# Engenharia de Segurança

## Portscan e Honeypots

Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco
kalinka@icmc.usp.br

## Footprint

- Footprint é a técnica de coletar informações sobre sistemas de computadores e sobre as entidades os quais eles pertencem



## Técnicas

- Consultas DNS
- Who is
- Consultas de rede
- Port-scaning
- Consultas SNMP
- Web Spidering
- Traceroutes
- Sniffing



## Utilidade

- Falta de atualizações
- Pouca proteção do sistema
- Senha vulneráveis
- Conta de guest?

## Levantamentos de Ativos e Passivos

- Footprinting passivo
  - Não deixa rastro, não são percebidos pelo alvos (sniffing)
  - Alguns sistemas anunciam a entrad em modo promíscuo (nos logs).
- Footprinting ativo
  - Utilizam conexões com o alvo, logo são rastreáveis.

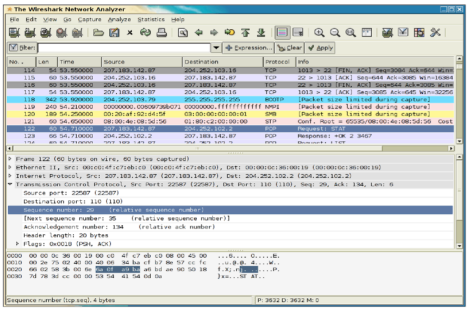
5

## Ferramentas

- Wireshark
- Zenmap
- Scanners

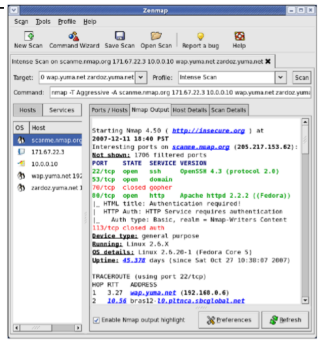
6

## Wireshark



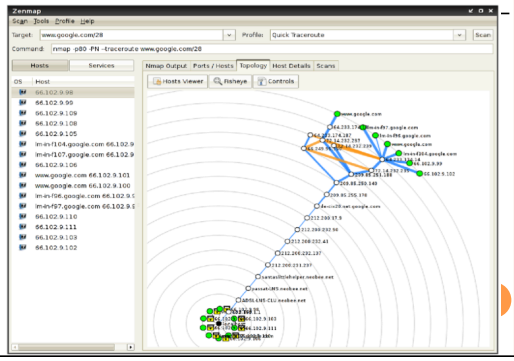
7

## Zenmap



8

## ZENMAP COMO UM SCANER



## PORTSCAN

- O ***Portscan***, que foi criado para que os administradores pudessem visualizar os serviços em sua rede, é como os atacantes geralmente começam a buscar informações em seu servidor.
- Verificam quais os serviços e portas que se encontram abertas e em uso no servidor. Capaz de localizar vulnerabilidades entre máquinas que se encontram na rede.

10

## PORTSCAN

- Bem, analogicamente, podemos comparar o Portscan com um ladrão, que vigia um bairro inteiro a procura de janelas e portas abertas, por onde possa entrar.
- Primeiro precisamos entender que, como manda a RFC do TCP, quando começamos uma negociação para conexão TCP com outro computador, mandamos um pacote com a *flag* SYN ativada, e ele deve então responder um pacote com as *flags* SYN+ACK ativadas, ou seja, quando há resposta a porta se encontra aberta, dependendo da porta sabemos qual o serviço que se encontra ativo nela.

11

- Ha vários tipos de *portscanner*, uns mais efetivos e mais "seguros" para os atacantes, ou seja, que deixam menos rastros:
  - **Método connect() ou "Dumb Scan"** – E um método antigo e simples, utiliza o processo que foi descrito acima. Felizmente, quando se utiliza esse método, o atacante, quando recebe o SYN+ACK, devolve um pacote ACK onde inicia a conexão, porem nesse mesmo momento é possível se identificar a origem. O ataque é facilmente detectado.
  - **Método Half-open** – Vendo o problema do método connect(), surgiu à ideia de se fechar a conexão mandando um RST ao receber o pacote SYN+ACK, uma vez que já era possível somente com ele saber se a porta estava aberta ou não. E como somente no ultimo ACK estava a origem do atacante não seria possível identifica-lo. Esse ataque foi considerado muito eficiente e por um momento indetectável. Porém os sistemas foram se aprimorando e começaram a identificar a origem no SYN inicial da conexão e não mais no ACK final. Tornando esse ataque facilmente identificável.

12

## PORTSCAN

- Começaram então a aparecer métodos obscuros de ataque, que são considerados mais efetivos:
  - **Fin Scan** – Baseados mais uma vez na RFC do TCP, que diz que toda porta fechada deve responder com a flag RST ao receber um pacote com a flag FIN ativada e as portas abertas simplesmente ignoram o pacote com a flag. Sendo assim, ao invés de mandar um SYN, começaram a mandar um FIN que não continha a origem do atacante, aquelas portas que respondessem com um RST estavam fechadas, as que não respondessem nada estavam provavelmente abertas. A desvantagem dessa técnica e que não possibilita a identificação de algum tipo de filtragem por um *firewall*. Além disso, a Microsoft não segue as recomendações da RFC e responde com RST em todas as portas.

13

## PORTSCAN

- **Null Scan** – Bem similar ao Fin *scan*, Assim como quando enviamos a flag FIN, ao enviar a flag NULL (onde desligamos todas as *flags* do pacote) as portas fechadas devem responder com a *flag* RST e as portas abertas simplesmente os ignoram. Porém os mesmos problemas do Fin scan também são aplicados aqui. Outro problema e que se o scan não identificar que a máquina esta "unreachable" ou tem algum tipo de filtragem ele pode entender que a porta que não responda está aberta, retornando uma informação que não é correta.
- **Decoy Scan** – Este é o método mais completo, pois envolve uma junção com uma da técnicas anteriores (menos a *connect*), tornando o ataque mais poderoso e trazendo informações possivelmente mais relevantes. Ele disfarça a origem real do ataque, enviado diversas origens como se fossem vários computadores fazendo o *port scan* ao mesmo tempo, sendo que somente alguns dos pacotes foram enviados pelo ip real do atacante confundindo o computador.

14

## HONEYPOTS E HONEYNETS

- A ideia é entender o conceito de honeypots e honeynets e saber como montá-las e quais informações podem fornecer.

15

## HONEYPOTS

- Em computação, honeypot é uma armadilha montada para refletir, analizar e contra atacar acessos não-autorizados a sistemas de informação. Em geral, consiste de um computador, dados falsos ou de uma pequena parte da rede que parece fazer parte da rede principal do alvo, mas na verdade se trata de uma rede isolada e constantemente monitorada. Um honeypot parece ter informações valiosas ao atacante.

16

## Honeypots – como montar?

- Usando computadores reais;
- Usando máquinas virtuais;
- Usando softwares falsos
  - Fakehttp
  - Fakeproxy
- Usando o deamon honeyd
- Usando o próprio Netkit

17

## Elementos importantes

- Ele deve passar informações ao atacante
- Ele deve ser totalmente isolado
- Ele deve ser continuamente monitorado

18

## Honeypots vs Honenets

- Um honeypot é um conjunto ou serviço falso que passa informações ao receptor, agindo como um único computador.
- Uma honeynet é uma rede de captura de informações sobre o ataque
  - O conceito pode ser expandido para *honeyclouds*

19

## Honeycloud (era *free*, não mais)


20

## COMO FUNCIONA A PRÁTICA

- Utilização da ferramenta honeyd
  - Opera através de uma configuração e um conjunto de scripts
  - O deamon captura os pacotes em modo promíscuo, interpreta e responde com o conteúdo adequado, de acordo com os scripts inteligentes.



# Engenharia de Segurança

**Footprint e Honeypots**

Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco
kalinka@icmc.usp.br