TABLE 1: Information needed to judge Validity of threats- GitHub scenario (For the column Actual, 1= actual threat, 0= fabricated threat)

ID	Actual	Justification	Textual hint	Additional hint (DFD)
1	1	Using simple alert-style email notifications, attackers are able to steal credentials to gain access to development code, intellectual property, and project details	Assumptions confirms a successful social engineering attack	Absence of a multi-factor authentication process
2	1	Git's configuration files are plain-text, when leaked an attacker can view the personal information of the repository owner, leading to information disclosure attack. In addition, the attacker can alter the default bahaviour set by the owner by manually editing these values	The YAML configuration files are automatically executed	Presence of configuration file data store
3	1	An attacker can exploit vulnerabilities (e.g., lack of authentication) in the git protocol to send multiple requests that overload or crash the server causing a DoS attack	Assumption confirms attacker's ability to submit malicious code	Absence of authentication process
4	1	An attacker can get server and .../path/to/repo from git remote (only if the remote repository is of the ssh type). This permits the execution of server-side Git commands implementing the pull/push functionality	Assumptions confirm that server has pulled and is running malicious code	Presence of run new push data flow
5	1	With root access, the attacker is able to perform any actions on GitHub, such as creating a pull/push request in a repository	Assumption that the attacker can submit code remotely	Presence of a commit and push data flow
6	0	It is possible for an attacker to submit a fake push request to other remote code repositories if they have access to the necessary credentials or exploit a vulnerability in the system. This could potentially cause a Denial of Service attack. But the assumptions clearly states that permission management is properly implemented	Confirmation of properly implemented permission management	Presence of a protocol process which executes push commands from legitimate users
7	0	It is possible for an attacker to browse the project yml file, especially if the repository is public. However, an attacker with access to the yml files can only view the workflow of the repository	Confirmation that an attacker can browse the projects yml files	Presence of trust boundary around the GitHub engine
8	0	To perform actions on GitHub, such as creating a pull request in a repository, a person must have sufficient access. The only way to gain push access is when the admin/owner of the repository assigns these permissions to you. Hence, an attacker can't gain access if they are not directly assigned to them	Confirmation that attacker has push access only, which limits their functionality	Presence of a trust boundary between • user and local Git server • local Git server and remote server
9	0	An unauthenticated and non-privileged attacker can submit custom code into a remote repository, if vulnerabilities exists. However, changes made to code in a repository are often reviewed before being approved and merged to the master node. The approval process reduces the risk of unauthorized changes	Assumption that the attacker can reach the remote repository (e.g. through the Internet)	Presence of a trust boundary around remote Git server

TABLE 1: Information needed to judge Validity of threats- GitHub scenario (For the column Actual, 1= actual threat, 0= fabricated threat)

ID	Actual	Justification	Textual hint	Additional hint (DFD)
10	0	GitHub uses HTTPS, which encrypts traffic between the server and client, which ultimately prevents unauthorized access. The HTTPS protocol provides an additional layer of security for users	Confirmation that GitHub uses HTTPS while the attacker managed to compromise HTTP	Presence of trust boundaries around local Git server, Remote Git server and GitHub actions

TABLE 2: Information needed to judge Validity of threats- Kubernetes scenario (Actual column, 1= actual threat, 0= fabricated threat)

ID	Actual	Justification	Textual hint	Additional hint (DFD)
1	1	In cases where the Kubernetes cluster is deployed in a public cloud (e.g., AKS in Azure, GKE in GCP, or EKS in AWS), compromised cloud credentials can lead to cluster takeover. Attackers with access to the cloud account credentials can access the cluster's management layer	Confirmation that attacker can remotely interact with the server	Presence of data flow saving pod configurations in the etcD
2	1	This threat is possible by gaining access to a cluster using leaked credential configuration files, or by interacting with an unauthenticated API server of a cluster	Assumptions confirms a successful social engineering attack	Absence of a multi-factor authentication process
3	1	A container without CPU and memory resource limits can use up all the worker node's resources (e.g., cryptocurrency mining scripts). This will likely cause crashing of other containers running on the same node, and in the worst case, the node itself	Confirmation that the attacker obtained a remote shell on a pod	Presence of pods and worker-nodes in the same trust boundary
4	1	Running a compromised image in a cluster can compromise the cluster. Attackers who get access to a private registry can plant their own compromised images. A user can then pull the latter. In addition, users often use untrusted images from public registries (such as Docker Hub) that may be malicious (supply-chain attacks)	• Scenario description detailing how a container image can be retrieved from a a registry • Confirmation that attacker has permissions to upload or modify images in the registry used by the Kubernetes cluster	Presence of image registry and image data store in the same trust boundary

TABLE 2: Information needed to judge Validity of threats- Kubernetes scenario (Actual column, 1= actual threat, 0= fabricated threat)

ID	Actual	Justification	Textual hint	Additional hint (DFD)
5	1	Either by exploiting a vulnerability and getting a reverse shell, or by laterally moving to a privileged container, if an attacker gets a reverse shell on a privileged container, that is equivalent to having access to the container's worker node, which means cluster take-over	• Scenario description of multiple containers being run inside a single pod • Confirmation that attacker has access to privileged container	Presence of a trust boundary around the pod
6	0	The threat is fake because the network policies are implemented correctly, so no network traffic is allowed between the two namespaces	Confirmation that network policies are implemented correctly to segment the namespaces	Absence of container processes in different namespaces
7	0	An attacker with access to image A can only reverse image A and Dockerfile A, but not other images or Dockerfiles	Textual description of how a container image is retrieved from a registry	Presence of data flow to retrieve images where image name and tag are required
8	0	If an attacker compromises a running container, they can run any type of software/tool/process, independently from what is specified in the Dockerfile	Textual description of what is contained in a container (i.e. source code, libraries, and other dependencies)	Absence of security monitoring process
9	0	This threat may have been true if anonymous access was enabled in the API server. However, this is not specified, making it invalid because the attacker does not have enough privileges to upload custom configurations	No confirmation on whether anonymous access is enabled in the API server	Presence of trust boundary between user and master node containing the API server
10	0	A network bridge is a Layer 2 network divide. If the cluster is using a Layer 3 network plugin, the network bridge is not present, thus it can not be exploited	Scenario description that Layer 2's virtual devices such as a network bridge are not usable in Layer 3	Absence of processes for several pods

TABLE 3: Technical knowledge needed to judge Validity of threats- GitHub scenario (Actual column, 1= actual threat, 0= fabricated threat)

ID	Actual	Justification	Technical knowledge
1	1	Using simple alert-style email notifications, attackers are able to steal credentials to gain access to development code, intellectual property, and project details	Knowledge on how social engineering attacks are carried out and the success factors
2	1	Git's configuration files are plain-text, when leaked an attacker can view the personal information of the repository owner, leading to information disclosure attack. In addition, the attacker can alter the default bahaviour set by the owner by manually editing these values	Knowledge that Git configuration files are plain text and contains sensitive information such as usernames and email addresses

TABLE 3: Technical knowledge needed to judge Validity of threats- GitHub scenario (Actual column, 1= actual threat, 0= fabricated threat)

ID	Actual	Justification	Technical knowledge
3	1	An attacker can exploit vulnerabilities (e.g., lack of authentication) in the git protocol to send multiple requests that overload or crash the server causing a DoS attack	Knowledge on • Exploiting authentication vulnerabilities (e.g., brute force attack on weak passwords, social engineering attacks) • How to perform a denial of service attack and its success factors
4	1	An attacker can get server and .../path/to/repo from git remote (only if the remote repository is of the ssh type). This permits the execution of server-side Git commands implementing the pull/push functionality	Knowledge that with the correct SSH keys the user can be authentication to the Git shell without requiring additional log in information)
5	1	With root access, the attacker is able to perform any actions on GitHub, such as creating a pull/push request in a repository	knowledge that only root access allows the execution of core git actions
6	0	It is possible for an attacker to submit a fake push request to other remote code repositories if they have access to the necessary credentials or exploit a vulnerability in the system. This could potentially cause a Denial of Service attack. But the assumptions clearly states that permission management is properly implemented	Knowledge on • Git's permissions management • Success factors of a DoS attack
7	0	It is possible for an attacker to browse the project yml file, especially if the repository is public. However, an attacker with access to the yml files can only view the workflow of the repository	Knowledge that a YAML file contains human-readable data serialization used to define workflows and configuration. They are not executable.
8	0	To perform actions on GitHub, such as creating a pull request in a repository, a person must have sufficient access. The only way to gain push access is when the admin/owner of the repository assigns these permissions to you. Hence, an attacker can't gain access if they are not directly assigned to them	Knowledge of Git's permissions management which controls a user's access to a repository through the assigned roles
9	0	An unauthenticated and non-privileged attacker can submit custom code into a remote repository, if vulnerabilities exists. However, changes made to code in a repository are often reviewed before being approved and merged to the master node. The approval process reduces the risk of unauthorized changes	Knowledge of Git's version control alongside pull request approval process
10	0	GitHub uses HTTPS, which encrypts traffic between the server and client, which ultimately prevents unauthorized access. The HTTPS protocol provides an additional layer of security for users	Knowledge that HTTPS offers an additional security layer, which encrypts communication

TABLE 4: Technical knowledge needed to judge Validity of threats- Kubernetes scenario (Actual column, 1= actual threat, 0= fabricated threat)

ID	Actual	Justification	Technical knowledge
1	1	In cases where the Kubernetes cluster is deployed in a public cloud (e.g., AKS in Azure, GKE in GCP, or EKS in AWS), compromised cloud credentials can lead to cluster takeover. Attackers with access to the cloud account credentials can access the cluster's management layer	Knowledge that exploiting authentication vulnerabilities and gaining valid credentials can lead to a cluster take-over

TABLE 4: Technical knowledge needed to judge Validity of threats- Kubernetes scenario (Actual column, 1= actual threat, 0= fabricated threat)

ID	Actual	Justification	Technical knowledge
2	1	This threat is possible by gaining access to a cluster using leaked credential configuration files, or by interacting with an unauthenticated API server of a cluster	Knowledge on how social engineering attacks are carried out and the success factors
3	1	A container without CPU and memory resource limits can use up all the worker node's resources (e.g., cryptocurrency mining scripts). This will likely cause crashing of other containers running on the same node, and in the worst case, the node itself	Knowledge that a container has no resource constraints, by default. However, mitigations such as setting runtime configuration
4	1	Running a compromised image in a cluster can compromise the cluster. Attackers who get access to a private registry can plant their own compromised images. A user can then pull the latter. In addition, users often use untrusted images from public registries (such as Docker Hub) that may be malicious (supply-chain attacks)	Knowledge that a cluster's access can be set as public or private
5	1	Either by exploiting a vulnerability and getting a reverse shell, or by laterally moving to a privileged container, if an attacker gets a reverse shell on a privileged container, that is equivalent to having access to the container's worker node, which means cluster take-over	Knowledge of lateral movement between containers
6	0	The threat is fake because the network policies are implemented correctly, so no network traffic is allowed between the two namespaces	Knowledge on the benefits of using namespaces; isolate resources, define access controls, e.t.c
7	0	An attacker with access to image A can only reverse image A and Dockerfile A, but not other images or Dockerfiles	Knowledge that one can only reverse engineer the images one has access to. Images are typically stored in separate filesystems and registries, and they are isolated from one another at runtime
8	0	If an attacker compromises a running container, they can run any type of software/tool/process, independently from what is specified in the Dockerfile	Knowledge that with the right permissions (root access) to the container, an attacker can download and run additional tools, install software, or modify existing files
9	0	This threat may have been true if anonymous access was enabled in the API server. However, this is not specified, making it invalid because the attacker does not have enough privileges to upload custom configurations	Knowledge that unauthenticated users can send API requests to the Kubernetes API server only if anonymous access is enabled
10	0	A network bridge is a Layer 2 network divide. If the cluster is using a Layer 3 network plugin, the network bridge is not present, thus it can not be exploited	Knowledge of the differences between Layer 2 and Layer 3 of the OSI model