Introdução ao Footprinting e Scanning de Redes: Análise de Protocolos de Comunicação e Avaliação de Vulnerabilidades

Universidade de Aveiro

Henrique Gala, Iury Figueredo

VERSAO

Introdução ao Footprinting e Scanning de Redes: Análise de Protocolos de Comunicação e Avaliação de Vulnerabilidades

Dept. de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

Henrique Gala, Iury Figueredo
(131898) henriqueggala@ua.pt, (132655) iuryfigueredo@ua.pt

2 de novembro de 2025

Resumo

Este projeto no âmbito da Licenciatura em Engenharia de Computadores e Informática (LECI) explora a relevância da Cibersegurança (CS) e do Reconhecimento de redes (RR) para a arquitetura de sistemas. Sendo analisado de forma rigorosa as técnicas de *port scanning* e Footprinting (FP) **WikipediaFootprinting**, de muita importância para avaliar a superfície de ataque em qualquer rede. **O foco principal** está nas análises detalhadas de protocolos como o Protocolo de Transmissão (TCP) e Protocolo de Mensagens de Controlo da Internet (ICMP), detalhando como *scans* furtivos, como o *NULL* Scan, exploram as regras protocolares (Reset (RST) vs. *drop*) assim podendo inferir o modo da porta sem realizar por completo o *three-way handshake*.

A discussão crítica tem em causa a solução dos mecanismos de defesa. Demonstra-se que o **ACK Scan** é decisivo para o engenheiro, porque distingue e mapeia as regras de filtragem entre *firewalls* sem e com estado. Identificou-se ainda fragilidades nas arquiteturas, como por exemplo a previsibilidade do **Initial Sequence Number (ISN)** em *firmwares* Unfied Extensible Firmware Development (UEFI), que perturba a segurança da TCP *Session Hijacking*.

A robustez da arquitetura de segurança é suportada pelo desempenho do *kernel*, onde conceitos chave como *zero copy* e *tasklets* concretizam a escalabilidade necessária para a deteção de intrusos. As **decisões éticas e legais** são abordadas com forte importância, reforçando que o domínio destas técnicas avançadas deve ser aplicado somente no âmbito do Ethical Hacking (EH) e com extrema responsabilidade.

Para finalizar, esta pesquisa fornece o **conhecimento fundamental** para que os futuros engenheiros possam proteger e defender sistemas de forma proativa.

Índice

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

Com a evolução da tecnologia e o aprimoramento dos ambientes digitais, viu-se uma crescente necessidade no aumento da CS, garantindo a integridade e confidencialidade da arquitetura de rede e dos sistemas. Desta forma, estão sendo desenvolvidos vários métodos para assegurar a proteção dos usuários, dos quais serão tratados nesse relatório.

O desenvolvimento desse trabalho está fundamentado nos três pilares da CS. O primeiro é o *FP*, que é uma técnica usada por especialistas da área para recolher informações sobre sistemas computacionais e seus donos. O segundo é o *Scanning*, que ocorre após o *FP* e tem como principal objetivo identificar pontos fracos em sistemas informáticos, através do mapeamento da arquitetura da rede e os serviços em execução, com intenção de previnir a exploração por cibercriminosos. Isso inclui falhas de configuração, bugs, etc.. Por fim, "Avaliação de Vulnerabilidades"é, resumidamente, um método para identificar, avaliar e relatar possíveis brechas de segurança no ambiente virtual de uma organização e ocorre no final do processo de Reconhecimento de Redes (RR).

Pelo facto das técnicas de *Scanning* terem como fundamento a análise das respostas dos protocolos na Camada de Transporte, um dos objetivos desse trabalho é explicar o TCP e o Protocolo de Internet (IP), e o ICMP, de forma a que o leitor fique a entender seus principais conceitos. Para além disso, temos como meta detalhar as técnicas de *Scanning* e analisar as implicações na arquitetura da rede no âmbito da segurança virtual.

Este documento está dividido em quatro capítulos. Depois desta introdução, no Capítulo 2 é apresentada a metodologia seguida, no Capítulo 3 são apresentados os resultados obtidos, sendo estes discutidos no Capítulo 4. Finalmente, no Capítulo 5 são apresentadas as conclusões do trabalho.

Capítulo 2

Metodologia

O trabalho realizado tem natureza predominantemente teórica e analítica, dando ênfase na compreensão de conceitos fundamentais do*EH*. A metodologia seguida teve como foco garantir a precisão técnica dos fundamentos e uma descrição bem fundamentada dos métodos de reconhecimento.

2.1 Análise das Especificações de Protocolo

A base teórica do trabalho está assente na pesquisa e análise das Especificações de Protocolo (EP) definidas pelo Internet Engineering Task Force (IETF), que regem a ação das comunicações via internet. Esta abordagem foi necessário para fundamentar de forma técnica as inferências de *scanning*.

2.1.1 Especificações Chave

Foram analisadas duas especificações chave:

RFC 793 (*Transmisssion Control Protocol* - TCP): RFC793 Essa análise permitiu que fosse possível dar a entender sobre o funcionamento dos campos de controlo (*flags*) como **SYN**, **ACK** e **RST**. O tipo de funcionamento padrão da *flag* RST em ambiente de conexão fechada (ou inexistentes) é a ferramenta que justifica as técnicas de ***scanning* furtivo** (*stealth scanning*).

RFC 792 (*Internet Control Message Protocol* - ICMP): RFC792 Esta especificação forneceu o conhecimento necessário para a etapa de *FP* (através das mensagens *Echo/Echo Reply*) de forma a justificar a lógica de deteção de portas **User Datagram Protocol (UDP)** fechadas, através da mensagem ***Destination Unreachable***(Tipo 3, código 3 - *Port Unreachable*).

2.2 Contextualização e Validação

Toda a informação que foi retirada dos protocolos foi contextualizada e aplicada através da revisão de literatura específica sobre a **aplicação prática** dessas regras, como por exemplo nas ferramentas de *Port Scanning (PS)* mais utilizadas. Foi estudada a forma de operação de técnicas como o *Números de confirmação (ACK)* scan para **mapeamento de *firewalls***.

Adicionalmente, nosso estudo baseou-se na análise de vulnerabilidades da Camada de Transporte, realizando uma conexão da parte teórica do campo ***Números de sequência (SEQ)*** (RFC 793) à atividade real. Esta pesquisa foi validada com uma revisão do vetor de ataque *TCP Session Hijacking* e a sua importância nos dias de hoje, demonstrado pela vulnerabilidade CVE-2023-45237 **CVE202345237** (falha de ISN previsível em firmware).

2.3 Declaração

O presente trabalho contou com o apoio da ferramenta de Inteligência Artificial generativa Gemini 3 para efeitos de correção ortográfica, revisão gramatical e otimização textual. Contudo, todas as ideias, conceitos e descrições foram integralmente desenvolvidos pelos autores, baseando-se na investigação das referências bibliográficas citadas.

Capítulo 3

Resultados e Descrição

Este capítulo contém a base técnica necessária para compreender o processo de RR e as suas vulnerabilidades.

3.1 Fundamentos de Protocolo para o Scanning

Esta secção estabelece as regras da Camada de Transporte e Camada de Internet que as ferramentas de scanning exploram.

3.1.1 O Protocolo de Controlo de Transmissão (TCP)

RunmoduleTCPSession

O **Protocolo de Controlo de Transmissão (TCP)**, especificado na **RFC 793**, é a fundação da comunicação fiável e orientada à conexão na Internet. Embora o seu propósito seja garantir a entrega ordenada dos dados, as regras estritas que governam a sua operação (como a resposta a segmentos inesperados) são o ponto de exploração central para as técnicas de scanning furtivo. O TCP é responsável por transportar informações vitais, incluindo os endereços de host de origem e de destino.

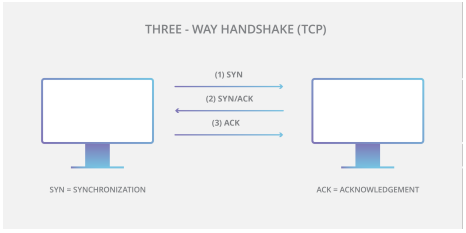

Figura 3.1: Ilustração do *Three-Way Handshake*

Elementos do Cabeçalho TCP

O cabeçalho TCP utiliza campos de 16 e 32 bits para manter o estado da conexão e assegurar a fiabilidade dos dados.

- **Portas**: Os campos `Source Port` e `Destination Port` utilizam ambos 16 bits para identificar as aplicações envolvidas na comunicação.
- **Sequence Number (Número de Sequência) (32 bits)**:
  - **Transferência de Dados**: Indica o número do byte cumulativo do primeiro byte de dados de aplicação contido no segmento TCP.
  - **Início de Conexão**: Quando a flag `SYN` está presente, este campo contém o **Initial Sequence Number (ISN)**. O flag `SYN` consome uma unidade de espaço de número ISN + 1
- **Acknowledgement Number (Número de Confirmação) (32 bits)**:
  - **Controlo de Fluxo**: Utilizado pelo recetor para informar o emissor qual é o próximo byte de dados que espera receber. O valor é o número de sequência mais alto recebido, mais um.
  - **Piggybacking**: Após o estabelecimento da conexão, a maioria dos segmentos de dados enviados de volta e para a frente terão a flag `ACK` e um `Acknowledgement Number` válidos, um processo denominado **piggybacking** que garante o fluxo de dados contínuo e fiável.

Estabelecimento e Encerramento

A conexão TCP é estabelecida através do procedimento **Three-Way Handshake** (aperto de mão a três vias), um ciclo de respostas entre os dois *endpoints*. O encerramento (`CLOSE`) indica que um utilizador não tem mais dados para enviar, mas pode continuar a receber dados até que o outro lado também envie um `CLOSE`.

O Papel Crítico do `RST` (Reset) no Scanning

A flag `RST` (*Reset*) é a base do *stealth scanning* e dos métodos de inferência de portas. A regra fundamental do TCP afirma que:

> Se um segmento de entrada for recebido por uma porta onde nenhum processo está à escuta (estado `CLOSED`) e o segmento não for um reset válido, um `RST` deve ser enviado em resposta.

Isto significa que o envio de um `SYN` (ou qualquer outro flag isolado) para uma porta fechada provoca a resposta `RST`, confirmando o estado fechado da porta.

Lógica dos Scans Furtivos (*Stealth Scans*)

As técnicas `FIN`, `NULL` e `XMAS` exploram a exceção a esta regra de reset para alcançar a furtividade:

- **Porta Fechada**: Se o sistema receber um segmento que não tenha `SYN`, `RST` ou `ACK` ativos (como nos Scans `FIN`, `NULL` ou `XMAS`), a regra do `RST` é acionada, e um `RST` é enviado em resposta.
- **Porta Aberta**: Se o segmento for recebido por uma porta aberta (onde um processo está à escuta), mas o segmento for inválido ou inesperado, o sistema deve **descartar** (*drop*) o pacote e **não responder**.

A ausência de uma resposta `RST` perante um segmento `FIN`, `NULL` ou `XMAS` permite ao scanner inferir que a porta está **aberta** (ou filtrada), pois evita o handshake completo e a respetiva deteção por mecanismos de defesa de rede. A furtividade refere-se a qualquer técnica que utilize pacotes que parecem ser erros ou restos de sessões antigas, evitando a deteção.

3.1.2 O Protocolo de Mensagens de Controlo da Internet (ICMP)

O Protocolo de Mensagens de ICMP, detalhado na RFC 792, é um protocolo auxiliar que funciona como parte adicional do IP. O seu objetivo principal não está focado no transporte de dados, mas sim em fornecer um feedback e reportar erros sobre problemas no ambiente de comunicação, como quando um *datagram* não consegue alcançar seu destino

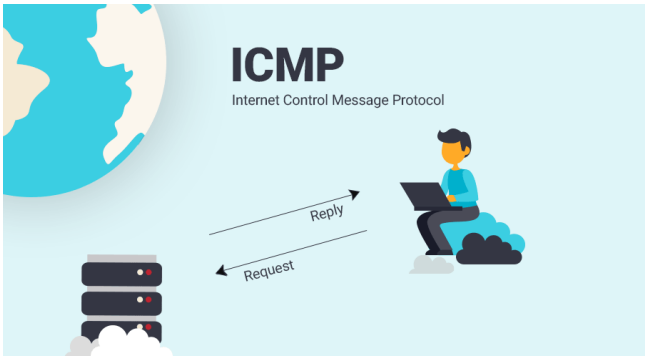

Figura 3.2: Ilustração do *ICMP Demonstração*

Mensagens Chave para o RR

O modo em que o **ICMP** relata os erros é explorado nas duas primeiras fases do reconhecimento de redes: O ***Host Discovery*** e o ***Scanning UDP***.

***Echo Request* e Echo Reply (Tipos 8 e 0):** Estas mensagens são a base do ***FP* Ativo** (conhecido como *ping*).

- O emissor envia uma mensagem ***Echo Request*** (Tipo 8) e, se o host alvo estiver ativo e não filtrado (ou seja, estiver apto a receber e enviar mensagens), ele responde invertendo os endereços e alterando o código do tipo para ***Echo Reply*** (Tipo 0).
- Esta troca confirma a presença do host na rede.

***Destination Unreachable* (Tipo 3):** Esta mensagem é crucial para o *scanning*, pois reporta falhas de entrega.

- O ICMP inclui vários códigos de erro, entre eles o **Código 3 (*Port Unreachable*)** sendo o mais relevante para o ***scanning UDP***.
- **Lógica de *Scanning UDP***: Como o Protocolo de Datagrama do Utilizador (**UDP**) é *connectionless* (sem conexão) e não exige confirmação, a única maneira de um scanner ter a certeza de que uma porta está **fechada** é quando o recetor responde com uma mensagem ICMP ***Destination Unreachable*** (Tipo 3, Código 3).
- Se a porta estiver **aberta**, o host apenas descarta o pacote e **não envia resposta**.

Limitações do ICMP no Scanning

A utilização do ICMP no *scanning* apresenta limitações que afetam a velocidade e a precisão das varreduras:

- **Taxa de Limitação (*Rate Limiting*)**: Muitos sistemas operativos e dispositivos de rede limitam a taxa de envio de mensagens ICMP *Port Unreachable* por padrão. Isto força os scanners a abrandar o ritmo das sondas UDP para evitar que o host alvo descarte as mensagens de erro por excesso de tráfego, prejudicando significativamente a velocidade do ***UDP* Scan**.
- **Filtragem Total**: Os firewalls são frequentemente configurados para bloquear todas as mensagens do ICMP de entrada ou saída para esconder a presença de hosts, tornando os scans de *Host Discovery* ineficazes.

3.1.3 O Processo de Reconhecimento de Redes

RainforestScanners A forma como o scanning funciona se baseia na aplicação direta das regras de protocolo definidas para as Camadas de Internet e de Transporte. O processo de reconhecimento de redes segue uma progressão de raciocínio iniciada pela identificação de alvos ativos (**Host Discovery**), antes de avançar para a varredura e análise detalhada das portas (**Port Scanning**).

Descoberta de Hosts (*Host Discovery*)

A primeira etapa do reconhecimento ativo de uma rede é a Descoberta de Hosts, que tem como finalidade determinar quais endereços IP correspondem a sistemas ativos. Esta técnica explora as mensagens de controlo definidas no **Protocolo de Mensagens de Controlo da Internet (ICMP)**, conforme a **RFC 792**.

O método mais comum está assente nas mensagens **Echo Request** e **Echo Reply**:

- **Sonda**: O scanner envia uma mensagem **ICMP Echo Request (Tipo 8)** para o endereço alvo. A pesquisa do scanner tem como meta a determinação do estado do host (se está ligado e a funcionar).
- **Resposta**: Se o host alvo estiver ativo e o tráfego ICMP não estiver filtrado, ele responde com uma mensagem **Echo Reply (Tipo 0)**. Para esta resposta ser formada, o host recetor realiza uma inversão nos endereços de origem e destino, alterando o código do tipo e recalculando o checksum.
- **Consequência**: A receção do Echo Reply confirma a presença do host na rede, permitindo que a fase de Port Scanning seja iniciada.

Tabela 3.1: Comportamento de Inferência de Host Ativo (RFC 792)

Sonda (Tipo)	**Protocolo**	**Resposta do Host Alvo**	**Inferência do Scanner**
Echo Request (Tipo 8)	ICMP	Echo Reply (Tipo 0) recebido.	Host Ativo (Online)
Echo Request (Tipo 8)	ICMP	Nenhuma Resposta após timeout.	Host Inativo ou Tráfego ICMP Filtrado
Envio de Pacote (qualquer)	ICMP	Destination Unreachable (Tipo 3, Código 1)	Host Inativo ou Inalcançável

3.1.4 Técnicas de Descoberta de Portas (*Port Scanning*)

Esta secção detalha a aplicação prática dos princípios fundamentais de protocolo para a avaliação do estado das portas. Isso é possível através de **flags de controlo do TCP** e os **códigos de erro do ICMP** para deduzir o estado da porta, sejam elas abertas, fechadas ou filtradas.

Scanning por Conexão Total (*Full Connect*)

Esta metodologia é considerada a abordagem mais básica, mas menos furtiva, de varrimento de portas. O **Full Connect Scan** tenta estabelecer uma conexão completa, concluindo o **Three-Way Handshake do TCP** (`SYN` → `SYN-ACK` → `ACK`).

- **Lógica**: O scanner permite que a conexão seja estabelecida na sua totalidade, enviando o segmento `ACK` final.
- **Vantagem**: É um método simples e confiável.
- **Desvantagem**: Pelo facto de ser completado o handshake, esta técnica é facilmente detetada, deixando registos completos de conexão nos logs do sistema operativo alvo, o que a torna ineficiente em situações em que a furtividade é essencial.

Scanning Furtivo (*SYN Scan*)

O **SYN Scan**, também conhecido como **half-open scanning**, é a técnica de port scanning mais comum e eficiente. É baseada na regra da **RFC 793** para a flag `RST` e na lógica do Three-Way Handshake para inferir o estado da porta sem completar a conexão.

- **Lógica da Sonda**: O scanner envia um pacote com apenas a flag `SYN` ativa.
- **Consequências**:
  - **Porta Aberta**: O alvo responde com um pacote `SYN + ACK` (o segundo passo do handshake). O scanner não envia o `ACK` final, mas sim um `RST` imediato para impedir que a conexão seja estabelecida. O alvo não regista uma conexão completa, garantindo a classificação de "furtivo" (daí o termo *half-open*).
  - **Porta Fechada**: O alvo responde imediatamente com um pacote `RST` (*Reset*) e `ACK`. Esta resposta confirma, segundo a RFC 793, que a porta se encontra de facto fechada.

Scanning Stealth Baseado em Flags (FIN, NULL, XMAS)

As técnicas de varrimento baseadas em flags exploram a regra fundamental do TCP detalhada na **RFC 793** relativa ao modo de atuação do flag `RST` (*Reset*) em portas fechadas. Têm como principal objetivo criar pacotes que pareçam ser erros ou restos de sessões antigas, para evitar mecanismos de reconhecimento de defesa de rede.

A lógica é a seguinte:

- **Porta Fechada**: A RFC 793 diz-nos que, no caso de um sistema receber um segmento TCP que não faça parte de uma conexão ativa e que não contenha os flags `SYN`, `RST` ou `ACK` ativados (como nos scans `FIN`, `NULL` ou `XMAS`), o sistema deve enviar um pacote `RST` como resposta.
- **Porta Aberta**: Caso contrário, se a porta estiver a escutar (aberta), o sistema deve **descartar** (*drop*) o pacote sem enviar qualquer resposta.
- **Inferência Furtiva**:
  - **Resposta** `RST`: Confirma que a porta está **Fechada**.
  - **Sem Resposta**: Indica que a porta está **Aberta** (ou filtrada por um firewall).

Esta distinção, que explora a ausência de uma resposta ao invés de sua presença, é a essência da furtividade.

Scanning UDP

NmapPortScan O User Datagram Protocol (UDP) é um protocolo *connectionless* (sem conexão), utilizado na maior parte das vezes para transmissões sensíveis ao tempo (como DNS e VoIP). A ausência de mecanismos de handshake e de confirmação torna o scanning UDP relativamente mais lento e desafiador do que o TCP.

- **A Sonda**: O scanner envia um datagram UDP (geralmente vazio) para a porta alvo.
- **Consequências (Base ICMP)**:
  - **Porta Fechada**: O host alvo deve retornar uma mensagem **ICMP Destination Unreachable** (Tipo 3, Código 3 – *Port Unreachable*), conforme está descrito na **RFC 792**. Esta é a única resposta que confirma uma porta fechada.
  - **Porta Aberta/Filtrada**: Se a porta estiver aberta, o serviço pode responder com um datagram UDP (confirmando que está aberta). Se a porta estiver filtrada ou aberta sem responder, o scanner acaba por não receber resposta, classificando a porta como `open|filtered`.

- **Limitação da Taxa**: O principal desafio é a **limitação da taxa de ICMP** (*rate limiting*) imposta por muitos hosts, o que faz com que o scanner tenha que abrandar e esperar por *timeouts* longos, atrasando de forma significativa a varredura.

Scanning ACK (Mapeamento de Firewall)

O **ACK Scan** é uma técnica única, pelo facto de não ter como objetivo a determinação do estado de uma porta. Seu propósito é **mapear o ruleset do firewall**, determinando se ele é *stateful* ou *stateless*.

- **A Sonda**: O scanner envia um pacote apenas com a flag `ACK` e um número de sequência aleatório, sendo que esse pacote não faz parte de nenhuma conexão ativa.
- **Consequências**:
  - **Unfiltered (Firewall Stateless)**: Se o firewall permitir o tráfego de entrada e o pacote chegar ao host, este, por não ter conexão ativa, irá responder com um pacote `RST`. Assim, o scanner classifica a porta como **unfiltered** (alcançável), indicando que o firewall é **stateless** e não está a inspecionar o estado da sessão.
  - **Filtered (Firewall Stateful)**: Caso o firewall não receba nenhum pacote (descartando-o silenciosamente) ou se o scanner receber uma mensagem **ICMP Destination Unreachable** (com códigos como 0, 1, 10, 13), a porta é classificada como **filtered**. Isto indica que um firewall **stateful** inspecionou o pacote e bloqueou por não fazer parte de uma sessão estabelecida.

Tabela 3.2: Sumário das Técnicas de Port Scanning

Técnica de Scanning	**Protocolo Base**	**Lógica de Inferência**	**Vantagem/Objetivo**
Full Connect	TCP (RFC 793)	Completa o Handshake (SYN → SYN-ACK → ACK).	Confiável; Padrão não-furtivo.
SYN Scan (Half-Open)	TCP (RFC 793)	Envia SYN; Resposta SYN-ACK → Aberta. Resposta RST → Fechada.	Furtivo (não completa o handshake).
FIN/NULL/XMAS	TCP (RFC 793)	Nenhuma Resposta → Aberta. Resposta RST → Fechada.	Furtivo (evita detecção por firewalls simples).
UDP Scan	UDP (RFC 792 ICMP)	ICMP Port Unreachable → Fechada. Nenhuma Resposta → Aberta/Filtrada.	Determinar serviços sem conexão (DNS, VoIP).
ACK Scan	TCP / ICMP	Recebe RST → Unfiltered (Firewall stateless). Nenhuma Resposta → Filtered (Firewall stateful).	Mapear o ruleset do firewall.

Capítulo 4

Análise

Esta análise tem como principal objetivo interpretar de forma crítica os resultados obtidos no Capítulo 3, relacionando-os com o funcionamento interno dos protocolos abordados, com os **mecanismos de proteção que existem nas redes modernas** e com a perspetiva **ética e legal** associada ao uso das técnicas de FP e *scanning* em redes de comunicação. A análise possui uma estrutura que fornece ao leitor não apenas uma compreensão técnica, mas também uma visão sistemática do impacto do RR em ambientes reais, evidenciando a sua relevância para a engenharia de redes e conceção de sistemas de segurança.

Para tal, são examinadas as interações entre os **mecanismos de defesa** (como *firewalls* e sistemas de deteção de instrusões) e as técnicas de *scanning*, avaliando o seu efeito sobre:

1. a arquitetura dos sistemas;
2. a robustez dos sistemas;
3. as implicações éticas e legais do uso dessas técnicas.

4.1 Interpretação dos Scans

O *scanning* de portas é a **primeira fase do reconhecimento**, onde o invasor ou analista de segurança procura mapear a **superfície de ataque** de uma rede. A eficácia destas técnicas reside na interpretação cuidadosa das respostas, que são ditadas pelas **regras principais dos protocolos** ICMP e TCP.

4.1.1 Aprofundamento das Regras de Protocolo para a Furtividade

A lógica do *scanning* furtivo (*stealth*) explora o comportamento dos Sistema Operacional (SO) **Russinovich2020OSStory**em relação aos segmentos TCP não esperados.

- **O papel do RST:** De acordo com a RFC 793, se uma portats **NÃO** tem um processo à escuta (estado de *CLOSED*, o sistema envia um RST em resposta a qualquer outro segmento de entrada (em exceção a outro RST). O concebimento de um pacote RST em resposta a um segmento Synchronize (SYN)indica que a porta encontra-se fechada.

- **O Scan Furtivo (*NULL* Scan):** As técnicas como **NULL Scan** enviam um segmento TCP sem quaisquer flags definidos (SYN, RST, ACK, etc.) A lógica de resposta a este segmento é o fator que mostra o estado atual de uma porta:

  1. **Porta fehada:** Envia um ***RST*** como resposta, em ordem a regra do protocolo.
  2. **Porta aberta:** O sistema deve **descartar (*drop*)** o pacote e não enviar qualquer resposta, assim a ação tornando-se inivisível ao *firewall* mais simples.

Esta diferença de comportamento é o ponto de partida para o **reconhecimento passivo**, permitindo o analista diferenciar portas fechadas de abertas sem completar o *three-way handshake* - Processo de 3 etapas usados pelo TCP a fim de estabelecer uma **conexão confiável** cliente/servidor.

4.1.2 A Vulnerabilidade do Initial Sequence Number (ISN)

IBMVulnerabilityAssessment

A segurança na arquitetura TCP, núcleo da comunicação na Internet, depende da imprevisibilidade e aleatoriedade dos números sequenciais . O TCP utiliza os SEQ e de confirmação ACK para garantir a organização e confiabilidade da entrega de dados, um conceito conhecido como *piggybacking*. O campo *Sequence Number* (32 bits) contém o ISN no segmento SYN inicial.

Uma vulnerabilidade crítica, do tipo vista no *EDK2 Network Package* (CVE-2023-4527), represeta um erro na geração deste número, permitindo que um invasor consiga **prever o ISN.**

- **Implicações no Session Hijacking**: O ataque - TCP *Session Hijacking* depende de forma exclusiva da identificação dos números SEQ e ACK atuais da sessão em causa para a concretização de pacotes válidos que o servidor validará. Porém se o ISN for previsível, o invasor dispensa ou torna mais fácil a fase de **rastreamento da conexão** *Tracking the Connection*.

- **Ataque Facilitado:** Não por *sniffing* o tráfego para conseguir o posterior número *seq* certo, a previsibilidade do ISN permite o intruso simplificar ou simplesmente contornar esta parte. Isso compromete a confiabilidade na **Camada de Transporte** e permite que o invasor seguir para a fase de **desincronização da conexão**, onde coloca seus própios pacotes maliciosos na sessão.

A ocorrência desta fragilidade promove a importância de uma **arquitetura de segurança** que engloba desde à pilha de rede ao *firmware* (UEFI).

4.2 Sistemas de Segurança

A capacidade de eficiência do *port scanning* como uma técnica de reconhecimento é enfrentada diretamente pela arquitetura de segurança, como os Deteção de intrusão (IDS) e *Firewalls*. A visualização dos scans, nomeadamente o **ACK Scan**, permite um engenheiro determinar também as **regras de filtragem** implementadas na rede, não só apenas portas abertas.

A tabela 4.1 sumariza a eficácia das técnicas de *scanning* que exploram as ações dos protocolos contra os mecanismos defensivos com estado.

Tabela 4.1: Eficiência do Scanning contra Ssistemas Defesinvos

Tipo de scan	**Necessita privilégios?**	**Furtividade Alta?**	**Detetável por Firewall Stateful?**
Connect Scan	Não	Baixa	Sim (Fácil)
SYN Scan	Sim	Média	Sim (Moderado)
NULL Scan	Sim	ALta	Não (Simples)
ACK Scan	Sim	N/A (Mapeia Regras)	Não (Mapeamento somente)

4.2.1 O ACK Scan

Este é um scan de mapeamento de regras de *firewall*, e não para determinação do estado da porta. Ele atua enviando um segmento TCP com somente o ACK definido. Como o ACK só deve ser definido em segmentos que pertencem a uma conexão TCP já existente, o segmento não é um requisito válido para começar uma comunicação.

A utilização do mesmo baseia-se no entendimento da seguinte reposta:

- **Porta Não Filtrada (Resposta = RST):** Se o sistema em causa (ou o *firewall* anterior ao mesmo) não filtrar ativamente a porta ou não possuir a informacão de uma conexão com o devido segmento, será respondido com um pacote RST. O *scanner* designa a porta como **não filtrada**, confirmando que é alcançável pelo pacote ACK, porém seu estado final (fechado ou aberto) -> **indefinido**.
- **Porta Filtrada (Sem Resposta / ICMP)** Se for filtrada por um *firewall*, este irá **descartar** o pacote pois nao há resposta ou enviar uma mensagem de erro *Destination Unreachable*). Por fim o scanner rotula a porta como **filtrada**.

A diferença entre não ter qualquer resposta ou receber um RST é o ínicio para o engenheiro inferir se um *Firewall Stateful* está em uso. Um *Firewall Stateful* encontra o estado das conexões ativa e descaratia um pacote ACK o qual não pertenca a qualquer sessão já estabelecida.

4.2.2 Arquitetura do Kernel

A competência de um sistema de segurança (como um *firewall* baseado em software ou um IDS) de encontrar scans e tratar tráfego de rede é unido a arquitetura do SO.

- **Deteção inteligente:** O *port sacnning*, principalmente em grande escala, pode originar um volume de tráfego que necessita um enorme desempenho do kernel. A preferência por mecanismos bem estruturados (como as *Deferred Procedure Calls* ou *DPCs* no Windows e os ***tasklets*** no Linux) solucionou problemas de escalabilidade em sistemas *multi-processor*, possibilitando a **mitigação e deteção** de scans mais produtivo e estável. A engenharia atual obriga que os sistemas de segurança sejam implementados em *kernels* preemptíveis e reentrantes, essenciais para a caspacidade de resposta.

- **Desempenho e Escalabilidade:** A pilha IP/ICMP e seus devidos drivers de rede são dependentes da eficácia do *kernel*. Problemas relacionados com a arquitetura como o *Thundering Herd* (onde múltiplas threads acionam para a mesma requisição de rede, posteriormente ocorrendo picos de atividade excessivos), e a falta de **mecanismos de cópia zero** (*zero copy*) em certas arquiteturas, impactaram a velocidade de resposta e escalabilidade em cargas de trabalhos em empresas.

4.3 A Visão Legal e Ética na Engenharia de Computadores

O entendimento aprofundado das técnicas de *port scanning* e das vulnerabilidades de protocolos, como a previsibilidade do **ISN**, é o que diferença o **formado de LECI** em relação a um utilizador normal. Este conhecimento abrangente impõe um dever ético e legal significativo no suporte e no esquema de sistemas de segurança.

4.3.1 Decisão e Linha Ética

A diferença entre uma atividade maliciosa e o reconhecimento legítimo consiste na **autorização** e na **intenção**.

- O *expert* de segurança (*Ethical Hacker*) aproveita de scans ocultos, como o ***NULL Scan*** ou o ***SYN Scan***, com o objetivo de **mitigar e encontrar falhas** antes que um invasor as investigue. Esta é uma etapa crucial para o projeto de sistemas confiáveis e robustos.

- Ao construir um sistema, o engenheiro deve ter a habilidade de **prever a utilização de scans** e outras técnicas de ataque, incorporando certas etapas de segurança:

1. **Configuração de *Firewalls Stateful*:** A arquitetura de segurança necessita de dar prioridade a *firewalls* com estado (*stateful*) capazes de rejeitar e detetar pacotes anómalos, como segmentos ACK ou SYN os quais não fazem parte de um *handshake* estabelecido.
2. **Monitorização (IDS):** IDS devem ser programados para identificar um comportamento *scanning* (e.g., diversos pacotes SYN não seguidos de ACK). Para isso, o kernel do SO deve utilizar arquiteturas eficazes, como os ***tasklets***, assim garantindo que o tráfego de rede seja processado de forma escalável, sem qualquer sobrecarregamento.

A **escolha de uma decisão no projeto de sistemas** deve ser informada pela lógica de invasão, assim possibilitando que os pontos fracos explorados pelos *scans* (como a ausência de respostas em portas abertas por *NULL Scans*) sejam detetadas pela camada de defesa.

4.3.2 Legalidade do Reconhecimento

Apesar do conhecimento destes utensílios serem vitais, a aplicação dos mesmos sem a permissão por escrito e explícito do propietário do sistema constitui, na maioria das jurisdições, um **violação legal.**

- A utilização de técnicas para **desincrozinar a conexão** (como o envio de apcotes *null* ou RST/SYN forjados) para realizar uma *TCP Session Hijacking* é obviamente uma criminalidade ao ser realizado contra um alvo não autorizado.
- Por mais inofensivo que seja **mapear a arquitetura de segurança** de terceiros por *port scanning*, independente do *scan* ser considerado furtivo ou ¨passivo¨, pode muitas vezes ser entendido como uma tentativa de **acesso não autorizado**.

A importância deste tópico para a Engenharia de Computadores e Informática (ECI) é a obrigação de um engenheiro operar em relacão a um quadro de **implicações éticas e legais**, reconhecendo que a crucial averiguação de sistemas e capacidade de análise pode e deve ser usada só e somente para objetivos de melhoria e defesa da arquitetura de segurança.

Capítulo 5

Conclusões

Este trabalho se aprofundou em domínios da CS e do RR, tendo como foco a exploração dos protocolos de rede e sua mecânica rigorosa, fundamental para a formação em ECI. A missão de contextualizar as técnicas de *port scanning* e FP, e de estudar criticamente a arquitetura de segurança, foram conseguidos. A análise detalhada do ICMP e do TCP tiveram a função de fundamento para compreender a manipulação inerente ao design protocolear e como a mesma pode ser para mapear a **superfície de invasão de um sistema**.

O que foi encontrado demonstra que a eficácia do *port sacnning* é proveniente da interpretação precisa das **bandeiras (*flags*) TCP**, tais como SYN, RST e ACK. A técnica de *scanning* furtivo, por o exemplo o **NULL Scan**, capitaliza na regra protocolares que aconselha um sistema com a porta aberta **descartar (drop)** um pacote não solicitado, por outro lado um sistema com a porta fechada deve responder com um RST. Esta conclusão sublinha a classe das ferramentas de reconhecimento e a demanda de arquiteturas de segurança **também de classe**.

No plano da arquitetura de segurança, a análise crítica revelou dois pontos muito importantes para a Engenharia de Computadores.

1. O *ACK scan* mostrou ser uma ferramenta de grande importância para o analista, pois não tem como objetivo identificar o estado da porta, mas sim o **mapeamento de regras de *firewall***, distinguindo sistemas que usam filtragem com estado (*stateful*) em comparação aos que não usam.

2. A vulnerabilidade no ISN em *firmwares* UEFI demonstra que falhas de segurança podem existir na base da arquitetura do sistema, tornando mais fáceis ataques de TCP *Session Hijacking*.

O progresso dos *kernels* em SO, como a manutenção de erros de concorrência como o *Thundering Herd*, e a adaptação do Linux para incluir recursos como os *zero copy* (copia zero) e **tasklets**, é a conclusão dos engenheiros à necessidade de segurança empresarial e escalabilidade em relação aos perigos descobertos. Este

trabalho idealiza que a informação sobre métodos de invasão, como o *port scanning*, é o pilar para a **implementação e construção de defesas robustas**. A Engenharia de Computadores deve operar com uma visão de segurança que proteja todos os níveis, desde a gestão de concorrência no *kernel* ao *firmware*.

Por fim, é obrigatório que um futuro Engenheiro, reconheça os efeitos legais e éticos associados ao uso destas técnicas e ferramentas. Este conhecimento abrangente confere um poder que deve ser utilizado sob uma situação de autorizaçao explícita e **exclusivamente para fins de defesa**, reforçando a relevância do EH como prática académica e profissional.

Figura 5.1: Ilustração de crimes virtuais em várias áreas

Capítulo 6

Contribuições dos autores

Este trabalho foi dividido igualmente, focando-se na divisão entre a base técnica e o desenolvimento da análise cítica e das suas conclusões.

Henrique Gala (HG) foi responsável pela redação do Capítulo 1 (Introdução), do Capítulo 2 (Metodologia), do Capitulo 3 (Resultados), tal como a gestão do repositório git, estrutura técnica do documento e a pesquisa.

Iury Figueredo (IF) foi responsável pela redação do Resumo, do Capítulo 4 (Análise) e do Capitulo 5 (Conclusão), tal como a gestão do repositório git, estrutura técnica do documento e a pesquisa.

Percentagem de contribuição de cada autor: **HG** - 50%, **IF** - 50%

Repositório GitHub: ieci2025-ap-g1

Acrónimos

LECI Licenciatura em Engenharia de Computadores e Informática

HG Henrique Gala

IF Iury Figueredo

CS Cibersegurança

FP Footprinting

RR Reconhecimento de Redes

TCP Protocolo de Transmissão

IP Protocolo de Internet

SEQ Números de sequência

ICMP Protocolo de Mensagens de Controlo da Internet

ACK Números de confirmação

EH Ethical Hacking

IETF Internet Engineering Task Force

ISN Initial Sequence Number

RST Reset

SYN Synchronize

IDS Deteção de intrusão

SO Sistema Operacional

ECI Engenharia de Computadores e Informática

UEFI Unfied Extensible Firmware Development

EH Ethical Hacking

EP Especificações de Protocolo

UDP User Datagram Protocol

PS Port Scanning