

Network Traffic Analysis – Phase 2: Anomaly Detection

Phase 2 focuses on the simulation and analysis of clear-text HTTP traffic within a controlled laboratory environment. The objective of this phase is to observe and document communication patterns that, while legitimate, could appear suspicious if detected outside a lab setting.

Lab Setup and Traffic Generation

The traffic was generated in an isolated lab environment composed of Windows and Linux virtual machines. The Windows host acted as the client, while the Linux host provided a simple HTTP service. Repetitive HTTP requests were generated using command-line tools and captured on the Windows host using Wireshark with protocol-based filtering applied.

Simulated HTTP Communication

The captured traffic demonstrates simulated HTTP traffic captured in a controlled lab environment using Wireshark with an HTTP display filter applied. The Windows host initiates multiple HTTP GET requests toward a Linux server, which responds consistently with HTTP 200 OK messages.

The repetitive request-and-response pattern demonstrates clear, unencrypted HTTP communication, making it suitable for observing traffic frequency, request paths, and response behavior. This type of traffic can be used as a reference when identifying anomalies such as excessive requests, unusual endpoints, or unexpected response patterns.

The image shows a Wireshark capture of network traffic on an Ethernet interface. The display filter is set to 'http'. The packet list shows several HTTP GET requests and their corresponding 200 OK responses. The packet details pane for packet 505 is expanded, showing the Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol layers. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
440	8.3088229...			HTTP	216	GET / HTTP/1.1
444	8.3114813...			HTTP	976	HTTP/1.0 200 OK (text/
505	9.2128615...			HTTP	216	GET / HTTP/1.1
509	9.2142920...			HTTP	976	HTTP/1.0 200 OK (text/
533	9.7242874...			HTTP	216	GET / HTTP/1.1
538	9.7259733...			HTTP	976	HTTP/1.0 200 OK (text/
554	10.070152...			HTTP	216	GET / HTTP/1.1
558	10.071739...			HTTP	976	HTTP/1.0 200 OK (text/

Packet 505 details:

- Frame 505: Packet, 216 bytes on wire (1728 bytes captured)
- Ethernet II, Src: PCSystemtec_0d:78:16 (08:00:0d:78:16:08), Dst: 08:00:00:00:00:00
- Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.1
- Transmission Control Protocol, Src Port: 6455, Dst Port: 80
- Hypertext Transfer Protocol

Packet bytes (hex): 0000 08 00 27 e9 ee ce 08 00 27 0d 78 16 0010 00 ca cb 01 40 00 80 06 00 00 c0 a8 0020 01 15 fc 12 1f 90 f3 7d 0e 8d e4 37 0030 00 ff 84 d5 00 00 47 45 54 20 2f 20 0040 2f 31 2e 31 0d 0a 55 73 65 72 2d 41 0050 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 0060 69 6e 64 6f 77 73 20 4e 54 3b 20 57 0070 77 73 20 4e 54 20 31 30 2e 30 3b 20 0080 53 29 20 57 69 6e 64 6f 77 73 50 6f 0090 68 65 6c 6c 2f 35 2e 31 2e 32 36 31 00a0 34 36 32 0d 0a 48 6f 73 74 3a 20 31

Repetitive HTTP Request Pattern

This screenshot shows repetitive HTTP GET requests exchanged between two hosts within a local network. The client repeatedly requests the same resource, and the server consistently responds with HTTP 200 OK messages.

This behavior represents a controlled traffic pattern commonly used to simulate beaconing-like communication patterns or automated polling. While benign in this lab scenario, similar repetitive communication patterns may warrant further investigation in real-world environments when observed at unexpected intervals or destinations

[illegible]

Conclusion

Phase 2 demonstrated how controlled HTTP traffic can be generated and observed within a lab environment. The captured traffic showed clear and repetitive request-and-response patterns, providing a useful example of communication behavior that may appear suspicious when compared against normal baseline traffic.

This phase highlights the importance of understanding traffic patterns before labeling activity as malicious, reinforcing a core SOC skill: distinguishing between legitimate automated behavior and potential indicators of compromise.

Skills demonstrated: Network traffic analysis, HTTP protocol analysis, Wireshark filtering, baseline comparison, SOC investigation methodology.