sessão *TLS*, como os utilizados para ligar chaves públicas com uma determinada identidade (chave pública *TLS* ligada a um servidor com recurso a um *URI*, como um *DNS*). Uma vez que os *AP* não possuem identidades tão fortes como *URI*, podem utilizar certificados do tipo *self-signed* - utilizando o seu *SSID* ou endereço *MAC* ou até mesmo uma combinação de ambos. A sessão *TLS* garante proteção contra ataques do tipo *replay* e privacidade no envio de dados, uma vez que se a chave for comprometida apenas aquela sessão se encontra comprometida. Os autores aconselham a utilização deste método, bem como em conjunto com o método de análise de contexto referido anteriormente para uma maior segurança.

Estes métodos aliados a uma monitorização constante, não só de *Evil Twins* como também de ataques *deauthenticantion* através de soluções como *Intrusion Detection Systems (IDS)* que analizam o tráfego em tempo real, identificam atividade suspeita na rede e consequentemente geram alertas dos mesmos é possível aumentar a segurança em relação a estes ataques (Mohanan, 2024). Algumas organizações criaram *Wireless Intrusion Detection Systems(WIDS)* que permitem analisar e monitorizar dispositivos *wireless*, detetar *Evil Twins* e *Rogue Access Points* e alertar os administradores de alguma anomalia detetada na rede (Almjamai, 2022).

QR Codes

Para uma mitigição deste tipo de ataques, de injeção de código bem como *phishing* neste cenários é necessário também um reforço de literacia digital dos utilizadores uma vez que a maioria dos utilizadores não sabe navegar de forma segura pela *internet*, não sabendo distinguir *websites* malignos e seguros (Rafsanjani et al., 2023). Que nem todos os *links* que acedam através da leitura de *QR Codes* são benignos. Podem automaticamente executar *scripts* e desencadear uma série de ciberameaças. Tendo uma maior formação poderão assim também detetar *links* que pareçam suspeitos e não prosseguir com o redirecionamento do *QR Code*, caso o *scanner* do *QR Code* permita visualizar o *URL* do *website*.

Outra forma de mitigar este tipo de ataques é através da utilização de uma *framework* como a *QsecR*, que divide a leitura e *scanning* de *QR codes* em 3 fases (Rafsanjani et al., 2023):

1. ANÁLISE DO *URL*: realiza a leitura do *URL* que está contido no código, analisa se este está encurtado ou se tem alguma extensão de ficheiros.
2. EXTRAÇÃO DE CARACTERÍSTICAS E CLASSIFICAÇÃO: utiliza *blacklists* recolhidas de outros modelos de deteção de *websites* maliciosos e também baseados em *machine-learning*. Utiliza 62 classes para a classificação das características com base nessas *blacklists*, lexicais, *host-based* e *content-based*.
3. DETEÇÃO DE *URL* MALICIOSO: Aplicação de funções que auxiliam na avaliação de se tratar de um *URL* malicioso ou não.

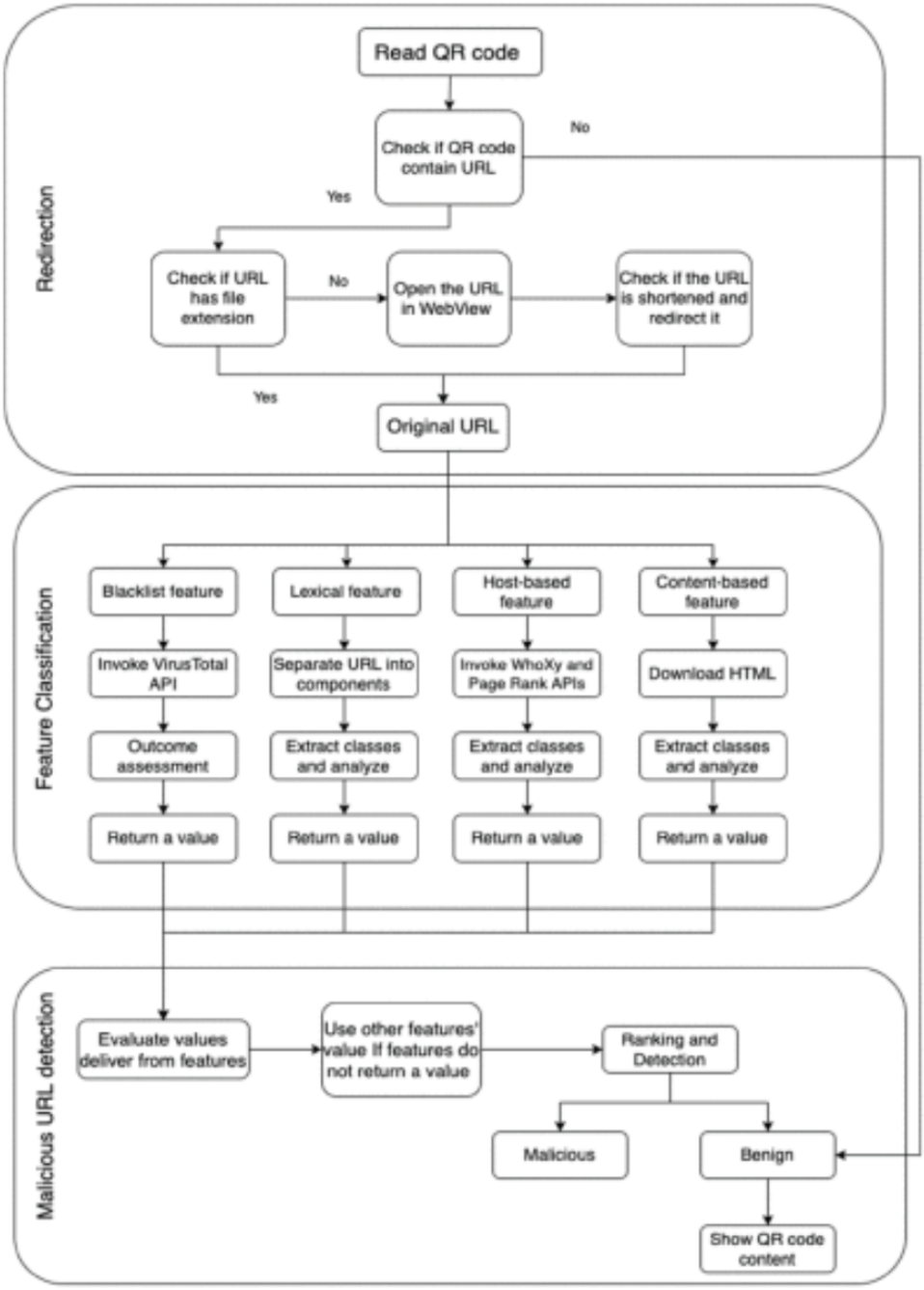


Figura 2.9: *Framework QsecR* (Rafsanjani et al., 2023)

A mitigação dos ataques com códigos *QR* também pode passar por proteger a integridade dos mesmos (códigos que são alterados por atacantes) através da inserção de assinaturas digitais e/ou métodos de encriptação (Krombholz et al., 2015; Yong et al., 2019) e utilização de *software* de verificação das mesmas - falhando a verificação, fica bloqueado e o utilizador é apresentado com uma opção de não prosseguir para o *website*. Uma das abordagens analisadas por Yong et al. (2019)

utiliza este mecanismo de assinaturas combinado com um servidor de verificação e prevenção de atividades maliciosas que está equipado com melhores antivírus, *firewalls* e *Intrusion Detection Systems* do que os que estão instalados no dispositivo das potenciais vítimas e também utiliza uma *sandbox* para prevenção da violação de privacidade, acesso a informação e acesso remoto aos dispositivos das vítimas através da aplicação de *scanning*. Também é bastante comum incorporar a verificação automática do *URL* diretamente nos *browsers* dos utilizadores, como o *Google Safe Browsing* - utiliza uma *framework* para detetar ataques do tipo *phishing* e de *malware* bem como fornece uma *API* disponível publicamente e a *PhishTank API* (Krombholz et al., 2015; Yong et al., 2019). Kharraz et al. (2014) acreditam que a mesma abordagem pode ser utilizada para detetar códigos *QR* maliciosos nas aplicações que realizam *scanning* e nos *browsers* móveis, em que forneceriam um *feedback* imediato aos utilizadores se o *URL* que está contido no código tem carácter malicioso e permitindo ao utilizador não avançar com a pesquisa desse *URL*.

Visual QR Codes - tornar os códigos mais complexos uma vez que quanto mais complexos forem, mais difícil e caro se torna de os modificar. Este processo passa por utilizar um conjunto de cores e texturas mais complexos na criação do código (Krombholz et al., 2015).

Deteção de *URL* malicioso, através da análise de características como: lexical, popularidade do *link*, conteúdo da página *web*, *DNS*, fluxo de *DNS*. Também é aconselhada a utilização de *black-/whitelists* de *URL*s frequentemente atualizada (Krombholz et al., 2015).

Pré-análise do conteúdo da página *web* bem como a resolução de algum tipo de *URL* encurtado, demonstração do *URL* ao utilizador. No caso de subdomínios, Krombholz et al. (2015) recomendam o destaque do segundo nível do domínio. Os autores também concluíram que nenhuma das aplicações analisadas fornecia informação acerca de métodos *callback* nem de tráfego oculto e, portanto, recomendam que o conteúdo descodificado seja mantido localmente e que não seja transmitido a terceiros.

3

DESENVOLVIMENTO

Primeiro Projeto

Foi realizada uma pesquisa inicial acerca do protocolo Wi-Fi 802.11X (redes a/b/g/n), que mecanismos de segurança e autenticação podem ser utilizados, como WPA2 e WPA *Enterprise*, RADIUS, MAC *authentication* e foi feito também o levantamento das eventuais ferramentas necessárias para a realização do projeto/trabalho.

Para a realização deste projeto foi utilizado um computador portátil **Macbook Pro M1**, com um processador M1 Pro de 10 *cores*, 16GB de RAM. Máquina virtual *VMware Fusion Pro 13.6* com o sistema operativo *Kali Linux* - versão 2024.2, adaptadores Wi-Fi USB **ALFA AWUS036ACS** e **TP-Link Archer T3U Plus AC1300** com o standard 802.11ac, ou seja, permitem trabalhar com larguras de banda de 2.4 e 5 GHz.

Foram instaladas ferramentas para auxílio das tarefas, como o EAPHammer, que permite criar e configurar *Access Points* do tipo *WPA-Enterprise* de forma mais rápida e também o *Wifi-Crack*, para realizar ataques do tipo *Deauth* na rede.

EAPHammer

EAPHammer é uma ferramenta utilizada para realizar ataques a redes Wi-Fi, especialmente para o tipo *WPA-Enterprise*, facilitando no processo de criação de *Evil Twins*, contendo uma interface bastante simples de utilizar. Através de alguns comandos, é possível configurar um *Evil Twin* pronto a utilizar. O EAPHammer permite criar *Evil Twins*, realizar a captura de credenciais, intercetar falhas no protocolo de autenticação EAP e assim obter acesso a credenciais dos utilizadores e automatizar processos de ataques a redes (*evil twins*, *captive portals*, entre outros). Esta ferramenta também permite criar, importar e gerir certificados x.509 necessários para a realização de ataques *WPA2-Enterprise*, como o *Evil Twin*. O EAPHammer também possui um método em que tenta realizar o *downgrade* do mecanismo de autenticação com o objetivo de facilitar a obtenção de credenciais. Foi criado um *script* que realiza a configuração do *access point* automaticamente e também

para omitir as passwords introduzidas pelas potenciais vítimas. Estas credenciais adquiridas são guardadas num ficheiro de output log. Neste caso, a utilização deste tipo de ferramentas serve para criar uma rede semelhante à rede legítima da instituição e um atacante pode fornecer acesso à rede autêntica, realizar o encaminhamento de tráfego para a rede autêntica e realizar ataques de MITM. Neste projeto, não existiam intenções de prosseguir com esta última vertente, apenas testar se os alunos e docentes se ligavam a este *access point* e introduziam as suas credenciais.

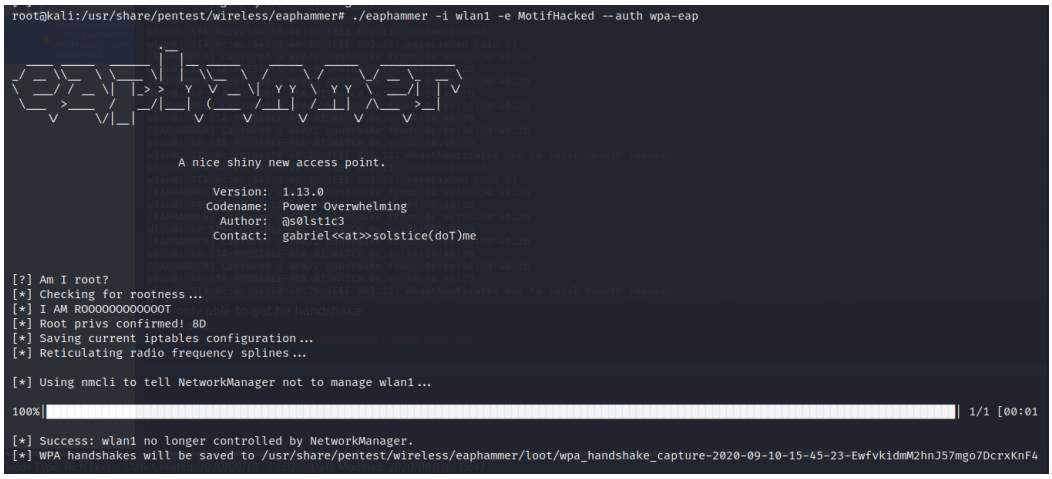


Figura 3.1: EAPHammer - Inicialização

Wifi-Crack

Wifi-Crack é uma ferramenta em *Python* para realizar ataques a redes Wi-Fi com o objetivo de obter *handshakes*, ataques do tipo PMKID, ataques de *Denial of Service* ou criar *Evil Twins*. Foi utilizada para realizar ataques do tipo *Deauth* pois centraliza todos os processos necessários para executar o ataque de forma mais rápida (Deteção do *Access Point*, canal utilizado, tempo de ataque que o utilizador deseja e ataque de desautenticação).

Após feitas as configurações necessárias, foram iniciados os ataques nas zonas mais frequentadas do instituto - bar, pisos de salas de aulas e biblioteca com o objetivo de captar o maior número de credenciais possível, tendo um adaptador Wi-Fi a realizar ataques de *deauth* e outro como *Evil Twin*.

Para executar o EAPHammer, ocultar as *passwords* inseridas pelo utilizador tanto no terminal como no ficheiro de output, criei um *script* em *bash*, de forma a tornar este processo automático e não ser possível em momento algum ter acesso às passwords inseridas pelos estudantes ou docentes, uma vez que estas *passwords* não

só concedem acesso à rede Wi-Fi da instituição, como também fornecem acesso à área de alunos e docentes - para consulta e divulgação de material de estudo, notas, informações de pagamentos de propinas, entre outros.

Listagem 1: *Script* para executar EAPHammer e ocultar *passwords* inseridas

```
#!/bin/bash
eaphammer -i wlan1 -e "ISPGAYA" -c 6 --auth wpa-eap --creds | grep -v "password"

python3 cleanFile.py

cp /home/kali/eaphammer/logs/hostapd-eaphammer.log outputfile.log
```

Listagem 2: *Script* em *Python* para remover as passwords do ficheiro de log do EAPHammer

```
import os
path="/home/kali/eaphammer/logs/"
with open(os.path.join(path,"hostapd-hammer.log"),"r") as f:
        lines=f.readlines()
with open(os.path.join(path, "hostapd-hammer.log"), "w") as f:
        for line in lines:
                if "password" not in line:
                        f.write(line)
```

Listagem 3: *Script* para executar o Wifi-Crack

```
#!/bin/bash
python3 wifiCrack.py -i wlan0 -m DAuth
```

Durante alguns dias a rede do ISPGAYA esteve em baixo (em manutenção) e utilizaram *Access Points* sem autenticação de acesso livre, temporariamente. Realizei então a alteração do *Evil Twin* para tentar captar os utilizadores com um SSID de rede semelhante ao utilizado.

Listagem 4: *Script* de *Evil Twin* para replicar a rede temporária ISPGAYA

```
#!/bin/bash
eaphammer -i wlan1 -e "ISPGAYA OPEN 1" -c 6 --auth wpa-eap --creds | grep -v
↪  "password"

python3 cleanFile.py

cp /home/kali/eaphammer/logs/hostapd-eaphammer.log outputfile.log
```

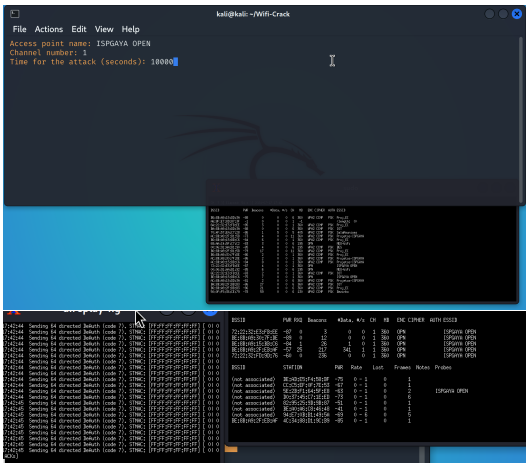


Figura 3.2: Wifi-Crack em utilização para descoberta de APs e ataques de *deauth*

Também é importante notar que os *Evil Twins* não tinham os SSIDs completamente idênticos aos *AP*s genuínos, conforme ordens superiores, para evitar confundir os alunos e docentes, e apenas para conduzi-los ao erro de se ligarem aos *Evil Twins*, o que poderá ter influenciado os resultados de utilizadores afetados, pois os mesmos poderiam suspeitar do SSID. Também não consegui realizar a *bridge* entre a placa de rede utilizada para criar o *Evil Twin* e o *AP* genuíno, utilizando o *routing* entre os mesmos e tendo o servidor de *dnsmasq* configurado para tal, o que poderia ter tornado ainda mais interessante esta atividade ou projeto, numa perspetiva de ataque do tipo MITM, demonstrando que poderia ser feito o *routing* do tráfego dos utilizadores ou até alterá-lo, escalando-o. Tentei utilizar outras ferramentas como as utilizadas pelos autores do trabalho Palamà et al. (2023) - *hostapd*, *dnsmasq* como já referi, *freeradius*, mas sem sucesso. Dentro destas condições foi possível observar resultados positivos deste projeto, onde foram capturadas credenciais inseridas por alunos bem como alguns acessos que não entravam para o ficheiro de *logs* do *EAPHammer*, apenas eram registadas entradas no terminal em que o *EAPHammer* estava a ser executado - ligavam-se ao *AP* mas a conexão não era estabelecida como mostra a figura 3.4.

```
GTC: Tue Feb 27 17:29:41 2024
        username:       ispg2021101726
```

Figura 3.3: Exemplo de credencial - ficheiro de *log* do EAPHammer

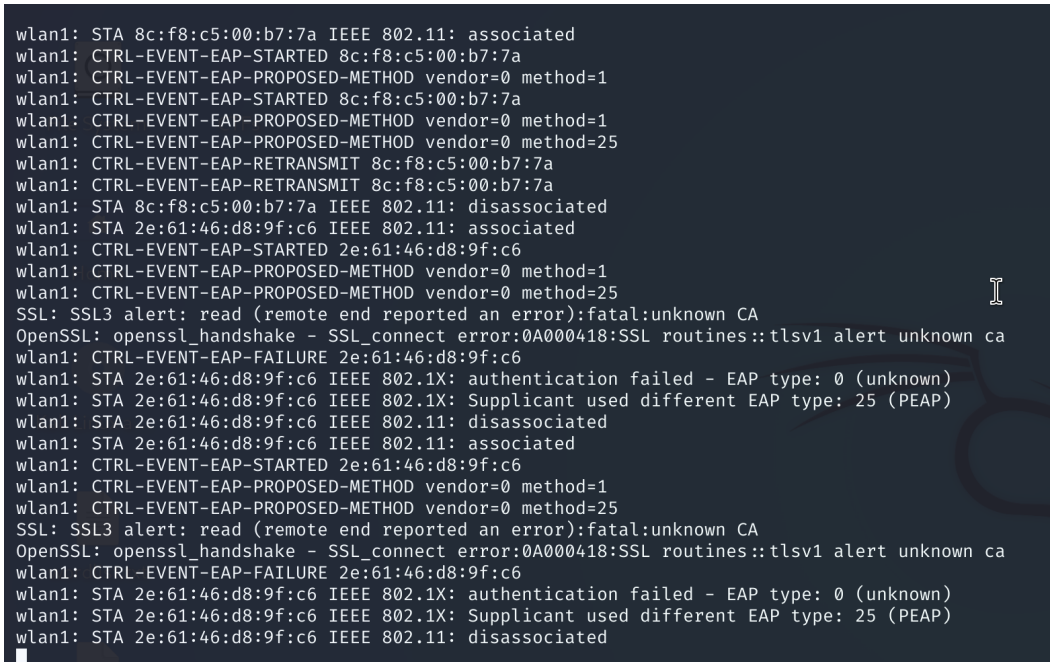


Figura 3.4: Exemplo de ligação feita com o *Evil Twin* mas não gera credênciais

Tendo em conta estas informações, foram registadas pelo menos 23 ocorrências ou credenciais inseridas pelos alunos durante este projeto - números um pouco escassos, tendo em conta o número de alunos que existem no ISPGAYA, mas pode ser visto como um resultado positivo devido ao relativo sucesso na obtenção de credenciais dos mesmos. Como é possível observar na tabela 3.1 os métodos de autenticação em que foi possível captar as credenciais inseridas pelos utilizadores. No caso da figura 3.4, a autenticação não é bem-sucedida (muito possivelmente porque os utilizadores não aceitaram o certificado enviado pelo *Evil Twin*, que era do tipo *self-signed*, ou porque os dispositivos não aceitavam outro método de autenticação senão o *EAP-TLS* e *EAP-TTLS*). Algo que também é relatado no trabalho apresentado por Chatzoglou et al. (2021), em que os ataques não foram bem sucedidos para dispositivos que utilizavam métodos de autenticação baseados em *TLS*, como os referidos anteriormente. Deve-se ao facto de como o AP falso atuava em simultâneo como servidor *RADIUS*, não possuía um certificado válido - certificado gerado por uma entidade em que o dispositivo pode confiar, apenas auto-gerados. Como foi referido anteriormente, o facto dos SSIDs não serem iguais poderá ter afetado os utilizadores no momento de ligação ao *Evil Twin*, uma vez que poderiam suspeitar da legitimidade do mesmo e não proceder à ligação ao mesmo.

Tal como os autores referem no trabalho Palamà et al. (2023), os alunos forçam a conexão dos seus dispositivos ao *AP* e aceitam o certificado enviado pelo *Evil Twin* sem qualquer validação do mesmo, efetivamente tornando ineficientes alguns

Tabela 3.1: Métodos de autenticação pelos utilizadores

Tipo de Autenticação	Percentagem (%)	Valor Absoluto
EAP-GTC	78,3%	18
MS-CHAPv2	21,7%	5

mecanismos de segurança implementados pelos dispositivos iOS (por exemplo) que evitam essa conexão automática. Também em concordância com o trabalho destes mesmos autores, o método de autenticação menos utilizado foi o *MS-CHAP v2* que era o recomendado/diretrizes dos administradores de rede desses institutos de ensino. O método *EAP-GTC* foi o mais utilizado apenas num dos institutos (medicina). Este mecanismo de autenticação é selecionado também devido ao facto do EAPHammer possuir um ataque (*EAP-GTC Downgrade attack*) em que realiza o *downgrade* do método de autenticação *EAP* no momento de negociação do mesmo. Com este método as passwords são então enviadas em *plaintext* (Vink, 2020).

Segundo Projeto

A segunda atividade ou projeto visava alertar os utilizadores para os possíveis perigos que existem através da utilização de *QR Codes* - sob um ponto de vista voltado para *social engineering*. Um atacante poderá ter uma página duplicada de algum serviço ou até mesmo um produto fictício e o utilizador, ao fazer *scan* do *QR Code* para esse serviço, sofrer consequências como execução de código, ataques de XSS, execução de *payloads* e propagação de *malware*. Podem também ser vítimas de *phishing*, com o possível roubo de informação pessoal do utilizador.

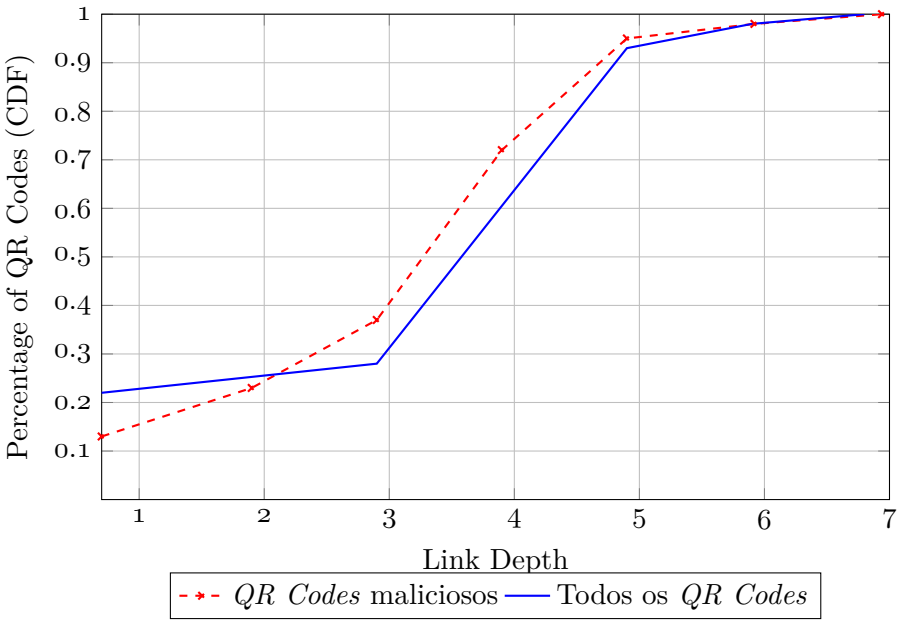
Criei então uma "ideia de negócio"que se tratava de um espaço em realidade virtual de interação de utilizadores - "SPACZ", para assim captar a atenção dos utilizadores, tendo também criado um póster com o *QR Code* e espalhei-o pelos lugares mais utilizados do ISPGAYA, tendo criado também um *URL shortener* para os utilizadores não conseguirem ver o *URL* completo do website e não suspeitarem até abrirem a página e, nesse momento, já teriam executado qualquer *script* que esta contivesse. O objetivo era ter este produto/serviço de fachada para quando os utilizadores abrissem o link que estava associado ao *QR Code*, acionava um *script*. Neste caso, teria apenas o link do Matomo e/ou Google Analytics para questões de métricas, conseguindo assim saber o número de utilizadores que teria aberto o website através da leitura do *QR Code*.

Figura 3.5: Póster com o QR Code utilizado

Figura 3.6: Posters afixados

Foram conseguidas no total 24 visitas ao *website*. O que pode ser visto como positivo, apesar de serem relativamente poucas em relação à proporção do ISPGAYA, provando assim que seria possível escalar os ataques após esta primeira interação com os utilizadores/vítimas, para algumas das várias áreas exploradas anteriormente como possíveis ameaças do *scan* de *QR Codes*. Este ataque poderia ser muito bem

sucedido, uma vez que as vítimas não eram redirecionadas para outro *website*, permaneciam nesta página e teriam código a ser executado em *background*. Também poderiam não suspeitar pois, como indicam os estudos de Kharraz et al. (2014), os *links* maliciosos habitualmente têm uma profundidade de 4, onde quase todos podem ser considerados maliciosos, como podemos observar no gráfico de seguida.



Também como é referido no estudo de Vidas et al. (2013) na época que publicaram este trabalho, já existiam *exploits* e vulnerabilidades que visavam o navegador *WebKit* ou manipulador de conteúdos como ficheiros *PDF* em todos os dispositivos *iOS* e *Android* que foram observados nesse estudo. Para ser possível concretizar este tipo de ataques, o atacante apenas necessita que o utilizador visite essa página com conteúdo malicioso, sendo mais facilmente conseguido com códigos *QR*. Um ataque bem sucedido fará com que o atacante tenha acesso remoto aos mesmos recursos que o navegador. Em alguns casos, é suficiente para realizar a leitura de *cookies* ou obter *passwords* do *website*. Noutros casos o atacante necessita de escalar privilégios como realizar o *rooting* ou *jailbreaking* do dispositivo, que também dependem da versão do sistema operativo para o seu sucesso.

Desenvolvimentos futuros

Para possíveis desenvolvimentos futuros, penso que seria interessante conseguir realizar o *bridge* entre o *Evil Twin* e o *Access Point* legítimo ou com um diferente provedor de serviço, que permita a ligação das vítimas, tornando o ataque mais

discreto e permitindo a visualização e manipulação de tráfego por parte do atacante (ataque *Man-in-the-middle*) e, como é realizado no trabalho Palamà et al. (2023), utilizar essas configurações (*hostapd*, *freeRADIUS* e *dnsmasq*) para a criação do *AP* mas também para identificação de dispositivos para posterior análise e comparação. Também seria de interesse testar o *Evil Twin* utilizando um SSID correspondente ao autêntico para realizar a comparação entre a utilização de um semelhante e um idêntico, qual dos cenários seria mais apelativo aos utilizadores e potenciais vítimas.

Em relação ao projeto dos códigos QR, penso que seria interessante explorar a parte de execução de *scripts* nos dispositivos dos utilizadores, uma vez que não foi possível utilizar *scripts* efetivamente nocivos para os utilizadores. Também seria de interesse ter uma secção onde os utilizadores introduzissem dados numa perspetiva de *phishing*, como a introdução de dados pessoais num formulário em forma de *pop-up*, por exemplo, para posterior análise.

4

CONCLUSÕES

Durante este período de estágio no ISPGaya tive a oportunidade de realizar estes dois projetos que acredito que tenham ajudado no meu desenvolvimento profissional.

Este estágio proporcionou um ambiente propício para desenvolver algumas competências técnicas, ainda que com algumas limitações, com a utilização do *Kali Linux* e algumas ferramentas que estão inseridas no mesmo, bem como criar e utilizar *scripts* para auxiliar o meu trabalho. Também consegui praticar algumas técnicas de *social engineering* e observar o seu resultado em primeiro plano.

Em relação ao primeiro projeto, encontrei algumas dificuldades já referidas anteriormente como não conseguir realizar o *bridge* entre a placa de rede e o *access point*, o que teria tornado este projeto mais interessante, uma vez que seria possível analisar o tráfego dos utilizadores e tornaria mais discreto o *Evil Twin* e o facto de os SSIDs não serem iguais, o que poderia tornar para os utilizadores suspeito e não quererem fazer a conexão com o mesmo.

Outro desafio foi encontrar uma solução para não se poder ver as passwords inseridas pelos utilizadores tanto em tempo real (terminal a executar o EAPHammer), bem como dos ficheiros de *log*, tendo de criar uma *script bash e Python* para as remover enquanto o EAPHammer estivesse em execução e quando acabasse, remover do ficheiro de log.

No que toca ao segundo projeto, com os *QR codes*, seria interessante ter mesmo um *script* que realizasse algo em segundo plano mas poderia colocar em risco os utilizadores da instituição, logo essa questão foi posta de parte e foram utilizados apenas mecanismos e *scripts* de análise de comportamento dos utilizadores, para entender o número de utilizadores que acederam ao website.

Apesar das dificuldades experienciadas, acredito que as consegui enfrentar e ultrapassar e que os objetivos dos projetos foram cumpridos, fortalecendo a minha capacidade de resolução de problemas. No primeiro projeto consegui criar *Evil Twins* que simulavam os *Access Points* legítimos da instituição e consegui que alunos se ligassem. Sendo possível, se fosse o pretendido, escalar o ataque. Com o segundo projeto, achei bastante interessante o facto de criar uma ideia de raiz para tentar

despertar interesse nos utilizadores e incentivá-los a aceder ao website que, também caso pretendido, injetaria algum *script* para escalar este ataque.

Esta experiência de estágio confirmou o meu gosto e interesse na área de Cibersegurança, estando bastante determinado em prosseguir a minha carreira neste vasto campo.

BIBLIOGRAFIA

A., Kamble. e D. Kshirsagar (2023). *Feature Selection in Wireless Intrusion Detection System for Evil Twin Attack Detection*. 2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT).

Abo-Soliman, M. e M. Azer (2017). *A Study in WPA2 Enterprise Recent Attacks*. Institute of Electrical e Electronics Engineers.

— (2018). *Tunnel-Based EAP Effective Security Attacks*. 2018 Tenth International Conference on Ubiquitous e Future Networks (ICUFN).

Almjamai, S. (2022). *A Comprehensive Taxonomy of Attacks and Mitigations in IoT Wi-Fi Networks: physical and data-link layer*. Linnaeus University, Faculty of Technology, Department of computer science e media technology (CM).

AlQahtani, S. e M. AlOraini (2012). *Resolving Wireless Security Limitations Using a New Wi-Fi Secure*. 2012 IEEE 12th International Conference on Computer e Information Technology.

Asaduzzaman, M., M. Majib e M. Rahman (2020). *Wi-Fi Frame Classification and Feature Selection Analysis in Detecting Evil Twin Attack*. 2020 IEEE Region 10 Symposium (TENSYMP).

Averin, A. e N Zyulyarkina (2020). *Malicious Qr-Code Threats and Vulnerability of Blockchain*. 2020 Global Smart Industry Conference (GloSIC).

Bani-Hani, R., Y. Wahsheh e M. Al-Sarhan (2014). *Secure QR code system*. 2014 10th International Conference on Innovations in Information Technology (IIT).

Bauer, K, H. Gonzales e D. Mccoy (2008). *Mitigating Evil Twin Attacks in 802.11*. 2008 IEEE International Performance, Computing e Communications Conference.

Briones, J., M. Coronel e P. Chavez-Burbano (2013). *Case of study: Identity theft in a university WLAN Evil twin and cloned authentication web interface*. 2013 World Congress on Computer e Information Technology (WCCIT).

Cassola, A. et al. (2013). *A Practical, Targeted, and Stealthy Attack Against WPA Enterprise Authentication*. Proceedings of NDSS, vol. 2013.

Catur Bhakti, M. A., A. Abdullah e L. T. Jung (2007). *EAP-based Authentication with EAP Method Selection Mechanism*. International Conference on Intelligent e Advanced Systems 2007.

Chatzoglou, E., G. Kambourakis e C. Kolias (2021). *Empirical Evaluation of Attacks Against IEEE 802.11 Enterprise Networks: The AWID3 Dataset*. IEEE Access (Volume: 9).

Esser, A. (2017). *A Wi-Fi intrusion testing framework for pentesters*. Instituto Universitário de Lisboa.

FreeRadius (23 de jan. de 2024). *Authorization, Authentication, Accounting*. Accessed: 2024-01-23. URL: https://networkradius.com/doc/current/concepts/introduction/AAA.html.

Goel, N., A. Sharma e S. Goswami (2017). *A Way to Secure a QR Code: SQR*. International Conference on Computing, Communication e Automation (ICCCA2017).

Jung, B. et al. (2022). *ZTA-based Federated Policy Control Paradigm for Enterprise Wireless Network Infrastructure*. 2022 27th Asia Pacific Conference on Communications (APCC).

Kharraz, A. et al. (2014). *Optical Delusions: A Study of Malicious QR Codes in the Wild*. 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems e Networks.

Khasanova, A. (2021). *Detection of Attacks on Wi-Fi Access Points*. 2021 IEEE Conference of Russian Young Researchers in Electrical e Electronic Engineering (ElConRus).

Krombholz, K. et al. (2015). *QR Code Security - How Secure and Usable Apps Can Protect Users Against Malicious QR Codes*. 2015 10th International Conference on Availability, Reliability e Security.

Kumar, A. e P. Paul (2016). *Security Analysis and Implementation of a Simple Method for Prevention and Detection against Evil Twin Attack in IEEE 802.11 Wireless LAN*. 2016 International Conference on Computational Techniques in Information e Communication Technologies.

Kuo, E., M. Chang e E. Kao (2018). *User-Side Evil Twin Attack Detection Using Time-Delay Statistics of TCP Connection Termination*. International Conference on Advanced Communications Technology(ICACT).

Kyungroul, L. et al. (2016). *Analysis on Manipulation of the MAC Address and Consequent Security Threats*. Association for Computing Machinery.

Maharjan, Nilaa (2020). *Demonstration of Mass JavaScript Injection for Cryptojacking using MITM*. Research Gate.

Martens, S. et al. (2018). *Multifunctional sequence-defined macromolecules for chemical data storage*. Nature Communications.

Modi, V. e C. Parekh (2017). *Detection & Analysis of Evil Twin Attack in Wireless Network*. International Journal of Advanced Research in Computer Science.

Mohanan, Remya (2024). *What Is an Evil Twin Attack? Definition, Detection, and Prevention Best Practices*. Website. https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-evil-twin-attack/.

Networks, Aruba (29 de jan. de 2024). *How MAC authentication works*. Accessed: 2024-01-29. URL: https://www.arubanetworks.com/techdocs/AOS-CX/10.13/HTML/security_6200-6300-6400/Content/Chp_Port_acc/mac-aut-fl-10.htm.

Palamà, I et al. (2023). *Attacks and vulnerabilities of Wi-Fi Enterprise networks: User security awareness assessment through credential stealing attack experiments*. ELSEVIER.

Pasala, A. e S. Mukherjee (2024). *Variable masking pattern-based QR codes for high security*. 2024 IEEE International Conference on Electronics, Computing e Communication Technologies (CONECCT).

Rafsanjani, A. et al. (2023). *QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework*. IEEE Access (Volume: 11).

Rubens, A. et al. (2000). *Remote Authentication Dial In User Service (RADIUS) - RFC 2865*. Accessed: 2024-07-31. URL: https://datatracker.ietf.org/doc/rfc2865/.

Sharma, V. (2012). *A Study on Malicious QR Codes*. International Journal of Computational Intelligence e Information Security.

Shrivastava, P., M Jamal e K. Kataoka (2020). *EvilScout: Detection and Mitigation of Evil Twin Attack in SDN Enabled WiFi*. Institute of Electrical e Electronics Engineers.

Soon, T. (2008). *QR Code*. Synthesis Journal.

Staddon, E., V. Loscri e N Mitton (2021). *Attack Categorisation for IoT Applications in Critical Infrastructures, a Survey*. Appl. Sci. 2021, 11(16), 7228.

Syahrial, S. et al. (2024). *Design and Implementation of A WiFi Password Cracking Tool Based on the Evil Twin Method*. 2024 International Conference on Electrical Engineering e Computer Science (ICECOS).

Tick, A. (2018). *IT Security as a Special Awareness at the Analysis of the Digital/E-learning Acceptance Strategies of the Early Z Generation*. 22nd IEEE International Conference on Intelligent Engineering Systems.

Tiwari, S. (2016). *An Introduction To QR Code Technology*. 2016 International Conference on Information Technology.

Vidas, T. et al. (2013). *QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks*. International Conference on Financial Cryptography e Data Security.

Vink, M. (2020). *A Comprehensive Taxonomy of Wi-Fi Attacks*. Radboud University Nijmegen.

Yong, K., K. Chiew e C. Tan (2019). *A survey of the QR code phishing: the current attacks and countermeasures*. 2019 7th International Conference on Smart Computing & Communications (ICSCC) (GloSIC).

D E C L A R A Ç Ã O

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"Teste e validação de ameaças informáticas em contexto de Ensino Superior"*, é original e foi realizado por Estudante Luís Miguel Babo Costa (2222886) sob orientação de Professor Doutor Paulo Jorge Ferreira Batista Pinheiro Cordeiro (paulo.cordeiro@ipleiria.pt).

Leiria, março de 2025

Estudante Luís Miguel Babo Costa