Michael Aguirre & Francesca Castagnetti
CSC-494: Computer Science Capstone
Professor Japkowicz
Dec. 11, 2025

Project ALCHEMY: Rules of Engagement

I. What are Rules of Engagement?

In the cybersecurity landscape, the rules of engagement are a crucial document that sets the boundaries and guidelines for a penetration test or other security assessment. This agreement dictates what is in-scope, the timeframe of the test, and how to handle sensitive data to prevent accidents or legal issues. On an enterprise level, these agreements sanction engineers to intentionally break or breach a live service often in the hopes of finding exploitable avenues to patch. Because ALCHEMY is on a closed network, the concern of harming our WIFI-networks is eliminated. Yet, we still outline the extent of our simulated MITM to illustrate the attack's specification, purpose, and some context on how it works. In the final iteration of ALCHEMY, ROE will be an important feature that can be edited by a user. This would encourage the educational aspect of our work as it pushes users to consider what kind of attack they want to employ, its timeframe, and the severity of the impact he is trying to create with his simulation.

The ROE will be divided into several categories with small paragraphs to describe the document's specificities.

II. Scope of Operations

All offensive activity is strictly limited to the virtual environment defined by Project ALCHEMY. Under no circumstances may these techniques, tools, or configurations be directed at systems, networks, or services outside the lab.

Only the Target VM is authorized to be "attacked" during MITM scenarios. The sole objective of these attacks is to perform controlled network redirection and traffic/data collection. The Target VM is treated as a simulated client. Target-vm will create traffic over the network by establishing a connection to services-vm. Target-vm then transmits a handful of packets to services-vm to verify connectivity.

The Sensor-vm is strictly passive. Its role is to observe network traffic, generate telemetry, and produce structured logs for later analysis. No direct manipulation, active probing, or offensive activity is permitted from the Sensor VM. Users must not modify system files, services, or configurations on the Sensor VM beyond what is required to keep Suricata and its logging pipeline functioning.

The Attacker VM, which hosts the HTTP(S) proxy (e.g., mitmproxy), is the sole authorized injection point for offensive actions. All traffic interception, modification, and logging must originate from this VM. Students may work with proxy rules, scripts, and configuration files on this machine, but other VMs must not be repurposed as additional attack platforms.

III. Allowed Actions

Within the defined scope, only the following types of actions are permitted. These actions demonstrate the connection between the virtual machines on our network and demonstrate a successful baseline function as described in the documentation.

Limited, controlled traffic modification is allowed exclusively through the proxy. Students may perform benign header or body rewrites—such as injecting harmless strings, manipulating cookies, or altering user-agent headers—to demonstrate how MITM attacks can reshape traffic in transit. These modifications must not be destructive, must not exfiltrate sensitive data, and must remain clearly within the educational context.

Operators may configure and use a transparent or explicitly configured proxy on the Attacker VM to intercept HTTP and HTTPS flows originating from the Target VM and destined for the Services VM (or approved external test endpoints, if enabled).

MITM activities must occur only within clearly defined "attack windows" scheduled as part of the experiment. Outside of these windows, the Target VM should operate under normal, non-intercepted conditions whenever possible. Timeboxing ensures that baseline and attack traffic are clearly distinguishable and that any misconfigurations are easier to detect and roll back.

IV. Prohibited Actions

To protect the integrity of the lab environment and uphold ethical standards, the following actions are explicitly forbidden:

Participants may not deploy payloads intended to delete data, corrupt files, deface content, or crash services on any VM. Denial-of-service attacks, filesystem corruption, or intentional destabilization of the Services or Target VM are out of scope.

Students must not install backdoors, create unauthorized user accounts, or configure persistence mechanisms on any VM. The environment is designed to be reset via snapshots; leaving hidden persistent access points directly conflicts with the lab's safety and reproducibility goals.

All attack traffic must remain strictly confined to the host-only lab network. Using MITM tools, scripts, or configurations to probe or attack the host machine, the broader campus network, home networks, cloud services, or the public internet is strictly prohibited. Any such behavior is a serious violation and may result in disciplinary action.

Attempts to gain root or system-level privileges on the Target or Sensor VM are not allowed. The objective of this project is network-level interception and detection, not full host compromise. Users must operate within their assigned privileges and must not exploit vulnerabilities on the lab machines for privilege escalation.

V. Safety, Logging, Responsibility

Project ALCHEMY emphasizes safe experimentation and transparent, repeatable workflows. The following practices are required:

If the stability of the Target or Sensor VM is affected—whether by misconfiguration, failed experiments, or unexpected side effects—operators must restore the VM from a known-good snapshot. Snapshots are the primary safety mechanism to ensure that the environment can be returned to a clean state quickly and reliably after each experiment.

All commands, configuration changes, and proxy scripts used during an experiment must be documented. This includes noting which rules were active, which traffic was intercepted or modified, and any adjustments made to Suricata or logging settings. The goal is that another student or instructor could reproduce the same scenario based solely on the written record.