Prof. esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp

- **Professor (contratado):**
- **Pós-graduação:** Segurança em redes de computadores - Wyden Facimp
- **Professor (Efetivado):**
- **Graduação:** Todo núcleo de T.I. - Wyden Facimp
- **Gerente de Projetos na Motoca Systems**

Redes sociais:

- **Linkedin:** https://www.linkedin.com/in/thalles-canela/
- **YouTube:** https://www.youtube.com/aXR6CyberSecurity
- **Facebook:** https://www.facebook.com/axr6PenTest
- **Instagram:** https://www.instagram.com/thalles_canela
- **Github:** https://github.com/ThallesCanela
- **Github:** https://github.com/aXR6
- **Twitter:** https://twitter.com/Axr6S

Aula prática utilizando o Wireshark

Uma poderosa ferramenta de análise de tráfego de rede

Introdução:

- Explique brevemente o que é o Wireshark e como ele pode ser usado para analisar o tráfego de rede.
- Discuta a importância de entender o tráfego de rede para identificar problemas de segurança, otimizar o desempenho e solucionar problemas de conectividade.
- Demonstre como iniciar o Wireshark nos computadores dos alunos e familiarize-os com a interface básica.

O Wireshark pode ser usado para várias finalidades na análise de tráfego de rede:

- Solução de problemas de rede: O Wireshark permite identificar problemas de conectividade, atrasos na rede, perda de pacotes e outras questões relacionadas ao desempenho da rede. Ao analisar o tráfego, é possível identificar gargalos, configurações incorretas ou anomalias na comunicação.

O Wireshark pode ser usado para várias finalidades na análise de tráfego de rede:

- Segurança de rede: O Wireshark auxilia na detecção e investigação de atividades maliciosas na rede. Ele permite identificar tentativas de invasão, tráfego suspeito, comportamento anômalo e até mesmo análise forense de incidentes de segurança.

O Wireshark pode ser usado para várias finalidades na análise de tráfego de rede:

- Desenvolvimento e depuração de aplicativos de rede: Ao analisar o tráfego de rede gerado por aplicativos, é possível entender a interação entre os sistemas, diagnosticar problemas de comunicação, verificar a conformidade com protocolos específicos e otimizar o desempenho dos aplicativos.

O Wireshark pode ser usado para várias finalidades na análise de tráfego de rede:

- Análise de protocolos: O Wireshark suporta uma ampla gama de protocolos de rede e permite inspecionar detalhadamente cada camada do modelo OSI. Isso possibilita entender como os protocolos funcionam, detectar erros na implementação, verificar se as negociações de protocolo estão ocorrendo corretamente e solucionar problemas de compatibilidade.

Captura de tráfego:

- Peça aos alunos que iniciem o Wireshark e selecionem a interface de rede correta para capturar o tráfego.
- Explique a diferença entre captura promíscua e não promíscua e incentive-os a experimentar ambas as opções.
- Instrua os alunos a iniciarem a captura de tráfego e a realizarem atividades normais de navegação na Internet ou transferência de arquivos.
- Após alguns minutos, peça-lhes para pararem a captura e salvarem o arquivo de captura para análise posterior.

Captura promíscua:

- Quando a captura promíscua está habilitada, o Wireshark é capaz de capturar todos os pacotes que passam pela interface de rede, independentemente do seu destino.
- Em outras palavras, o modo promíscuo permite que o Wireshark visualize e capture pacotes destinados a outros dispositivos da rede além do próprio computador em que o Wireshark está sendo executado.
- Essa funcionalidade é útil para analisar o tráfego de rede em um ambiente compartilhado, como uma rede local com fio ou uma rede sem fio.

Captura não promíscua:

- No modo não promíscuo, o Wireshark só captura pacotes que são diretamente destinados ao computador em que está sendo executado ou que são transmitidos por ele.
- Isso significa que o Wireshark não será capaz de capturar todos os pacotes da rede, apenas aqueles que estão envolvidos nas comunicações do próprio computador.
- Esse modo é geralmente usado quando é necessário analisar o tráfego local do dispositivo em que o Wireshark está sendo executado.

Análise do tráfego:

- Explique aos alunos como podem usar as diferentes opções de filtragem do Wireshark para focar em pacotes específicos ou tipos de tráfego.
- Instrua-os a abrir o arquivo de captura que salvaram e a explorar os pacotes capturados.
- Peça-lhes que identifiquem diferentes tipos de pacotes (por exemplo, TCP, UDP, HTTP) e analisem os cabeçalhos para obter informações relevantes, como endereços IP, portas e códigos de status.
- Desafie-os a encontrar pacotes suspeitos ou anormais que possam indicar problemas de segurança ou mau funcionamento da rede.

Filtro por endereço IP:

- Os alunos podem filtrar pacotes com base nos endereços IP de origem ou destino.
- Por exemplo, para visualizar apenas os pacotes enviados por um determinado endereço IP, eles podem aplicar o filtro: ip.src == endereço_IP.
- Da mesma forma, para visualizar apenas os pacotes destinados a um endereço IP específico, eles podem usar o filtro: ip.dst == endereço_IP.

Filtro por protocolo:

- Os alunos podem filtrar pacotes com base no protocolo de camada de transporte, como TCP, UDP ou ICMP.
- Por exemplo, para visualizar apenas pacotes TCP, eles podem aplicar o filtro: tcp.
- Para visualizar pacotes UDP, eles podem usar o filtro: udp.
- Para visualizar pacotes ICMP, eles podem usar o filtro: icmp.

Filtro por porta:

- Os alunos podem filtrar pacotes com base em portas específicas.
- Por exemplo, para visualizar apenas pacotes enviados ou recebidos na porta 80 (usada pelo protocolo HTTP), eles podem aplicar o filtro: tcp.port == 80.
- Da mesma forma, para visualizar pacotes na porta 53 (usada pelo protocolo DNS), eles podem usar o filtro: udp.port == 53.

Filtro por padrões de protocolo:

- Os alunos podem usar filtros para encontrar pacotes que correspondam a determinados padrões de protocolo.
- Por exemplo, para visualizar apenas pacotes HTTP que contenham a palavra "openai" no corpo da mensagem, eles podem aplicar o filtro: http contains "openai".
- Para visualizar pacotes DNS relacionados a um domínio específico, eles podem usar o filtro: dns.qry.name == "nome_do_dominio".

Filtrar por endereço IP de origem ou destino:

- ip.src == endereço_IP: Filtra pacotes com endereço IP de origem específico.
- ip.dst == endereço_IP: Filtra pacotes com endereço IP de destino específico.

Filtrar por protocolo de camada de transporte:

- tcp: Filtra pacotes TCP.
- udp: Filtra pacotes UDP.
- icmp: Filtra pacotes ICMP (utilizados por protocolos como ping).

Filtrar por porta:

- tcp.port == número_da_porta: Filtra pacotes TCP com uma porta de origem ou destino específica.
- udp.port == número_da_porta: Filtra pacotes UDP com uma porta de origem ou destino específica.

Filtrar por padrões de protocolo:

- http: Filtra pacotes HTTP.
- dns: Filtra pacotes DNS.
- ftp: Filtra pacotes FTP.
- smtp: Filtra pacotes SMTP (usados para enviar e receber e-mails).

Filtrar por expressões lógicas:

- Os alunos podem combinar diferentes filtros usando operadores lógicos, como and, or e not, para criar expressões mais complexas.
- Por exemplo, tcp and ip.dst == endereço_IP filtra pacotes TCP com um endereço IP de destino específico.

Discussão e conclusão:

- Encerre a atividade com uma discussão em grupo sobre as descobertas dos alunos.
- Peça-lhes que compartilhem suas observações, desafios encontrados e quaisquer perguntas adicionais sobre a análise do tráfego de rede.
- Destaque a importância de uma análise adequada do tráfego de rede para solucionar problemas e manter a segurança.
- Ofereça recursos adicionais, como tutoriais online ou documentação do Wireshark, para que os alunos possam continuar explorando a ferramenta por conta própria.