Business Proposal

OSTIF

Kubernetes Non-Core Audit

December 12, 2024

Number 84/2024 v1.1

1. Document Details

The document aims to highlight our business and technical proposal based on the needs expressed by **Open Source Technology Improvement Fund** (OSTIF – Customer) for a security audit by **Shielder S.p.A.** (Shielder – Supplier).

Specifically, this proposal concerns a Security Engagement of various **Kubernetes Non-Core** components and subprojects.

In this document OSTIF and Shielder may also be jointly referred to as the "Parties" and individually referred to as the "Party".

Classification	Internal use
Last Review	December 12, 2024
Author(s)	Claudio Mancini, Abdel Adim Oisfi

1.1. *Version*

Identifier	Date	Author	Note
v1.0	December 04, 2024	Claudio Mancini	First version
v1.1	December 12, 2024	Abdel Adim Oisfi	Redacted version

2. Summary

3. Company Presentation

Shielder is an **IT security company** with a team of specialized consultants in **Cyber**, **Hardware** and **Physical** Security. The company, founded in 2014, has a wide portfolio of customers, which can prove its expertise and skills. The daily driver of the team is to provide high-quality assessments and deliverables, which is possible thanks to big investments in **research** and **training**.

3.1. *Red Team*

The Red Team is composed by seven members who have worked over the years achieving extensive experience, in the following fields:

- Web Application Penetration Testing
- Mobile Application Penetration Testing
- Embedded System Penetration Testing
- Red Teaming
- Physical Penetration Testing
- Reverse Engineering
- Cryptography and Cryptanalysis
- Security Code Review
- Cloud Configuration Review
- Social Engineering
- Smart Contracts Penetration Testing

4. Project Description

OSTIF requested a proposal for a security engagement which focuses on Kubernetes (k8s) features and subprojects that have not been subject to significant formal security review.

Kubernetes is an open-source container orchestration platform initially developed by Google in 2014, inspired by their internal system Borg. It automates deployment, scaling, and management of containerized applications. Kubernetes has become the de facto standard for modern cloud-native architectures due to its flexibility and ecosystem support.

However, the rapid growth of Kubernetes has led to ecosystem fragmentation. Its core components are supplemented by a diverse set of subprojects, each addressing specific use cases like networking, storage, and observability. While this modularity fosters innovation, it also introduces complexity and potential security risks, especially as organizations integrate varying components and configurations.

To ensure the community benefits from this engagement, OSTIF conducted a thorough pre-engagement assessment to prioritize relevant components. This effort resulted in a curated list of high-priority and low-priority features and subprojects with active community support. The high-priority items are expected to undergo a comprehensive security review, while the low-priority ones could be addressed on a best-effort basis, contingent on the available time and budget.

The high-priority list follows with some notes by Shielder:

- **Cluster API** – a set of APIs and tools to standardize the cluster management. It is used by all the main cluster bootstrapping tools. The main threat is related to the presence of insecure defaults which could make standard deployments vulnerable to specific attacks.

- **Pod Security Admission** – a replacement for the Pod Security Policy. It is used to setup a policy and validate it during the Pod creation. The policies are meant to allow/dis-allow the creation of privileged Pods which could enable privilege escalation scenarios. The main threat is related to policy bypasses, which could allow an entity with the create-pod permission to arbitrarily escalate their privileges within the cluster.

- **Multiple Windows Components** & **Windows Privileged Containers and Host Networking Mode** – 5 years after the initial release Windows reached a production-level support in k8s. Since then, there has been a steady development of components aimed to port all the Kubernetes features to Windows deployments. While the support is quite mature from a functional perspective, the Windows-related components are fairly young and thus they were not subject to extensive security auditing. To correctly drive the effort and make sure all the sensitive areas are covered, the assessment of the Windows components will start with a thorough threat modeling exercise.

- **Konnectivity Client** – a component aimed to setup a TCP level proxy to allow cluster administrators to reach isolated clusters via their control plane. While the scope is focused on the client, also the server component will also be audited, in order to have a full understanding of the communication protocol and the potential risks. The main threats are related to person-in-the-middle attacks and to unsafe defaults which might overexpose the cluster from a networking perspective.

- **Shared Public Cloud Library** – a library which could be used by cloud providers to implement their services (storage, networking, IAM, etc.) directly inside Kubernetes. As the library is used to implement a lot of sensitive components, vulnerabilities in the library implementation might lead to disastrous consequences. Moreover, the library documentation will be reviewed to make sure that any security catch in its usage is well-documented.

- **Credential Provider Plugin** – a plugin which enables Kubernetes users to integrate external credentials providers in an easy way to fetch registry credentials at runtime. The plugin is implemented by invoking subprocesses that wrap the executables associated to the provider, handling the transformation of input and output data to match the interfaces of respectively the provider and the k8s ecosystem. This approach might be subject to different threats ranging from command injections and argument injections to parsing issues.

- **Image Builder** – a tool which is used to create Kubernetes VM images in a wide range of formats, ready to import in different cloud providers and hypervisors. The image building process supports custom stages to be configured, on top of the default ones. The main threat is related to the presence of any insecure default which could make all the VMs built via the image builder vulnerable (i.e. default credentials, overprivileged services, etc.).

- **CEL Admission Control / ValidatingAdmissionPolicy** & **CRD Validation Expression Language** – admission policies are used to specify custom acceptance or rejection rules of resources within the cluster. These policies could be simple (e.g. an attribute is set to a specific value) or more complex. In the latter case Kubernetes provides two strategies, the webhooks or the common expression language (CEL). While webhooks are extremely flexible, as they are pieces of Golang code developed ad-hoc for the check, they are also slow and complex. For this reason, k8s is encouraging cluster administrators to move most of the logic to CEL-based rules. CEL is an expression evaluation language developed by Google which is linear time, mutation free, and not Turing-complete. The main threat is caused by potential vulnerabilities in the CEL engine which might lead to bypasses of the expressions and denial of service attacks.

- **Ephemeral Containers** – a type of container which runs in an existing Pod and is supposed to be used for user-initiated tasks, such as troubleshooting. The ephemeral Pods should not leave artifacts in the Pod and should not change the security posture of the Pod by default, unless the cluster administrator chooses to run specific commands to do so. The main threat is related to potential insecure design strategies in how ephemeral containers are setup and which could allow a compromised Pod to elevate its privileges (i.e. becoming a privileged Pod while the ephemeral container is running).

- **Cgroups v2** – k8s migrated from cgroups v1 to v2 keeping the same end-user interface. To do that, the cgroups v2 interface converts some configurations from the v1 ranges/semantic to the v2 ones. This might lead to inconsistencies; therefore, the analysis will focus on making sure that the conversion operations are safe.

- **Seccomp by Default** – seccomp is a security feature of the Linux Kernel that can be used to restrict the kind of syscalls that a process can access. Kubernetes 1.19 promoted seccomp to GA, granting cluster administrators an additional security in-depth tool. Kubernetes comes with a default seccomp ruleset which can be extended by end-users. The audit will focus on the default ruleset to make sure it provides the documented security benefits.

- **Node Topology Manager** – a kubelet component that aims to coordinate the set of components that are responsible for performance optimizations. The main role of the topology manager is to simplify the hardware optimization configurations as they happen in a wide variety of components. The relevant threats of manager are not obvious and will require a dedicated thorough threat modeling exercise.

- **Support 3rd Party Device Monitoring Plugins** – Kubernetes enables cluster administrators to add custom devices (i.e. GPUs) in their clusters and assign them to Pods to perform specific operations requiring dedicated hardware. To better manage custom devices, k8s allows hardware vendors to create plugins which could configure and monitor them. This is done through a vendor-created privileged DaemonSet which mounts a unix socket from the Kubernetes host to which it communicates, enabling a bi-directional communication channel. The main threats are related to potential authorization issues which could grant a monitoring plugin the capability to escalate its privileges inside the cluster and/or to tamper other custom devices in the same cluster.

- **AppArmor Support** – AppArmor is a Linux Kernel security module which allows to restrict processes capabilities (e.g. opening a socket) with a granual approach using mandatory access control (MAC) strategy. Kubernetes supports AppArmor, allowing cluster administrators to assign AppArmor profiles to containers, limiting the impact of vulnerabilities. The audit will focus on the AppArmor implementation and to make sure it is consistent with the standard one and that it has no k8s-specific bypasses.

- **Local Ephemeral Storage Capacity Isolation** – a type of storage which could be used for ephemeral purposes and which lifetime matches the assigned Pod's one. The main threat is related to the data hygiene and to any potential insecure design strategy which could lead a Pod to access data written in previous executions of the Pod.

- **Cleaning Up IPTables Chain Ownership** – for historical reasons, Kubernetes had multiple components creating IPTables rules, resulting in rulesets containing duplicates and inconsistent strategies, thus hard to maintain and audit. Recently, the rules creation strategy changed, removing the management from kubelet and updating kube-proxy accordingly. The audit will focus on the applied changes to make sure that no issues have been introduced in the migration, making any sensitive component reachable in unintended ways.

The low-priority list contains ~80 entries, some of them being directly connected to high-priority features, while others are stand-alone contributions. Shielder will follow the approach below for each low-priority entry:

- An initial session to understand the feature and if it's bound to a high-priority entry:
    - If it is bound to a high-priority entry it will be included in the full-audit along with the high-priority component.
    - If it is not bound to a high-priority entry:
        - A lightweight threat modeling and risk assessment will be performed.
        - If any relevant threat/risk will be detected, the code will be audited to identify potential vulnerabilities.

While the main goal of the security engagement is to find vulnerabilities in the components in scope, it is also crucial to provide Kubernetes with some actionable improvements which could raise its security posture in the long term, preventing regressions. Moreover, the whole process is expected to be fully transparent, following the open-source philosophy. Based on this, Shielder will not only look for vulnerabilities but also for undocumented behaviors which could have security impacts, help the community to create unit-tests to prevent regressions, develop SAST rules to detect the vulnerabilities with the highest density, and document the main challenges in setting up debug-able environments to lower the entry barriers for other security researchers. Refer to chapter **4.2 - Deliverables** for more details about the expected deliverables.

4.1. Team

The security engagement will be multi-domain, covering a wide-range of technologies and requiring a deep understanding of complex threats and risks. To address this complexity, Shielder will assign the project to a heterogeneous team with vertical experience on different security aspects relevant for the project.

Precisely, the team will have experts in the following main fields:

- Cloud Security
- Reverse Engineering
- Windows Internals
- API Security

As a proof of past experience in the field, Shielder recently performed a security audit facilitated by OSTIF on Karmada (Kubernetes Armada) – a Kubernetes management system that enables transparent, multi-cloud Kubernetes in cloud-native applications, across multiple Kubernetes clusters and cloud providers. The report is not public yet, but it will be by end of 2024 and will contain relevant findings which have been addressed by the Karmada team. Quoting one of the Karmada lead developers zhzhuang-zju: "*During our recent security audit, your team demonstrated exceptional expertise in the realm of cloud-native multi-cluster security. Your professional testing methods and modeling techniques have uncovered numerous typical security issues, contributing significantly to the robust development of our project.*". The Karmada findings will be probably presented[1] at the KUBECON 2025 with a joint talk by the Karmada and the Shielder teams.

In addition to the direct experience reported above, it is also worth mentioning that, during the typical red teaming and penetration testing activities carried out by Shielder on a day-to-day basis, the team often encounters applications running on Kubernetes clusters, deployed in a plethora of different environments, ranging from on-premise to cloud providers. Therefore, the team has gained practical, first-hand experience on how attackers typically approach k8s clusters when looking for privilege escalation and lateral movement paths. This practical experience is often valuable in threat modeling assessments, since it is fueled by data coming from real scenarios.

4.2. *Deliverables*

The security engagement will have the following deliverables:

- A set of issues opened through the vulnerability disclosure procedures of the affected components and subprojects. The issues will be reported as soon as the findings will be identified. Each report will be concise, it will contain a clear explanation of the vulnerability, a set of easy steps to reproduce the issue, and the suggested remediations.
- A set of pull requests to implement unit-tests, fuzzing harnesses, and automated security testing based on what will be developed during the engagement. The goal is to provide the projects with long term improvements and decrease the likelihood of regressions.
- A set of tools/scripts developed during the engagement which could be used by the Kubernetes contributors and/or by security researchers to simplify observation of security-sensitive information about a cluster.
- Documentation on how to setup debugging labs to simplify and enhance vulnerability research activities on Kubernetes components. The goal is to lower the entry barriers for security researchers willing to contribute to k8s. The content of

[1] Subject to CFP approval.

this documentation will depend on the challenges that the Shielder team will face during the engagement, but some of the possible entries include:
  - How to intercept the intra-cluster network communications.
  - How to attach a debugger to the Windows Kernel of a Pod.
  - How to audit seccomp/AppArmor logs related to a Pod.
- A comprehensive final report which will contain the output of the whole assessment, including all the findings, all the covered areas, any shadow corner, and the long-term suggestions for the projects and for future audits.

The comprehensive report will contain the following sections and information:

- **An Executive Summary** – a high-level description of the security posture of the components, focused on the most harmful attack scenarios and the most relevant vulnerability classes.
- **A Technical Details** – where a deep dive of each vulnerability is presented along with the following information:
  - *Classification* – in terms of Critical, High, Medium, Low, or Informative based on the impact of the vulnerability.
  - *Description* – what is the vulnerability and where was it found.
  - Impact – what an attacker could do by exploiting the vulnerability.
  - *Attack Complexity* – which are the limitations for an attacker to exploit the vulnerability.
  - *Related Issues* – list and description of other vulnerabilities which could ease the exploitation.
  - *Proof-of-Concept* – a step-by-step guide to reproduce the vulnerability and/or a list of evidence (e.g., screenshots, snippets of code, …).
  - *Suggested Remediation* – actions to be taken to mitigate the vulnerability based on the state of art.
  - *References* – list of external references to deepen the knowledge about the vulnerability class and the common approaches for detection and mitigation.

5. Project Planning

OSTIF has provided Shielder with all the pertinent information required to define the scope of the project and assess its complexity. However, before initiating the project, it may be useful to schedule a *kick-off* call to delve into more details. Additionally, considering the nature of the project, additional information-sharing sessions may be required during the testing phase.

The project is proposed with a flat-rate approach; therefore, the number of person/days will be subject to various factors. The goal is to complete the project by the end of the first half of 2025, with the final comprehensive report being published during the second half of 2025.

This proposed timeline is subject to consolidation upon the acceptance of the business proposal.

6. Assumptions and Limits of the Project

A list of the assumptions and limits of the services requested follows:

- The security analysis provided by Shielder will be limited to the perimeter agreed with the customer.
- All the project's deliverables will be reviewed by the customer and then publicly published by both the parties.
- All the phases of the project will be performed remotely.
- Security assessments are time-boxed activities performed at a specific point in time; as such, their aim is to highlight the current security posture of the software and uncover vulnerabilities that can be detected by an experienced team during a limited amount of time. It must be acknowledged that these activities are unable to guarantee that a software is or will be free of bugs.

7. Data Sharing

Shielder suggests sharing sensitive information in one of the following ways:

- Email:
    - The documents could be shared using asymmetric encryption based on the PGP protocol. If this way is agreed Shielder will provide its team public key and the customer will do the same.
    - The documents could be shared using symmetric encryption using a secret shared through a different communication channel (i.e., SMS).
- Third-Party encrypted file sharing services:
    - If the customer has an encrypted file sharing service Shielder could share the documents through it.
- If the previous solutions are not acceptable by the customer, a tailored option will be discussed.

8. References and Certifications

Shielder is a trusted partner for a wide variety of customers who want to secure their critical assets. Due to the confidentiality requirements of some of the projects where Shielder is included, which are often a matter of national security, it is not possible to include a full list of its customers.

That said, here is a list of some private companies which relays on Shielder to evaluate the security of their products and infrastructures: Aizoon, EssilorLuxottica, Fastweb, Ferrero, Karmada, KPMG, Nozomi Networks, OSTIF, PayPal, Satispay, Telepass, UL, Würth Phoenix, Yunex Traffic, etc.

Some of the certifications held by Shielder team members follow:

- OSCP (Offensive Security)
- OSWE (Offensive Security)
- OSEP (Offensive Security)
- ECPPTv2 (eLearnSecurity)
- EXPTXv2 (eLearnSecurity)
- CRTE (PenTester Academy)
- CARTP (AlteredSecurity)
- CRTO (ZeroPointSecurity)

Company certifications:

- ISO/IEC 27001:2013 – EA 33, 35, 37
- ISO 9001:2015 – EA 33, 35, 37

Research is one of Shielder's pillars. Our team invest from 25% to 100% of time into 0day vulnerability research, exploit development, and training. By constantly pushing the boundaries of our knowledge and discovering new vulnerabilities, we contribute to the security of the digital ecosystem.

Some of the research team achievements are:

- Being in the Google Vulnerability Research Program (VRP) hall of fame.
- Being in the Facebook bug bounty program hall of fame.
- Being in the Twitter bug bounty program hall of fame.
- Being in the Microsoft bug bounty program hall of fame.

A list of some of the recent vulnerabilities found by our team follows (the full list could be found on the corporate advisory page[2] and blog[3]).

[2] https://www.shielder.it/advisories/
[3] https://www.shielder.it/blog/

CVE(s)	Vulnerability
N/A - 2024	Mozilla Thunderbird Appointments - Regular Expression Denial of Service (ReDos)
CVE-2024-42994 CVE-2024-42995	Vtiger CRM Multiple Vulnerabilities
CVE-2024-36455 CVE-2024-36456 CVE-2024-36457 CVE-2024-36458 CVE-2024-38491 CVE-2024-38492 CVE-2024-38493 CVE-2024-38494 CVE-2024-38495 CVE-2024-38496	Symantec CA PAM Multiple Vulnerabilities
CVE-2024-2044	PgAdmin4 – Remote Code Execution via Unsafe Deserialization
CVE-2024-26132 CVE-2024-26131	Element Android Multiple Vulnerabilities
CVE-2024-24752 CVE-2024-24753 CVE-2024-24754 CVE-2024-29186	Bref Multiple Vulnerabilities
N/A - 2023	Google Chrome - ServiceWorkers Can Access Cookies in Credentialless iframes
CVE-2023-28634	GLPI - Privilege Escalation from technician to super-admin
CVE-2022-3308	Google Chrome - Insufficient policy enforcement in Developer Tools
CVE-2022-1128	Google Chrome - Inappropriate implementation in Web Share API
CVE-2022-27873	Autodesk Fusion 360 <= 2.0.12887 "Insert SVG" Blind XXE
CVE-2021-41282	Remote Code Execution in pfSense <= 2.5.2
CVE-2021-38136 CVE-2021-38137	Multiple vulnerabilities in Corero SecureWatch Managed Services 9.7.2.0020
MMSA-2021-0055	Mattermost Server v5.32 > v5.36 Reflected XSS in OAuth flow
CVE-2021-28807	QNAP Q'center Post-Auth Remote Code Execution via QPKG
CVE-2021-28807	QNAP Q'center Virtual Appliance < 1.12.1014 Stored XSS
CVE-2021-31316 CVE-2021-31324	CentOS Web Panel idsession root Remote Code Execution
CVE-2021-31317 CVE-2021-31318 CVE-2021-31319 CVE-2021-31320 CVE-2021-31321 CVE-2021-31322	Telegram Multiple Vulnerabilites in Animated Stickers
CVE-2020-36197 CVE-2020-36198	QNAP MusicStation/MalwareRemover Pre-Auth Remote Code Execution
CVE-2020-17148	Microsoft Visual Studio Code SSH Plugin Remote Code Execution

9. Privacy and Data Protection

Shielder will process the data of the Customer and its contact persons in compliance with Regulation (EU) 2016/679 "GDPR" and more generally with Italian and European legislation on privacy and protection of personal data (the "Applicable Law").

The Customer – the party who owns or controls the data – engages Shielder – the party providing data processing services – to process personal data on its behalf in accordance with Art. 4 Sec. 2 and Art. 28 of the Regulation (EU) 2016/679 on the basis of the present Agreement.

In carrying out the activity, Shielder may process personal data of third parties of which the Customer may be the "Data Controller" and/or the "Data Processor". Shielder will manage such information according to the specifications on the processing of personal data pursuant to Art. 28 of the GDPR.

As a Data Processor, Shielder collects and processes various types of personal data solely for the purpose of facilitating the provision of Services as requested by the Customer. The specific Services to be rendered are detailed in the present Business Proposal. Shielder shall perform processing activities that are necessary and relevant to ensuring the execution of the Services, following the principles of purpose limitation and data minimization.

As a Data Processor, Shielder processes Customer’s data based on the instructions and legal grounds provided by the Data Controller. These legal grounds may include:

- Consent, where applicable, as provided by the Data Controller.
- The necessity of the processing for the performance of a contract or to take pre-contractual steps at the request of the Data Controller.
- Compliance with legal obligations to which the Data Controller is subject.
- Other legitimate interests pursued by the Data Controller, as communicated to Shielder.

Shielder will process any personal data in accordance with the principles of lawfulness, fairness, transparency, purpose limitation, data minimization, accuracy, storage limitation, integrity, and confidentiality.

Shielder takes data security seriously and implements appropriate technical and organizational measures to protect all collected data, as required by Art. 32 of the Regulation (EU) 2016/679. These measures guarantee an appropriate level of protection against potential risks posed to the confidentiality, integrity, and availability of personal data. The state of the art, the costs of implementation, and the nature, scope, context, and purposes of processing, as well as the risk of varying likelihood and severity for the rights and freedoms of natural persons, are all taken into account when implementing these measures.

All rules governing the data processing activities of Shielder are contained in the Data Processing Agreement, which ensures compliance with relevant data protection laws and regulations.

For any questions or concerns regarding the processing of personal data by Shielder, please contact our Data Protection Officer at dpo@shielder.it.

9.1. *Confidentiality*

The information contained in this document must be considered strictly confidential.

Shielder and the Customer are therefore required to:

- Do not use them for purposes other than the evaluation of the proposal.
- Do not disclose and ensure that it is not disclosed directly or indirectly to anyone other than the people directly involved in the evaluation of the same.
- Do not copy, reproduce or duplicate them without the prior written consent of Shielder.

10. Contacts

The relevant contact of the Shielder team follows.

Name Surname	Email	Phone Number
Abdel Adim Oisfi	abdeladim.oisfi@shielder.it	(+39) 393 – 16 66 814
Claudio Mancini	claudio.mancini@shielder.it	(+39) 345 – 57 18 634