




Cybersecurity Vulnerability Prioritisation via Risk Assessment

Steve Taylor¹, Panos Melas¹, Mike Surridge¹, Paolo De Lutiis², Manuel Leone², Martin Gilje Jaatun³, and Ravishankar Borgaonkar³

¹ University of Southampton, Southampton, UK
{S.J.Taylor, pmelas, ms8}@soton.ac.uk

² Security Engineering and Threat Management, TIM S.p.A, Torino, Italy
{paolo.delutiis, manuel.leone}@telecomitalia.it

³ Software Engineering, Safety and Security, SINTER Digital Trondheim, Norway
{Martin.G.Jaatun, Ravi.Borgaonkar}@sintef.no

Abstract. The Common Vulnerabilities and Exposures (CVE) database lists a large number of vulnerabilities that are present in specific versions of software libraries and applications, but although there is a severity ranking, it does not immediately follow that an identified vulnerability with high severity will be particularly important for a specific application. This paper presents the motivation for CVE Prioritization for a given case and describes an outline process for evaluating the priority of CVEs via risk assessment simulations.

Keywords: Vulnerabilities, CVE, SBOM, Risk, Software

1 Introduction

A Common Vulnerabilities and Exposures (CVE) instance describes a specific security issue in software or hardware, assigning it a globally unique identifier (e.g., CVE-2025-12345). It helps security professionals track and address security risks effectively in a standardized manner. The CVE initiative was launched at the beginning of 2000 by the US MITRE organization [1] and is today commonly used by the cybersecurity expert community and industry: many tools, such as scanners and (cyber) threat intelligence platforms, use CVE to identify and track vulnerabilities.

The total count of CVEs increases significantly every year. In 2025, more than 45.000 published vulnerabilities are expected [2]. This high number clearly highlights the complexity of vulnerability management, requiring security experts and companies to invest significant effort and resources to effectively safeguard their networks and mitigate potential risks. In fact, addressing this large number of vulnerabilities is often impractical due to market-driven time pressures, limited budgets, or the constrained remediation capabilities of device manufacturers. For instance, the Fraunhofer Institute reported between 348 and 579 high-severity CVEs per device [3]. Therefore, it is important to have tools and methodologies that help prioritize vulnerability remediation to achieve effective risk reduction at reasonable effort cost.

2 Related Work

2.1 Current Approaches

The Common Vulnerability Scoring System (CVSS) [4] uses a scoring approach, where the score ranges from 0 to 10, higher numbers indicating more critical security risks. The value represents the severity of a cybersecurity vulnerability based on the related impact and exploitability. The score does not represent the likelihood of an attack, although some parameters can be used indirectly to evaluate such aspects (e.g., if a CVE can be exploited via network, its exploitability likelihood is greater than a CVE that requires physical access to the device to be exploited). CVSS values (in particular CVSSv3, but partially also CVSSv4) are global and static values bound to the CVE intrinsic characteristics, without considering real-world factors like attacker motivation, availability of Proof of Concept (PoC) or public exploits, the target device’s configuration and its positioning within the network architecture, the presence of additional security mechanisms, such as firewalls, Intrusion Prevention System (IPS), etc. To overcome these limitations, Security teams typically consider additional sources of information. A key source is Cyber Threat Intelligence feeds that take into consideration the actual threats/risks the device containing the CVE is exposed to.

The Exploit Prediction Scoring System (EPSS) proposed by the FIRST organization in 2019 [5] is a framework based on Machine Learning algorithms, that estimates the probability of a CVE being exploited in the wild, thus helping organizations prioritize patching by focusing on vulnerabilities that are more likely to be actively attacked.

The Vulnerability Priority Rating (VPR), proposed and maintained by Tenable [6], is designed as an enhancement over the traditional CVSS scores and incorporates threat intelligence, vulnerability age, exploit availability, and asset context to prioritize vulnerabilities and to help organizations to focus on vulnerabilities that are most likely to be exploited in real-world attacks. The Known Exploited Vulnerabilities (KEV) catalog [7], maintained by the Cybersecurity and Infrastructure Security Agency (CISA), aims to identify vulnerabilities that are actively being exploited by cybercriminals. These vulnerabilities have been confirmed to be used in real-world attacks, making them high-priority targets for patching. While KEV primarily strengthens the security posture of U.S. government agencies, it also serves as a valuable resource for organizations worldwide.

Several initiatives on this topic have also been registered in the commercial domain. Among these, we mention Cisco Vulnerability Management (formerly Kenna.VM [8]), whose primary goal is providing a prioritized list of vulnerabilities analyzing data ingested from several sources (such as vulnerability scanners) and combining them with real-world exploit activity.

2.2 Main Limitations

The proposed methods for CVE prioritization have their own advantages and limitations. As correctly stated by Spring et al. [9], CVSS is primarily designed

to assess the technical severity of a vulnerability; but it is often misused for vulnerability prioritization and risk assessment. While CVSS measures severity, even with the inclusion of Temporal and Environmental scores, it does not assess risks. Consequently, its effectiveness for vulnerability prioritization remains limited. Moreover, it is not suitable for handling deployment in complex scenarios, nor can it be used to aggregate scores across multiple vulnerabilities [10]. EPSS, VPR and partially KEV enhance severity measurement through the contribution of Threat Intelligence. However, their support for multiple vulnerabilities remains limited, and like CVSS, they do not account for specific characteristics of the device under test and the system into which it is deployed.

3 Case Study: Residential Gateway

The case study for this work concerns a Residential Gateway (RGW). The RGW is a commodity device that provides connections for domestic subscribers of broadband services to the Wide Area Network (WAN) provided by the Internet Service Provider, which in this case study is a telecommunication company (TC). TC provides various communication services, including mobile telephony and residential data services. TC has defined rigorous security testing processes and a risk-based methodology to manage and maintain the cybersecurity of its infrastructures. Every device, prior to deployment in the field, must be tested within specialized labs to verify its actual security posture.

The RGW is a special kind of device from the cyber security point of view. Although it is based on a low-cost architecture, it is a key element for the interconnection (and security) between the internal LAN of the residential customers (where it is common to have IPTV, PC, and other user devices) and the Internet, which provides at the same time digital services but also cyber threats and risks. The patch management of such devices is also complex, especially considering that the number of devices deployed can be several million. Therefore, the efficient patching of the devices is a key requirement, and for this the assessment of exploitability and severity for CVEs affecting the RGW is needed to determine priorities for patching.

The conditions required to exploit vulnerabilities in a Residential Gateway can be highly specific to that device, potentially limiting the impact of the Threat Intelligence in accurately assessing the actual risk. The environment into which the RGW is deployed is also a key factor - the fact that a specific vulnerability is actively exploited in the wild may be irrelevant for a Residential Gateway due to its particular access model, which could prevent actual exploitation. Conversely, detecting that a specific device has been successfully targeted may occur too late. This challenge is especially evident for RGW devices, where the software or the device configuration is often customized by the Telco operator, further complicating traditional threat intelligence assessments. Moreover, in such a situation, these solutions provide limited assistance to executive management in understanding the real risk posture and how to mitigate a potential future security disaster.

In short, none of the methods considered are truly tailored to the target devices. While this is reasonable for general-purpose approaches, more specialized strategies could be applied in specific use cases or when the target device is well known, as in the RGW use case. These strategies should rely on more detailed models capable of accurately assessing the real exploitability of vulnerabilities and combining them with the effectiveness and power of the risk analysis approach.

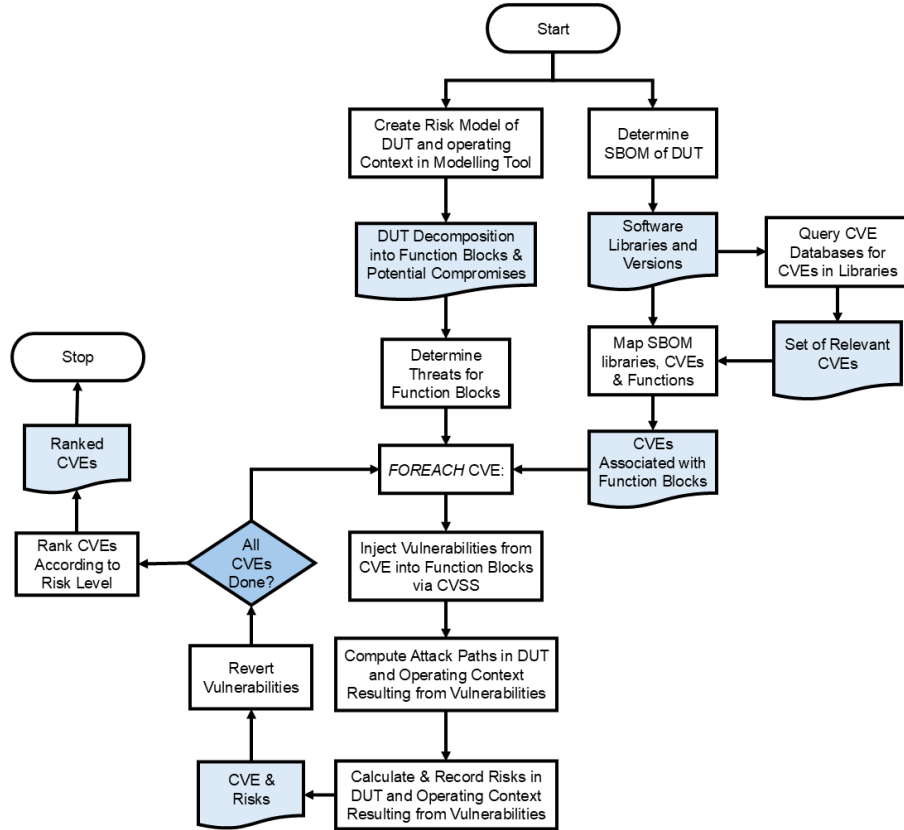


Fig. 1. CVE Prioritization Approach

4 Approach

Our approach to addressing these limitations is to use risk simulation to assess the compromises, along with their impacts and likelihood (key components of risk assessment), potentially caused by each CVE in the Device Under Test

(abbreviated DUT). The overall approach is shown in the process diagram of Fig. 1 and detailed steps of this process are discussed in the next subsections.

Risk assessment typically maps threats to consequences, where a consequence has a severity (the degree of the damage were the consequence to occur) and a likelihood (how probable the consequence is). Vulnerabilities are weaknesses that threats exploit, and thus risk assessment enables mapping of vulnerabilities in CVEs to consequences of importance to the stakeholders undertaking the analysis.

In this work, we have used the Spyderisk tool⁴ (Phillips et al [11]), a tool for risk simulation via a knowledge-based modelling approach. Here the user builds a model of their System Under Test (SUT) using predefined ICT elements such as computers, software processes, data, networks, routers plus the socio-technical environments and actors they operate in such as people and physical spaces.

In the Case Study, a model (shown in Fig. 2) is built of the RGW under test (the DUT), crucially in its deployment environment (the SUT), which enables relating risks in the deployment environment to vulnerabilities associated with CVEs. The deployment environment is a domestic situation containing typical elements on the private home network (e.g. computers, TVs, smart devices), users (e.g., the subscriber and their family), data (e.g., documents, music, photos, all of which are likely to be important to the subscriber and thus need protecting) plus the connection to the WAN provided by the ISP and the bridge to the Internet. In parallel, a Software Bill of Materials (SBOM) is generated from the DUT's binary firmware, which provides software packages and versions, and using this information, CVEs can be determined via lookup. The risk model is used to evaluate the CVEs associated with the RGW in simulation of the effects caused by the CVE under realistic expected usage conditions.

4.1 Attack Paths

A key concept from Phillips et al. [11] that underpins the approach described here is that of *attack paths*. These are chains of vulnerability-threat-consequence patterns, where a vulnerability (e.g. represented by a CVE) in a system component exposes it to a threat, which leads to a consequence (e.g., degradation of a key property of a component), which may also lead to a new vulnerability, which may lead to another threat, and so on. The consequences can be measured as risks, which comprise the severity of the consequence occurring, which is a subjective judgement and dependent on the stakeholder(s) involved or affected, and the likelihood, which is determined by the potential for exploitation of the vulnerability, the difficulty of executing the threat, any controls in place to manage threats, etc. Thus, from this repeating pattern of vulnerability-threat-consequence, attack paths can be formed that link vulnerabilities in one system component to consequences in it, or other connected entities. Via this mechanism, our approach is to evaluate CVEs by determining their simulated

⁴ <https://spyderisk.org>

resulting risk levels, to identify which vulnerabilities lead to the highest-level risks and thus which CVEs should be prioritized.

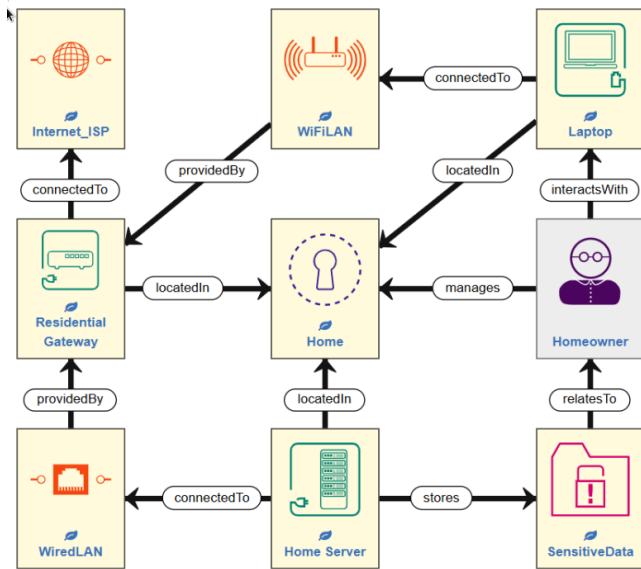


Fig. 2. Residential Gateway (router) in Context

An illustrative example of an attack path is provided in Fig. 2 and Table 1 showing a Residential Gateway (RGW) in a domestic context. The RGW provides: access to the Internet, a wired and wireless network, to which a laptop and a home server are connected. The home server stores sensitive personal data relating to the Home owner.

Here, a key consequence is: Loss of Confidentiality (or low Confidentiality) at SensitiveData, which is defined as: “*Disclosure of data to unauthorised parties, or a state where prevention or detection of such a disclosure cannot be ensured*”. Here, because the data is sensitive, the impact (or severity) of the unauthorized disclosure is high, and its likelihood is calculated to be very high. An Attack Path to this consequence is shown in Table 1.

The table illustrates that the consequence in one row leads to the vulnerability in the row beneath it. Here the root vulnerability is the assumed (low) trustworthiness (TW in the table) of everyone in the world leading to an intrusion into the Home. This leads to the consequence of theft of the Home Server and thence to the exposure of the Sensitive Data stored within it. This is a trivial example because the simple control of locking the door of the Home addresses it (by restricting access to trusted individuals), but it serves to describe the link between vulnerabilities, threats and consequences in an attack path. Because the risk model is constructed of the RGW in the domestic situation in which it is

Table 1. Attack Path to Loss of Confidentiality of Sensitive Data

Vulnerability	Threat	Consequence	Distance	Likelihood
Occupant TW at <i>World</i>	Physical intrusion into private space <i>Home</i> from <i>World</i>	Loss of Occupant TW at Home	5	Very High
Loss of Occupant TW at <i>Home</i>	Theft of device <i>Home Server</i> from <i>Home</i>	Loss of Possession at <i>Home Server</i>	3	Very High
Loss of Possession at <i>Home Server</i>	Physical access to data <i>SensitiveData</i> on stolen host <i>Home Server</i>	Loss of Confidentiality of <i>SensitiveData</i>	1	Very High

deployed, it represents the relationship between vulnerabilities in the DUT and the consequences they cause in the environment in which the DUT is deployed. This method of attack path analysis therefore considers the effects of chains of vulnerabilities and also considers the relationship between the DUT and its deployment environment, overcoming limitations of current approaches.

4.2 Functional Decomposition & Modelling

Fig. 2 illustrates a model of the RGW in context, and to map CVEs to the router, a functional decomposition of the core functionality of the RGW is undertaken. This entails examining the core functions of the router and building a risk model of these core functions, how they are configured and mapping this to software packages from the SBOM. The complete Spyderisk model for the RGW is shown in Fig. 3 and its core functionality is decomposed as shown in Fig. 4.

A basic residential gateway has a connection to the Internet, wired ethernet ports and a wireless access point. The RGW under test is based on OpenWrt⁵, whose networking relies on the Linux kernel networking subsystem which provides packet processing and routing functionalities. Kernel modules interact with device drivers that handle WAN, LAN, and WiFi interfaces, or hardware accelerators. Packet filtering and network address translation (NAT) are managed by the kernel's Netfilter framework. User-space services and tools such as ppp, dhcp, fw3, and hostapd manage various router networking functionalities including the WiFi access point. Their behaviour is configured through `/etc/config` files which can be modified via the unified configuration interface (LuCI / uhttpd) via a web browser.

The orange connections in Fig. 4 depict flows of control or management, from managing to managed component, and each path of configuration or management represents a potential attack path, since any exploitable vulnerability at a point on this path can affect the components downstream. For example, if there is a vulnerability in uhttpd, this can have potentially far-reaching effects, since

⁵ <https://openwrt.org>

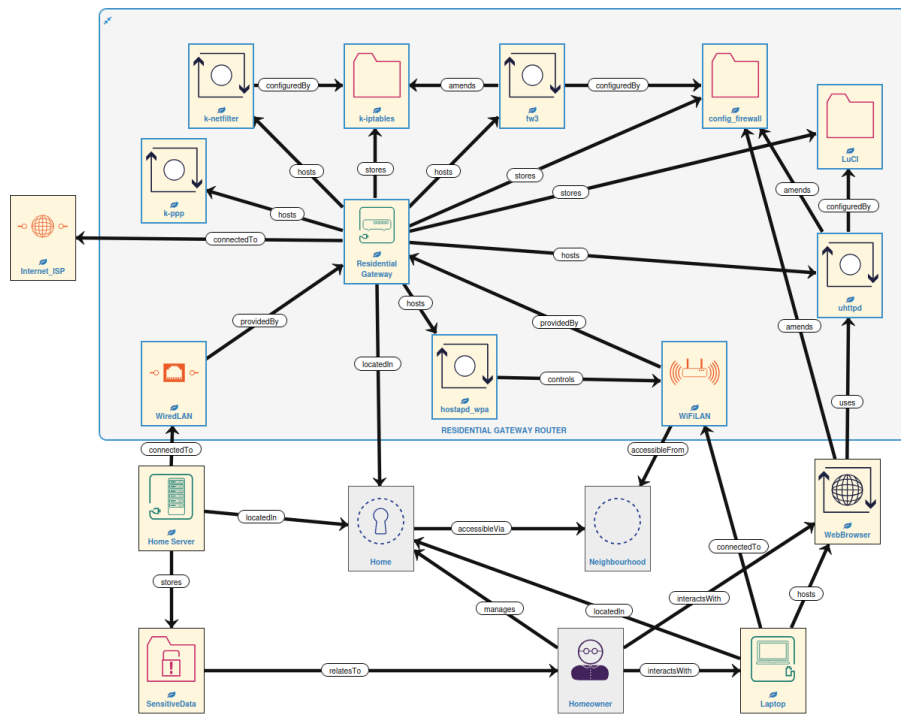


Fig. 3. RGW Spyderisk model

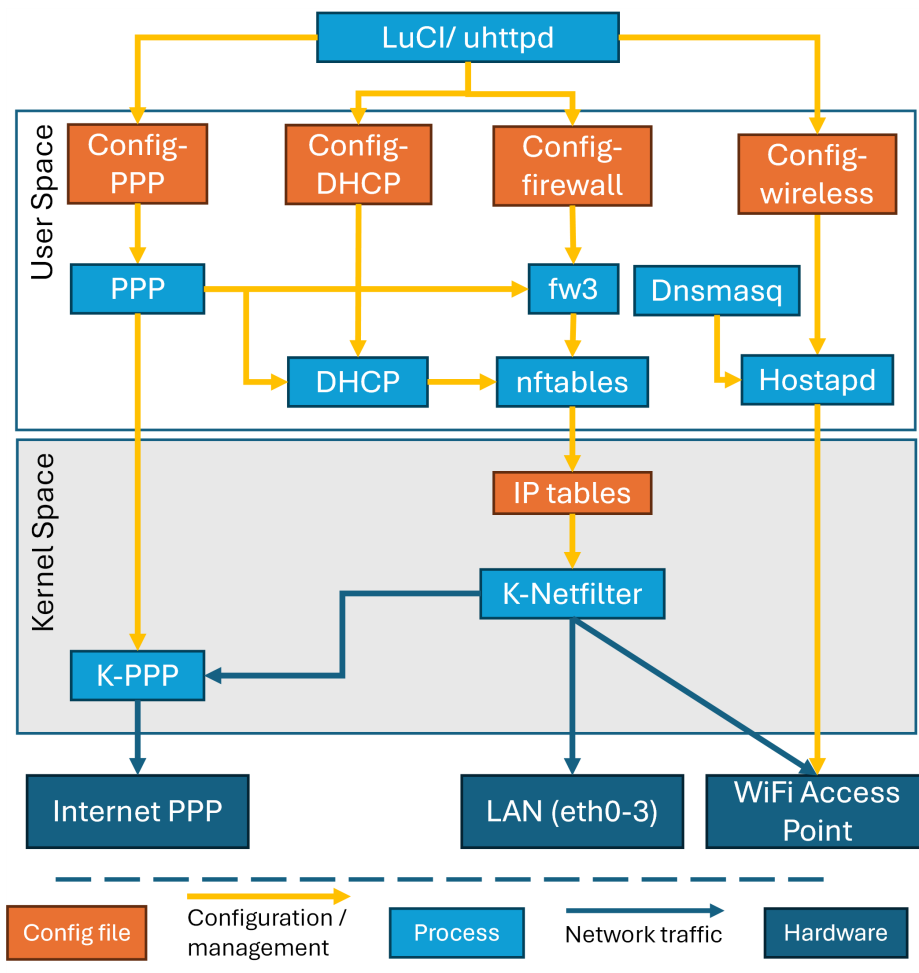


Fig. 4. RGW Functional Decomposition

it is the means by which all the downstream processes are configured, and which eventually controls the hardware networks provided by the RGW.

Using this functional decomposition, a risk model of the RGW is constructed in the simulation tool described by Phillips et al. [11]. The model uses “process” elements to describe the software processes (blue elements in Fig. 4) and “data” elements to describe configuration files (orange elements in Fig. 4), as well as specific elements for the hardware Internet, LAN and WiFi interfaces. These are connected as described in Fig. 4, thus modelling the software processes that control the networking elements, the config files that control them and the software processes that enable the config files to be updated. The resulting model is not shown for reasons of space but it is an expanded model of the RGW element of Fig. 2 (the RGW in context) with the topology of Fig. 4 (the software / hardware structure of the RGW).

Each of the processes in the RGW decomposition is a potential source of vulnerabilities, and determining which vulnerabilities may be applicable requires SBOM generation for the packages that provide that process.

4.3 SBOM Generation

In an ideal world, all vendors would include an up-to-date SBOM with their software/devices, but this is often not the case. If an unencrypted, un-obfuscated version of the binary firmware is available, tools can be used to extract library names and version strings directly from the firmware [12,13]. This can then be used to construct a (partial or complete) SBOM for further analysis.

An SBOM in the Software Package Data Exchange (SPDX) format may be generated using the Open Package Manager (OPKG) system. OPKG is a lightweight package management system that is used in OpenWrt without needing access to source code. OPKG can be used to provide a list of the installed software packages including package details such as:

- Package name
- Version name
- Depends list
- Architecture name
- Installed-Time
- List of installed files

This information is enough to construct an SPDX SBOM file with minimal mappings as shown in Table 2.

Table 2. Mapping OPKG to SPDX

OPKG Attribute	SPDX Attribute
Package	PackageName
Version	PackageVersion

4.4 Mapping CVE vulnerabilities to risk model

The SBOM package names and versions of the SBOM packages can be used as arguments to queries to look up CVEs from databases such as National Vulnerability Database⁶ using client-side query tools such as the CVE Bin Tool⁷. The result is a set of CVEs associated with the SBOM.

Each CVE returned by the query has associated CVSS vectors, and these vectors contain metrics that enable mapping into the risk model, via a mechanism named “Trustworthiness Attributes” (TWA). The mapping between CVSS and TWA is described by Phillips et al. [11], but briefly TWA describes attributes such as Confidentiality, Integrity, Availability (CIA) for data; and access vector, authentication requirements, user trustworthiness, availability, intrinsic trustworthiness, reliability, timeliness, malware, overloading and control for processes. CVSS vectors contain attributes covering access vectors to processes and data, e.g., whether it is remote or local, the attack complexity and the authentication required to exploit the vulnerability; plus impact in terms of Confidentiality, Integrity and Availability (CIA) on data and processes, which can be mapped to the TWA. Example mappings are described in Section 5.

4.5 Risk Assessment Per CVE

The risks resulting from each CVE are computed via an iterative loop. Each iteration performs the following actions:

1. Adjust the relevant TWAs associated with the CVE’s CVSS vector.
2. Calculate the attack path and resulting risks, thus determining the risks due to the CVE.
3. Record the risks associated with the CVE.
4. Reset the risk model to a known state.

The result is a set of attack paths and risks for each CVE in the SUT where the DUT is deployed. These results can be then ranked to show the CVEs with the highest level risks.

5 Example

Initial tests have been undertaken by mapping vulnerabilities into a risk model representing the RGW structure of Fig. 4 in the context of a domestic environment of Fig. 2. Appropriate controls were applied to the model to simulate a realistic environment (e.g. physical locks on the Home, encrypted communication in the wireless network, firewall blocks at the interface between the RGW and the Internet). The RGW under test is running OpenWrt, an open-source GNU/Linux distribution firmware for embedded devices. An SBOM was generated for OpenWrt and using queries based on the libraries and versions in the

⁶ <https://nvd.nist.gov/>

⁷ <https://github.com/intel/cve-bin-tool-action>

SBOM, CVEs were retrieved from the NVD database. These were successively applied to the risk model and risks calculated. An example CVE is discussed next.

CVE-2020-8597⁸ describes a buffer overflow vulnerability in the pppd (Point-to-Point Deamon) process with the CVSS vector:

AV:N/AC:L/Au:N/C:P/I:P/A:P

The vector indicates a remote code execution network based attack (AV:N) of low complexity (AC:L) that does not require authentication (Au:N). This results in partial compromise of Confidentiality, Integrity and Availability of data in the affected processes (C:P/I:P/A:P), which is interpreted as user-level access, according to scoring tip#9 of the CVSS v2.0 guide⁹. The vector is mapped to TWAs in the k_ppp model process as described in Table 3.

Table 3. Trustworthiness Attributes changes

Trustworthiness Type	From	To	Basis
Extrinsic-AU-TW	"Safe"	"Low"	No authentication needed (Au:N)
Extrinsic-U-TW	"Safe"	"Low"	User-level access trustworthiness from CIA:P
Extrinsic-VN-TW	"Safe"	"Low"	Remote network access trustworthiness from AV:N

In each of the TWAs, the value “Low” is determined by the attack complexity AC:L, meaning a low-complexity attack, therefore low trustworthiness results. The risk calculation is run and the resulting risk level becomes Very High, due to the significant rights potential gained from a low-complexity remote attack.

6 Conclusion

This paper presents an outline process for assessment of vulnerability (CVE) priority in Devices Under Test via linking CVEs in devices to system-level risks and using risk levels at system-level to determine the CVE priority. The outline addresses challenges of incorporating the DUT’s deployment environment and the propagated effects of vulnerabilities in the DUT to risks that affect important assets at system level. The approach enables analysts to declare their priorities in terms of risk severity, then use attack path analysis to determine the likelihood that the vulnerabilities lead to compromise of these priorities. Future work is aimed at running extensive tests on the use case, plus addressing the following sub-challenges.

1. Model resolution. The RGW is a complex system, including multiple packages in addition to the core gateway / routing functionality. Further work

⁸ <https://nvd.nist.gov/vuln/detail/CVE-2020-8597>

⁹ <https://www.first.org/cvss/v2/guide>

- is needed to determine the appropriate fidelity of the risk model, including modelling the critical and non-critical functions of a given router (even if functions are non-critical to the operation of the router, they may still provide entry points to attack paths to critical functions).
2. CVSS versions. CVSS is continually evolving and in many cases later versions are not backwards compatible with earlier versions because semantics of the vector has changed across the version. Thus it is challenging to support mapping from CVSS as it evolves to risk assessment and further work is needed to determine a robust mapping scheme.
 3. Function to Software Package Mapping. It is often the case that a core element of RGW function is delivered via multiple software packages, or that the same package delivers multiple RGW functions. Thus the mapping between the functions and actual packages is not straightforward and careful analysis of both the functional breakdown and the packages that support it is needed, requiring significant domain knowledge of hardware, software, networking etc.
 4. Exploitability of vulnerabilities in realistic system-level situations. Understanding under what circumstances a vulnerability becomes exploitable is needed via consultation with experts to build risk models that enable simulation of realistic scenarios where the RGW may be employed including different domestic environments, taking into account factors such as the types of smart devices connected to the private network, default configuration of such devices and different levels of cybersecurity awareness of domestic subscribers. In addition, understanding of the relative importance of different elements in the system (i.e., what is important to protect) is a key consideration. Evaluating the factors above to determine such scenarios is an immediate item of further work.

Acknowledgment

This work is performed within the Horizon Europe TELEMETRY (Trustworthy mEthodologies, open knowLedgE & autoMated tools for sEcurity Testing of IoT software, haRdware & ecosYstems) project, supported by EC funding under grant number 101119747, and UKRI under grant number 10087006.

References

1. MITRE: Mitre corporation. <https://www.mitre.org/>
2. Éireann Leverett: Vulnerability forecast for 2025. <https://www.first.org/blog/20250607-Vulnerability-Forecast-for-2025> (2025) FIRST.
3. Weidenbach, P., vom Dorp, J.: Home router security report 2020. Technical report, Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE (2020)
4. CVSS Special Interest Group: Common vulnerability scoring system version 3.1: Specification document (2019) Accessed 2025.

5. Roytman, M., Jacobs, J.: Predictive vulnerability scoring system. In: Black Hat USA, Las Vegas (August 2019)
6. Tenable: Vulnerability priority rating. <https://docs.tenable.com/vulnerability-management/best-practices/security/Content/VulnerabilityPriorityRating.htm>
7. Cybersecurity Infrastructure Security Agency (CISA): Known exploited vulnerabilities catalog. <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
8. Cisco: Cisco vulnerability management. <https://www.cisco.com/site/us/en/products/security/vulnerability-management/index.html>
9. Spring, J., Hatleback, E., Householder, A.D., Manion, A., Shick, D.: Towards improving CVSS. White paper, Carnegie Mellon University – Software Engineering Institute (2018)
10. FIRST: EPSS - frequently asked questions. <https://www.first.org/epss/faq>
11. Phillips, S.C., Taylor, S., Boniface, M., Modafferi, S., Surrige, M.: Automated knowledge-based cybersecurity risk assessment of cyber-physical systems. *IEEE Access* **12** (2024) 82482–82505
12. Messner, M., Kuehne, B., Eckmann, P., Hoxha, E.: Emba - the security analyzer for firmware of embedded devices. <https://github.com/e-m-b-a/emba> (Multiple other contributors).
13. Linux: strings(1) - linux man page. <https://linux.die.net/man/1/strings>