# Malcolm + Windows EWS + Labshock Integration Guide

This guide enables you to enhance the existing Labshock ICS environment with two powerful additions:

1. **Malcolm (Network Security Monitoring Platform)** – For real-time traffic inspection and protocol parsing, especially ICS-specific traffic like Modbus, DNP3, and EnIP.
2. **Windows EWS Container** – To simulate Windows host traffic (RDP, SMB, HTTP) for enriched detection use cases.

The goal is to help you monitor traffic during your Labshock exercises without altering your instructor-provided lab workflow.

## 📁 Folder Structure

Ensure the following directory layout inside your cloned Labshock project:

labshock/  
├── docker-compose.yml  
├── docker-compose.override.yml # You will create or replace this file  
├── malcolm/  
│ ├── zeek/logs/current/  
│ ├── logstash/config/logstash.yml # you need to create logstash.yml  
│ ├── logstash/pipeline/logstash.conf # you need to create logstash.conf  
│ ├── filebeat/filebeat.yml # you need to create filebeat.yml

## ✍️ Step 1: Create docker-compose.override.yml

Paste the following into docker-compose.override.yml in the root of your Labshock folder:

services:  
 malcolm-zeek:  
 image: ghcr.io/idaholab/malcolm/zeek:25.06.0  
 container\_name: malcolm-zeek  
 command: zeek -i eth1 -i eth2 # can only use one interface in Windows  
 cap\_add:  
 - NET\_ADMIN  
 networks:  
 l3\_network:  
 ipv4\_address: 192.168.3.93  
 l2\_network:  
 ipv4\_address: 192.168.2.93  
 volumes:  
 - ./malcolm/zeek/logs:/opt/zeek/logs  
 restart: unless-stopped  
  
 malcolm-logstash:  
 image: docker.elastic.co/logstash/logstash:8.13.4  
 container\_name: malcolm-logstash  
 volumes:  
 - ./malcolm/logstash/config/logstash.yml:/usr/share/logstash/config/logstash.yml:ro  
 - ./malcolm/logstash/pipeline/logstash.conf:/usr/share/logstash/pipeline/logstash.conf:ro  
 ports:  
 - "5044:5044"  
 networks:  
 - l3\_network  
 restart: unless-stopped  
  
 malcolm-filebeat:  
 image: docker.elastic.co/beats/filebeat:8.13.4  
 container\_name: malcolm-filebeat

command: filebeat -e -strict.perms=false  
 user: root  
 volumes:  
 - ./malcolm/filebeat/filebeat.yml:/usr/share/filebeat/filebeat.yml:ro  
 - ./malcolm/zeek/logs:/zeek/logs:ro  
 depends\_on:  
 - malcolm-logstash  
 networks:  
 - l3\_network  
 restart: unless-stopped  
  
 malcolm-elasticsearch:  
 image: docker.elastic.co/elasticsearch/elasticsearch:8.13.4  
 container\_name: malcolm-elasticsearch  
 environment:  
 - discovery.type=single-node  
 - xpack.security.enabled=false  
 networks:  
 - l3\_network  
 ports:  
 - "9200:9200"  
 restart: unless-stopped  
  
 malcolm-kibana:  
 image: docker.elastic.co/kibana/kibana:8.13.4  
 container\_name: malcolm-kibana  
 environment:  
 - ELASTICSEARCH\_HOSTS=http://malcolm-elasticsearch:9200  
 ports:  
 - "5601:5601"  
 networks:  
 - l3\_network  
 depends\_on:  
 - malcolm-elasticsearch  
 restart: unless-stopped  
  
 ews-win:  
 image: dockur/windows  
 container\_name: windows  
 environment:  
 VERSION: "11"  
 devices:  
 - /dev/kvm  
 - /dev/net/tun  
 cap\_add:  
 - NET\_ADMIN  
 ports:  
 - 8006:8006  
 - 3389:3389/tcp  
 - 3389:3389/udp  
 stop\_grace\_period: 2m  
 networks:  
 l3\_network:  
 ipv4\_address: 192.168.3.95

## 🔧 Step 2: Bring Everything Online

From the root of your labshock/ project:

docker compose down  
  
docker compose pull # Pulls Malcolm and Windows images  
  
docker compose up -d --build

Check that all containers, especially the malcolm-\* and windows, are up:

docker ps

## 🌐 Step 3: Access Kibana for Monitoring

Open a browser and navigate to:

http://localhost:5601

* Add index pattern: zeek-\*
* Explore:
  + conn.log – all connections
  + modbus.log, dnp3.log – ICS protocol activity
  + http.log, rdp.log, dns.log – Windows-related activity

You are now watching real-time Labshock activity with full ICS-aware parsing.

## 🧪 Step 4: Simulate Traffic from CLI

You can generate traffic between Labshock components using simple commands from within containers:

**From Pentest to EWS (RDP scan):**

docker exec pentest nmap -sS 192.168.3.95 -p 3389

**From Pentest to SCADA Web:**

docker exec pentest curl http://192.168.3.20:1881

**From SCADA to PLC (Modbus):**

docker exec scada modpoll -m tcp -t 4 -r 100 -c 5 192.168.2.10

These commands will trigger activity Zeek can parse and send to Malcolm.

## ✅ Final Notes

* You are **not modifying core Labshock functionality**; you’re adding observability.
* Traffic generated during your assigned labs will be passively captured by Zeek.
* Feel free to create dashboards or export logs if you want to analyze behavior after the lab session.

This setup is ideal for getting real-world visibility into ICS/SCADA operations and adversary emulation activity without disrupting the learning objectives.