Network Analysis Process

Network traffic analysis can range from simple observations to complex data capture and analysis.

Simple

Monitoring live traffic flow with tools like Wireshark or tcpdump can be a useful starting point to gain a general understanding of activities on the network. This can be used for simple purposes such as troubleshooting or understanding traffic patterns.

Complex

For more in-depth analyses, especially in security-focused work, NTA can become quite complex. In such analyses, taps can be used to capture data from the network, the captured data can be integrated with a SIEM system, and the captured packets (PCAP files) can be meticulously examined to identify attack signatures or suspicious activities.

Workflow

Generally, all analyses can be summarized into 3 steps:

1. Ingest:

- Data Capture: In this step, network traffic is captured using tools like tcpdump, Wireshark, or Beats in the Elastic Stack.
- Collection: Additional data can be collected from existing logs (firewall logs, application logs, web server logs, etc.). This provides a more comprehensive view of network activity.
- Data Storage and Indexing: Captured traffic can be stored as pcap files in Wireshark or sent to a central system like the Elastic Stack or a SIEM for data storage and indexing.

2. Filter:

- Identifying Targets: Be clear about what you are looking for: a specific type of attack, an anomaly for troubleshooting, performance issues, etc.

- Creating Search Queries: Use your tools to create filters that match the type of traffic you are looking for. For example:
    - Specific protocol (HTTP, DNS, etc.)
    - Traffic related to a source/target IP address
    - Communication using specific ports
- Processing Filtered Data: After applying filters, you work with smaller, focused data.

3. Analyze:

- Searching for Patterns and Anomalies: Depending on your targets, look for notable patterns in the filtered traffic. For example:
    - Significant increases or unexpected decreases in traffic
    - Traffic related to suspicious ports or target protocols
    - Signs of unauthorized access attempts (such as failed login attempts)
- Comparing Baseline: Understanding what 'normal' traffic looks like is important for context. A baseline helps you identify anomalies.
- Investigation and Reporting: When you find an anomaly, conduct a deeper investigation to determine the causes. You may need to create a report detailing the findings and possible actions.
- Taking notes (using paper, markdown, mind-mapping, etc.) and summarizing findings is important at this stage

---

tcpdump basic usage

installation

Most Linux distributions include tcpdump by default. If it's not installed, use your distribution's package manager:

- Debian/Ubuntu:
  sudo apt-get install tcpdump
- RHEL/Fedora:
  sudo dnf install tcpdump

Version Verification

Check your tcpdump installation and version:

```
which tcpdump
```

```
tcpdump --version
```

capturing traffic

You may need root or sudo privileges to run tcpdump.

To list available interfaces:

```
tcpdump -D
```

To start a simple packet capture on a specified interface (replace 'eth0' with the interface you want to capture on):

```
tcpdump -i eth0
```

understanding tcpdump output

Each line in the output represents a captured packet. The format of the packets is as follows:

- Timestamp: The time when the packet was captured.
- Protocol: Type of protocol (TCP, UDP, ICMP, etc.).
- Source and Destination: IP addresses and port numbers.
- Packet Size: Length of the packet in bytes.
- Additional Information: May include flags, sequence numbers, and other protocol-specific information.

For example:

```
13:38:45.232106 IP 192.168.122.59.58774 > 93.184.216.34.http: Flags [.],
ack 1021, win 501, options [nop,nop,TS val 1034903020 ecr 4089339606],
length 0
```

filtering traffic

Tcpdump includes filtering options to focus on specific traffic:

- By Protocol:
  ```
  tcpdump 'tcp'  # Only TCP traffic
  ```
- By Host:
  ```
    tcpdump 'host 192.168.1.100'
  ```
- By Port:
  ```
  tcpdump 'port 80'  # Traffic on port 80 (HTTP)
  ```
- Combinations: Using logical operators like or, and
  ```
  tcpdump 'tcp and port 22' # TCP AND port 22 (SSH)
  ```

reading and writing files

- Saving Captures:
  ```
  tcpdump -i eth0 -w output.pcap
  ```
- Reading from File:
  ```
  tcpdump -r output.pcap
  ```