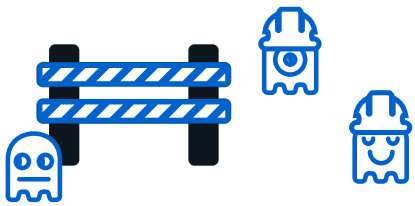
ADVERSARY ENGAGEMENT OPERATION BUILD GUIDANCE

In order to conduct adversary engagement operations, it is necessary to build two distinct environments: a Collection System (CS) and an Engagement Environment (EE). The Collection System is the set of systems used to gather artifacts and other data from an operation to monitor the engagement. The Engagement Environment is the set of carefully tailored, highly instrumented systems designed on an engagement-by-engagement basis as the backdrop to the engagement narrative. It is the actual environment that the adversary will operate in. This document details the best practices used by the MITRE Engage™ Team to setup each system.

COLLECTION SYSTEM (CS) BUILD GUIDANCE

The CS must remain functional during the operation. If anything happens to the CS, then our ability to see, and thus understand, the adversary's actions become limited at best and impossible at worst. Without the CS, we lose the ability to maintain operational safety, as we cannot identify when an engagement crosses a gating criterion. Therefore, it is essential to ensure the availability and integrity of the CS to operate within the Rules of Engagement. We prevent the adversary from controlling the CS by keeping it as separate as possible from the engagement environment. This separation also maintains the realism of the environment, as the adversary is unaware that they are being monitored. The simplest implementation of a CS is a passive network tap feeding a packet capture system. Since the adversary must interact with the system entirely over network communications, we can monitor all egress points. However, things like encryption can complicate our ability to inspect the data. In this case, the engagement operator may need to set up some form of Man-in-the-Middle processing to view data. This data manipulation may happen at the egress point, or it may involve manipulating system functions and applications within the

engagement environment. We will also want to have in-system collection (e.g., eventlog or syslog) and egress. A failsafe can be used to reduce the impact of unexpected errors (e.g., CS hardware failure). One possible implementation is a signal sent at regular intervals (e.g., 1 minute) from the CS to a device that can shut off the Internet connection if the beacon is not received. This automated kill switch is also useful to enable automatic or manual shutoffs if a gating criterion is crossed.



> Our ability to achieve our operational goals is dependent on our ability to safely and effectively monitor and control the engagement environment.
>
> Maretta Morovitz, MITRE Engage Lead

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD®

The hardware necessary for this system is dependent on the capture situation. For example, if using a network tap and it is 100Mbit copper, then the capture interface needs to support that speed and media type. Likewise, if the management network connection is 1Gbit fiber, then the management interface must match.

To prevent normal operations, such as log and analysis output, from interfering with the capture process, the capture data should be kept on a separate partition from the rest of the system. If using a Linux-based system, we recommend that this data partition should be formatted XFS. In our testing, the XFS file system format is the most responsive and carries the least overhead and impact on packet capture to disk.

ENGAGEMENT ENVIRONMENT (EE) BUILD GUIDANCE

The Engagement Environment may be based completely isolated from any real networks and systems, or fully integrated into a production environment. The operational objectives will drive this decision. For example, if you are attempting to identify an insider threat, an isolated environment may not be useful. However, if you are looking to gather new TTPs by running a malware sample, an isolated environment may be essential to maintain operational safety. Even when integrating deceptive assets into a production network, it may be necessary to obfuscate or reconfigure some elements of your golden image, or template image, deployed on these deceptive assets. These changes, even subtle ones, can protect your real infrastructure. For example, let's say you are deploying a decoy system with an intentional vulnerability, that you expect will be compromised. If you deploy your real golden image, you give your adversaries a window into your real system strengths and weaknesses. Additionally, when configuring your EE, it is important to understand your chosen Target Adversary's expectations. What does your adversary already know about the environment they expect to land in? What warning sides will make them back off? What elements will make them feel reassured that they are in the anticipated environment? The answer to questions such as these should drive the setup of your EE. Finally, the EE should be seeded with the appropriate deceptive artifacts, including Pocket Litter and Lures. These elements may include files, system artifacts, browser history, etc. They are intended to tell the engagement narrative and drive the engagement towards strategic outcomes. A full discussion of how to create these elements and where to place them is outside the scope of this document.

It is also important to have good operational security (OPSEC) such that one compromised operation does not mean that all operations are compromised. Therefore, we must be sure to vary key aspects of the simulation to prevent operations from being linked together. To do these, the team must be able to keep track of system configurations across operations. Which details may link operations, often depends on the sophistication and TTPs of a given adversary. However, some common observable aspects include the following:

- Build date and time
- MAC address
- Usernames
- Network name
- Passwords
- Computer name

When building an EE, it is equally important to consider the story you do not want to tell, as much as the story you do want to tell. Often the story you do not want to tell is directly related to weakness in your tooling or environment or dependencies based on operational resources. For example, if your EE is located in a shared space, it may be difficult or impossible to control sounds in the environment. In this case, it may be prudent to physically remove the microphone rather than risk an adversary deploying a recording capability. Then the operator might include an email in the victim's sent folder to the IT department complaining that their microphone is not working. Other elements that may need to be disabled include the camera, Bluetooth, and wireless capabilities.

Building and configuring the engagement environment also requires understanding the initial malware's dependencies to make it execute successfully. Such requirements may include things like specific versions of Adobe Reader or Microsoft Office, Active Directory, the lack of specific security patches, the lack of certain security features (e.g., ASLR), administrator privileges, or installation of the East Asian language pack. Some of this can be configured prior to any operation, whereas other times it will require rapid analysis of the infection vector.

For information about MITRE Engage,™ contact engage@mitre.org, visit us at engage.mitre.org, or connect with us on LinkedIn @MITRE Engage

MITRE's mission-driven teams are dedicated to solving problems for a safer world. Through our public-private partnerships and federally funded R&D centers, we work across government and in partnership with industry to tackle challenges to the safety, stability, and well-being of our nation.

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD®