

Relatório de Análise de Incidente: Detecção de Ataque de Brute Force via Wireshark

1. Introdução

O objetivo deste relatório é documentar a análise de um possível ataque de Brute Force detectado em um ambiente de rede utilizando a ferramenta **Wireshark**. O ataque simulado foca no serviço **FTP (porta 21)**, onde tentativas automatizadas de login com um conjunto de credenciais foram observadas. Este cenário visa avaliar a capacidade de detectar tráfego malicioso em tempo real e destacar a importância de configurar alertas para atividades suspeitas em redes corporativas.

Importância do Estudo: Ataques de força bruta são uma das ameaças mais comuns em ambientes corporativos. Embora simples, esses ataques podem ser extremamente eficazes se não forem monitorados adequadamente.

2. Metodologia

2.1 Ferramentas Utilizadas

- **Wireshark (versão 4.4.6)** - Usado para captura e análise detalhada de pacotes em tempo real. Essa ferramenta permite inspecionar tráfego de rede, identificar pacotes específicos e buscar padrões de comportamento que possam indicar um ataque ([Fonte: Wireshark](#)).
- **Suricata (versão 7.0.9)** - Usado para validação cruzada com regras personalizadas de IDS. A ferramenta foi configurada para disparar alertas sempre que um padrão de escaneamento de portas fosse identificado ([Fonte: Suricata](#)).

2.2 Configuração e Parâmetros

- **Wireshark** foi configurado para capturar pacotes em tempo real com filtros específicos para o tráfego FTP:

bash
ftp.request.command == "USER" && "PASS" ftp.response.arg == "Login incorrect."

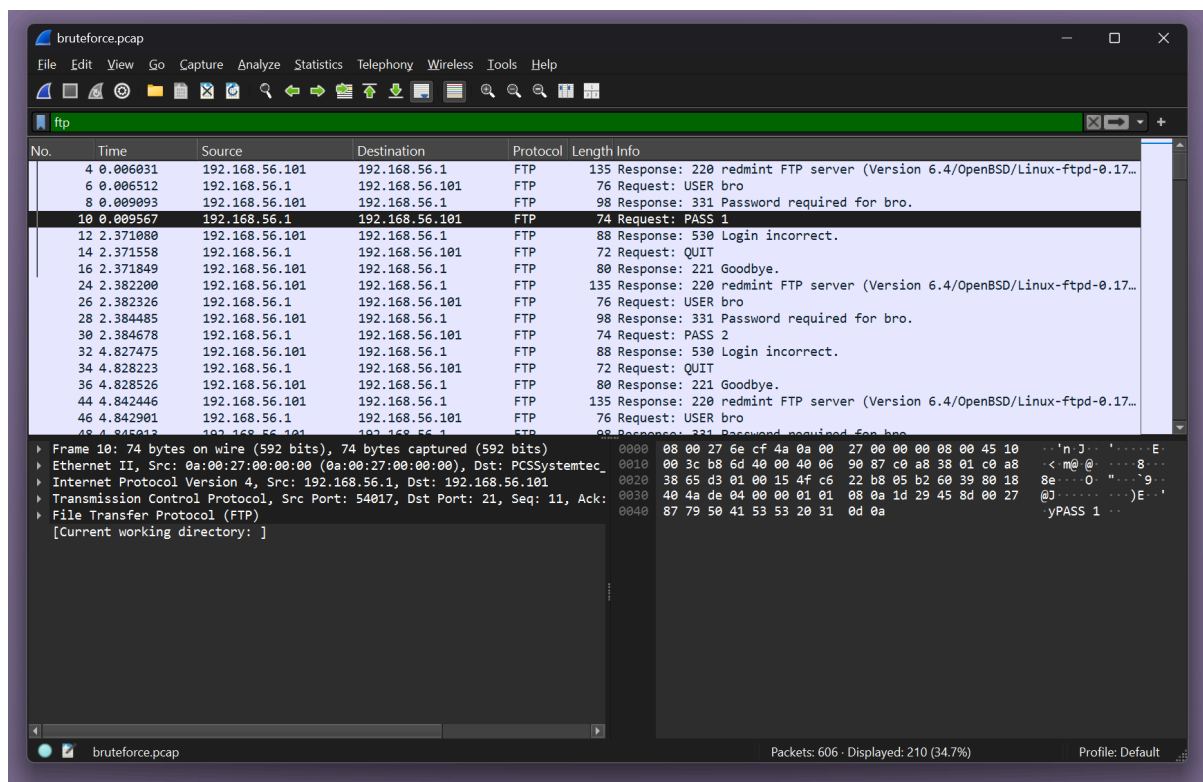
- **Suricata** foi configurado com uma regra personalizada para detectar **SYN Port Scans** e disparar alertas automaticamente:

bash
alert tcp any any -> any 21 (msg:"SYN Port Scan Detected"; flags:S; sid:1000001;)

3. Análise de Dados

3.1 Detecção de Brute Force via Wireshark

Durante a captura de pacotes com **Wireshark**, foi possível observar um comportamento típico de um ataque de força bruta em FTP. O tráfego de rede indicava múltiplas tentativas de login com a mesma sequência de comandos **USER** e **PASS**, onde a resposta do servidor era consistentemente "Login incorrect."

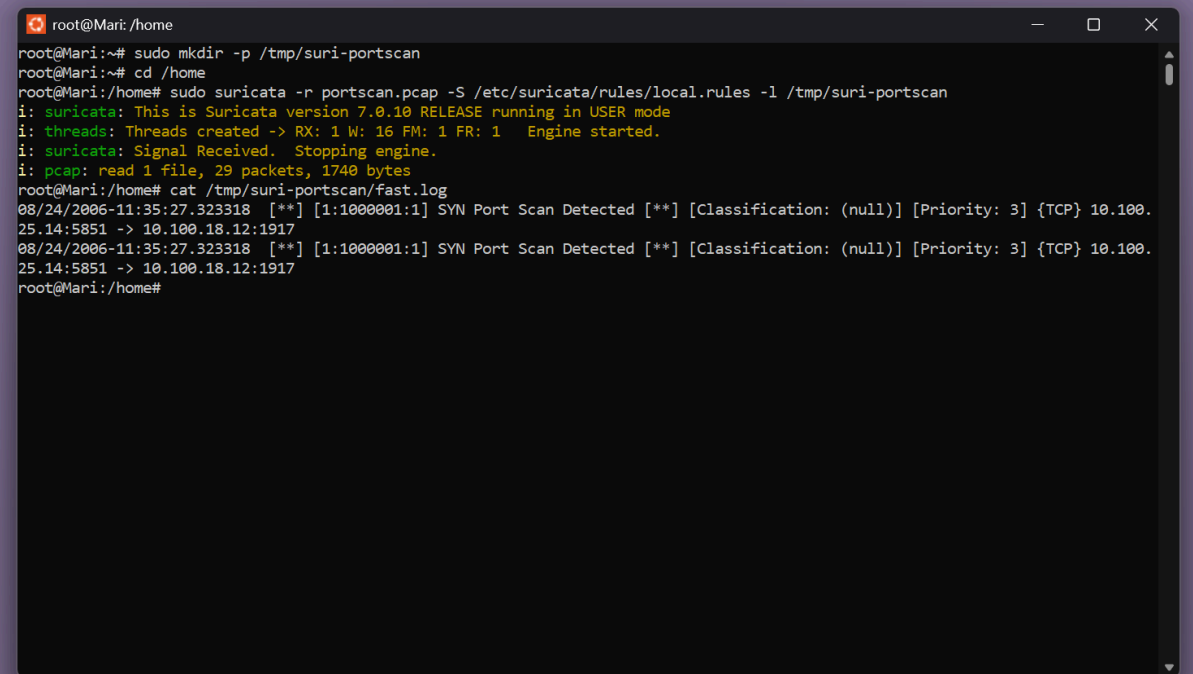


Padrão Identificado:

- Tentativas contínuas de login.
- Respostas de falha consecutivas para cada tentativa.
- Um padrão automatizado de envio de credenciais.

3.2 Análise Automática com Suricata

Com **Suricata** configurado, foi possível validar os resultados da análise manual. O IDS disparou um alerta informando **"SYN Port Scan Detected"**, indicando que múltiplas tentativas de conexão estavam sendo feitas sem um estabelecimento completo de conexão.



```
root@Mari: /home
root@Mari:~# sudo mkdir -p /tmp/suri-portscan
root@Mari:~# cd /home
root@Mari:/home# sudo suricata -r portscan.pcap -S /etc/suricata/rules/local.rules -l /tmp/suri-portscan
i: suricata: This is Suricata version 7.0.10 RELEASE running in USER mode
i: threads: Threads created -> RX: 1 W: 16 FM: 1 FR: 1 Engine started.
i: suricata: Signal Received. Stopping engine.
i: pcap: read 1 file, 29 packets, 1740 bytes
root@Mari:/home# cat /tmp/suri-portscan/fast.log
08/24/2006-11:35:27.323318  [**] [1:1000001:1] SYN Port Scan Detected [**] [Classification: (null)] [Priority: 3] {TCP} 10.100.
25.14:5851 -> 10.100.18.12:1917
08/24/2006-11:35:27.323318  [**] [1:1000001:1] SYN Port Scan Detected [**] [Classification: (null)] [Priority: 3] {TCP} 10.100.
25.14:5851 -> 10.100.18.12:1917
root@Mari:/home#
```

Essa combinação de análise manual com a detecção automática foi fundamental para confirmar o ataque de brute force, além de fornecer uma segunda camada de segurança para validar o comportamento detectado.

4. Resultados e Conclusões

4.1 Conclusão Técnica

A análise de tráfego revelou um padrão claro de ataque de **Brute Force** no serviço FTP. O comportamento observado — falhas repetidas e tentativas automatizadas — é típico desse tipo de ataque, e a detecção foi facilitada por filtros de **Wireshark**.

Ataque Confirmado: Tentativas repetidas de login com falhas consecutivas.

Ação Necessária: Implementação de políticas de bloqueio após um número específico de tentativas de login falhas e adoção de **Autenticação Multifatorial (MFA)** para serviços críticos.

4.2 Recomendações

- **Política de Bloqueio de IP:** Implemente uma regra que bloqueie temporariamente os **IPs** após um número de tentativas de login falhas. Essa medida ajuda a evitar ataques automatizados como o de força bruta.
- **Adoção de MFA:** Implementar a autenticação multifatorial pode mitigar os riscos associados a ataques de força bruta, tornando as credenciais comprometidas inúteis.
- **Monitoramento Contínuo:** A configuração de alertas automáticos para comportamentos anômalos, como tentativas excessivas de login, é essencial para a prevenção de ataques semelhantes.

5. Visão Técnica Geral

Projeto	Técnica de Detecção	Ferramentas	Habilidade SOC
FTP Brute Force	Manual via Wireshark	Wireshark	Análise de autenticação + filtros
Port Scan	IDS + Visual (Wireshark)	Suricata + Wireshark	Escrita de regras + padrão de scan
DNS Beaconing	Manual via Wireshark	Wireshark	Detecção de C2 e persistência furtiva

6. Por que isso importa?

Esses projetos **não são apenas exercícios** de laboratório; eles demonstram habilidades cruciais que um **SOC Analyst** deve ter, tais como:

- **Interpretação de tráfego malicioso:** Capacidade de identificar sinais de intrusão e atividades anômalas em pacotes de rede.
 - **Uso de ferramentas de mercado:** Proficiência no uso de ferramentas como **Wireshark** e **Suricata**, que são amplamente adotadas em ambientes de produção.
 - **Raciocínio SOC voltado para ações práticas:** Identificação de ataques em tempo real, definição de mitigação e tomada de decisões rápidas para proteger a rede.
-

7. Referências

- **Wireshark Documentation:** [Wireshark Docs](#)
 - **Suricata User Guide:** [Suricata Docs](#)
 - **TechSpot - Wireshark 4.4.6 Release:** TechSpot
 - **Suricata 7.0.9 Release Notes:** Suricata.io
-

8. Lições Aprendidas

- **Desafios:** Durante a análise, foi necessário identificar rapidamente os padrões típicos de um ataque de força bruta. O maior desafio foi garantir que o tráfego capturado estivesse corretamente filtrado, evitando false positives.
- **Como as Ferramentas Ajudaram:** **Wireshark** foi essencial para capturar e examinar pacotes, enquanto **Suricata** forneceu uma camada adicional de validação ao disparar alertas automáticos para os eventos detectados.
- **Melhorias Futuras:** Uma análise mais detalhada dos logs de autenticação do servidor FTP poderia melhorar a visibilidade sobre tentativas maliciosas, e a integração de sistemas de alertas automatizados poderia acelerar a resposta.