Vulnerability Categorization for Fast Multistep Attack Modelling

Dmitry Levshun, Andrey Chechulin
St. Petersburg Federal Research Center of the Russian Academy of Sciences
St. Petersburg, Russia
{levshun, chechulin}@comsec.spb.ru

***Abstract*—For many years, attack graphs have been one of the most popular approaches to model multistep attacks. This approach allows evaluating the possibility of each host in the system being compromised and to find attack paths with the most probability and impact. This paper describes an original approach to vulnerability categorization for fast multistep attack modelling. The novelty of the approach lies in the categorization of all available CVEs into 24 categories in accordance with their access vector, initial, and obtained access rights. After that, instead of vulnerabilities, only their categories are used for constructing attack graphs of each host of the analysed system. It helps to make this process more computationally efficient for each host, while those computations can be done in parallel. Moreover, we introduce assumptions to integrate second and third versions of the CVSS vulnerabilities descriptions and allow transitions of the attacker between different access vectors. For the experimental evaluation of the approach, it was decided to generate hosts with 10 random CVEs and CPEs, from 10 hosts to 250, while 10 hosts are added on each step. After that, for each host it is analysed if it is vulnerable based on the list of CVEs and their categories. Each step of the host generation was done 5 times, and average time consumption results are taken as a result. After that, the same experiment was redone, but with 50 random CVEs and CPEs for each host. The results showed that the suggested approach is 13.4 times faster at average for 10 CVEs and CPEs, while 23.0 times faster for 50 CVEs and CPEs. Moreover, we tested the suggested approach on a fixed number of hosts equal to 100, while changing the number of random CVEs and CPEs per host from 10 to 100 with the step equal to 10. This experiment showed that the categories-based approach is 30.7 times faster at average. In addition, pros and cons of the proposed approach and future work directions are indicated.**

I. Introduction

Network security is one of the most important tasks that are solved by scientists and developers all over the world [1], [2]. The key features of this information security direction are the diversity of threats and a wide range of security requirements [3]. Moreover, it is necessary to perform a risk analysis of protected systems regularly, because new vulnerabilities are found daily [4].

One of the most effective approaches for network security analysis is multistep attack modelling, which is based on the vulnerabilities in open databases [5]. Currently, the most popular vulnerability database is NVD [6]. It contains 199996 vulnerabilities in the CVE format [7], while each vulnerability has metrics in the CVSS format of the second, third, or both versions [8], [9]. Those metrics are most often used to perform risk analysis and evaluate if the host of the network can be compromised by the intruder with certain parameters [10].

The modelling helps to take into account not only separate hosts of the network, but possible attack paths of the intruder based on the connections between them [11]. It also helps to evaluate the security of the whole network based on the security of its hosts, as well as to find security bottlenecks [12]. One of the possible ways to perform such analysis is to construct attack graphs [13]. In those graphs, nodes represent hosts, while connections between them are defined not only by the availability of the network connection, but also by the possibility for intruders to compromise those hosts [14].

The main issue of attack graph approaches is that it is challenging for them to work in real or near-real time because the state of the network is constantly changing, while the number of vulnerabilities is constantly growing [15]. Thus, any solution that improves the efficiency of the attack graphs construction has high relevance and is welcome for the area [16].

In this work, we attempted to improve the efficiency of the host attack graph construction through the categorization of the CVEs. The novelty of the proposed approach lies in the categorization of all available CVEs into 24 categories in accordance with their access vector, initial, and obtained access rights. After that, instead of vulnerabilities, only their categories are used for the construction of attack graphs of each host. It helps to improve the efficiency of this process for every host of the network, while those computations can be done for each host separately.

The paper is organised as follows. In Section II, the state of the art in vulnerability categorization and multistep attack modelling is considered. Section III describes the new approach to vulnerability categorization. An original approach for multistep attack modelling that is based on the approach for vulnerability categorization is presented In Section IV. Section V contains the experimental evaluation of the developed approaches. In Section VI, the advantages and disadvantages of the presented approaches are considered. Section VII contains general conclusions and future work directions.

II. Related work

In this section, examples of the approaches for vulnerability categorization and multistep attack modelling are analysed in detail. Let us consider the approaches for vulnerability categorization first.

In [17], a systematic mapping study of cybersecurity threats and vulnerabilities is done. The authors identified the most important security vulnerabilities and the frequency of their occurrence. In total, 78 studies were analysed and most of them were lacking empirical validation and real implementation. Moreover, most of the analysed papers were targeted at phishing, denial-of-service, and malware.

The approach for the automatic generation of summaries of daily posted vulnerabilities and their categorization is presented in [18]. The authors assessed their approach on a set of 3369 labelled CVEs. After that, the results of the approach were evaluated by 15 master students and 4 security experts. The results showed that such summaries are useful for analysts during the vulnerability assessment.

An approach using text classification in order to identify CVEs that can be mapped to CWEs and CAPECs is proposed in the master thesis [19]. It is done because not all CVEs are linked with related CWEs in open databases or their CWE entries are too generic – for example, NVD-CWE-Other. As the results, the author achieved 90% accuracy among 111 384 CVEs on a 10-fold cross-validation.

In [20] the CVE-based classification of vulnerable IoT systems is provided. Authors analysed CVEs that are available for IoT devices in open databases. After that, an SVM algorithm was used to classify CVEs in accordance with classification of IoT systems. The goal was to describe IoT device vulnerabilities of different applications: home, industry, mobile controllers and networking. The authors plan to use this classification to recognize vulnerable IoT devices.

Let us consider the approaches for multistep attack modelling in more detail.

It is proposed to use attack graphs in the same way they are already used in computer networks to analyse vulnerabilities in microservice-based systems in [21]. To make it possible, authors relate microservices to network nodes and automatically generate attack graphs to make it possible. Those graphs help to identify and analyse possible attack paths in microservice-based container networks.

Work [22] is devoted to identifying and evaluating network security threats based on the attack graphs. The authors used security equipment performance data and CVSS data to generate the probabilistic attack graph model and obtain the network security index. The experimental results showed that the model is feasible and effective.

The method for automatic analysis of complex attack graphs both in microservices-based and multi-cloud infrastructures is presented in [23]. In the developed method, microservices, virtual system states, and cloud services are represented as graph nodes. Authors use prioritisation algorithms that use mathematical graph series and group clustering to make calculations more efficient. The main features of the proposed solution are as follows: analysis of the impact of system states on the ecosystem; analysis of the overall risk to the ecosystem of system states, vulnerabilities, and configurations; consideration of every potential sub-attack path and subliminal path on an attack graph.

In [24] an attack graph-based alert correlation approach is proposed. Firstly, the authors used the MulVAL toolkit to generate attack graphs. It uses information about known vulnerabilities and network connectivity as input data. After that, the initial security state is mapped to this attack graph based on the available security alerts. Moreover, attack sequences are outputted from the set of mapped alerts to reflect the initial attack paths, while similar sequences are clustered together to obtain attack scenarios. In the end, broken attack scenarios are merged by detecting unreported true negative alerts.

An artificial intelligence-based tool for automatic generation, updating, and refining attack graphs is presented in [25]. This tool uses textual descriptions of vulnerabilities to generate attack graphs automatically. In addition, the authors described the methodology to incrementally update attack graphs when the system changes. Moreover, the developed tool can reuse attack graphs during the generation of a network of networks, and join them together to create larger attack graphs.

III. Vulnerabilities categorization

The approach for vulnerabilities categorization is based on the second and third versions of the CVSS metrics of CVEs. And among all possible CVSS metrics, for the categorization it was decided to use such metrics as *access vector*, *privileges required*, and *obtained privileges*. Possible values of those metrics in different versions of the CVSS notations are presented in Table I.

TABLE I. CVSS metrics for vulnerabilities categorization

		CVSS v2	CVSS v3
Access vector		LOCAL ADJACENT NETWORK NETWORK	PHYSICAL LOCAL ADJACENT NETWORK NETWORK
Privileges required			NONE LOW HIGH
Impact	**Confidentiality**	NONE PARTIAL COMPLETE	NONE LOW HIGH
	Integrity	NONE PARTIAL COMPLETE	NONE LOW HIGH
	Availability	NONE PARTIAL COMPLETE	NONE LOW HIGH
Obtained privileges	**ALL**	TRUE FALSE	
	USER	TRUE FALSE	
	OTHER	TRUE FALSE	

It can be noted that the LOCAL access vector from CVSS v2 was divided into PHYSICAL and LOCAL access vectors in CVSS v3. Moreover, in CVSS v2 there are no privileges required metric, while in CVSS v3 – there are no obtained privileges metric. Moreover, our analysis showed that there are 199996 CVEs in NVD, while 173952 have v2 metrics and 115651 have v3 metrics. Both metrics are available for 100581 CVEs only. It means that without any assumptions,

the approach would miss more than half of available CVEs. Note that those numbers were last updated 03/11/2023.

Thus, it was decided to make the following assumptions:

- *privileges required* = *NONE* for CVEs with CVSS v2 metrics only;
- *obtained privileges* = *OTHER* for CVEs with CVSS v3 metrics only.

In addition, it was decided to change the descriptions and possible values of *privileges required* and *obtained privileges* metrics. Values for *privileges required* in the developed approach are as follows:

- NONE: CVSS NONE;
- USER: CVSS LOW;
- ADMIN: CVSS HIGH.

And for *obtained privileges*:

- NONE: CVSS ALL and USER are equal to FALSE, OTHER can be any;
- USER: CVSS USER is equal to TRUE, ALL and OTHER – FALSE;
- ADMIN: CVSS ALL is equal to TRUE, USER and OTHER – FALSE.

Based on the possible values of such metrics as *access vector* (PHYSICAL, LOCAL, ADJACENT NETWORK, NETWORK), *privileges required* (NONE, USER, ADMIN), and *obtained privileges* (NONE, USER, ADMIN), all CVEs were divided into 24 categories, description of which is available in Table II.

TABLE II. CVE CATEGORIES DESCRIPTION

	Description
C111	access PHYSICAL, required NONE, obtained NONE/OTHER
C112	access LOCAL, required NONE, obtained NONE/OTHER
C113	access ADJACENT_NETWORK, required NONE, obtained NONE/OTHER
C114	access NETWORK, required NONE, obtained NONE/OTHER
C121	access PHYSICAL, required NONE, obtained USER
C122	access LOCAL, required NONE, obtained USER
C123	access ADJACENT_NETWORK, required NONE, obtained USER
C124	access NETWORK, required NONE, obtained USER
C221	access PHYSICAL, required LOW, obtained NONE/OTHER/USER
C222	access LOCAL, required LOW, obtained NONE/OTHER/USER
C223	access ADJACENT_NETWORK, required LOW, obtained NONE/OTHER/USER
C224	access NETWORK, required LOW, obtained NONE/OTHER/USER
C131	access PHYSICAL, required NONE, obtained ALL
C132	access LOCAL, required NONE, obtained ALL
C133	access ADJACENT_NETWORK, required NONE, obtained ALL
C134	access NETWORK, required NONE, obtained ALL
C231	access PHYSICAL, required LOW, obtained ALL
C232	access LOCAL, required LOW, obtained ALL
C233	access ADJACENT_NETWORK, required LOW, obtained ALL
C234	access NETWORK, required LOW, obtained ALL
C331	access PHYSICAL, required HIGH
C332	access LOCAL, required HIGH
C333	access ADJACENT_NETWORK, required HIGH
C334	access NETWORK, required HIGH

It is important to note that each category of CVEs has its title encoded in the CXXX format, the definition of which is presented in Fig. 1.

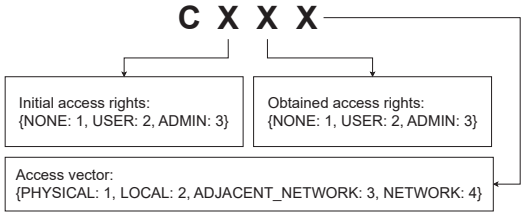

Fig. 1. Encoding of CVE categories

IV. ATTACK MODELLING

The developed approach for multistep attack modelling considers each host of the analysed network that can be initially reached by the intruder (network border). And if it is possible for the intruder to receive ADMIN access rights on the host (such host is considered as compromised), then any host that is connected with the compromised host can be as well reached by the intruder and become the next step of the multistep attack (generation of the attack graph). Thus, in this section it was decided to focus on the detailed description of host attack graphs, while network attack graph construction is typical and based on the host graphs analysis.

For the generation of host attack graphs, it was decided to divide the possible states of the intruder on the analysed host into 12 states in accordance with the intruders' access rights and vector, see Table III.

TABLE III. INTRUDER STATES ON THE HOST

		Access rights		
		NONE	**USER**	**ADMIN**
Access vector	**PHYSICAL**	S1	S5	S9
	LOCAL	S2	S6	S10
	ADJACENT NETWORK	S3	S7	S11
	NETWORK	S4	S8	S12

Transitions between those states are possible based on the exploitation of CVEs. The impact of different CVE categories on transitions of the intruder between its states on the analysed host are presented in Fig. 2.

Moreover, in the developed approach, it is assumed that if the intruder obtains USER or ADMIN access rights, then other access vectors that are different from the initial one become available to the intruder, see Fig. 3.

Availability of transitions between access vectors defines additional connections between the defined states of the intruder. The developed approach considers all possible transitions between those states as the host attack graph. This attack graph is presented in Fig. 4.

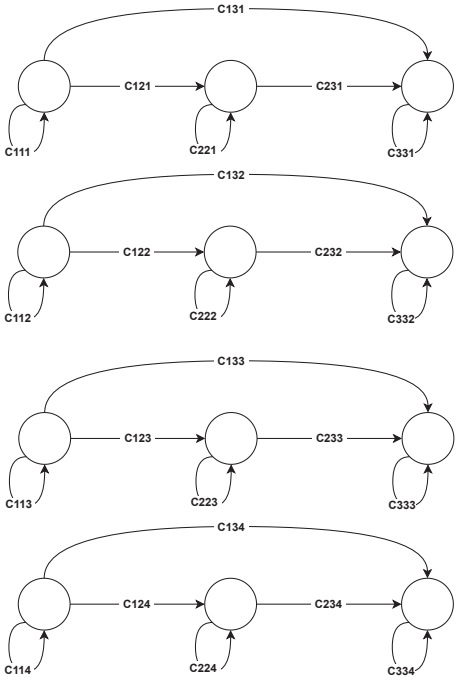

Fig. 2. Transitions between intruder states

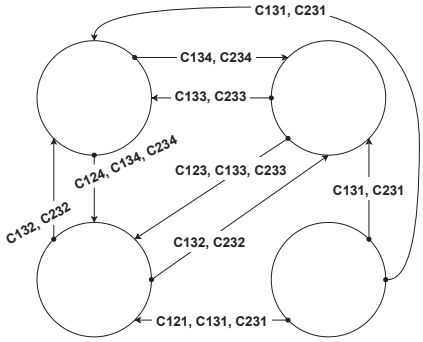

Fig. 3. Transitions between access vectors

For each analysed network host, the goal of the approach is to define if the intruder can reach either S9, S10, S11 or S12 state based on the exploitation of CVEs of different categories. The main benefit of using CVE categories instead of CVEs is the lower amount of computations required for each host's analysis. To confirm this, let us consider a small example.

As input data, the developed approach received a JSON-based data structure with the host's description. This description contains 10 random CPEs and CVEs. Let us assume that CPEs represent host configuration, while CVEs result from vulnerability scanning.

```
host = {
    "cpes": {
        "cpe:2.3:a:tor_browser_launcher_project:tor_browser_launcher
            :0.2.2:*:*:*:*:*:*:*",
        "cpe:2.3:o:hp:openvms_vax:6.0:*:*:*:*:*:*:*",
        "cpe:2.3:a:apache:wicket:7.11.0:*:*:*:*:*:*:*",
        "cpe:2.3:a:proxyman:proxyman:1.3.3:*:*:*:*:mac_os:*:*",
        "cpe:2.3:a:spip:spip:2.0.8:*:*:*:*:*:*:*",
        "cpe:2.3:a:novell:zenworks_configuration_management
            :10.3.1:*:*:*:*:*:*:*",
        "cpe:2.3:a:mozilla:thunderbird:45.3.0:*:*:*:*:*:*:*",
        "cpe:2.3:a:phore:phore:0.9.13.0:*:*:*:*:*:*:*",
        "cpe:2.3:h:emc:vplex_metro:-:*:*:*:*:*:*:*",
        "cpe:2.3:a:hp:hppscanto:0.105.89:*:*:*:*:*:*:*"
    },
    "cves": {
        "CVE-2010-4914", "CVE-2000-0515", "CVE-2008-0626", "CVE-2020-35458",
        "CVE-2021-34320", "CVE-2012-0065", "CVE-2020-24609", "CVE-2019-15519",
        "CVE-2008-4426", "CVE-2004-0657"
    }
}
```

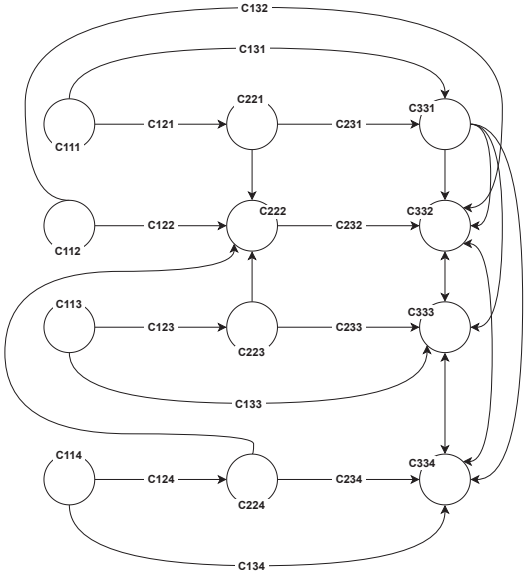

Fig. 4. Host attack graph

Based on CPEs analysis, the final list of the host CVEs is as follows:

```
"cves": {
    "CVE-2010-4914", "CVE-2000-0515", "CVE-2013-2118", "CVE-2016-3154",
    "CVE-2020-35458", "CVE-2019-15519", "CVE-2013-6346", "CVE-2009-3041",
    "CVE-2008-0626", "CVE-2021-34320", "CVE-2013-1079", "CVE-2020-24609",
    "CVE-2013-6347", "CVE-2004-0657", "CVE-2013-6344", "CVE-2013-4555",
    "CVE-2013-4556", "CVE-2013-6345", "CVE-2012-0065", "CVE-2013-3278",
    "CVE-2012-2223", "CVE-2013-7303", "CVE-2010-4229", "CVE-2016-3153",
    "CVE-2008-4426"
}
```

Thus, it is required to analyse 25 CVEs to understand if the host can be compromised by the intruder, while only 3 CVE categories are representing all those vulnerabilities (C134, C114 and C112):

```
cve_categories = {
    "C134": {"CVE-2000-0515"},
    "C114": {
    "CVE-2010-4914", "CVE-2013-2118", "CVE-2016-3154", "CVE-2020-35458",
        "CVE-2019-15519", "CVE-2009-3041", "CVE-2013-6346", "CVE-2020-24609",
        "CVE-2013-1079", "CVE-2013-6347", "CVE-2004-0657", "CVE-2013-4555",
        "CVE-2013-6344", "CVE-2013-4556", "CVE-2013-6345", "CVE-2012-2223",
        "CVE-2013-7303", "CVE-2010-4229", "CVE-2016-3153", "CVE-2008-4426"
    },
    "C112": {"CVE-2013-3278", "CVE-2021-34320", "CVE-2012-0065"}
}
```

V. Experimental evaluation

For the experimental evaluation of the approach, it was decided to generate hosts with 10 random CVEs and CPEs, from 10 hosts to 250, while 10 hosts are added on each step. After that, for each host it is analysed if it is vulnerable based on the list of CVEs and based on the list of their categories. Note that each step of the host generation was done for 5 times and average time consumption was taken as a result to limit the effect of randomness.

CVEs and CPEs were stored locally in the PostgreSQL database. Firstly, the list of CVEs was extended in accordance with CPEs for both approaches. After that, for the default approach (analysis of CVEs list), CVSS metrics were extracted and analysed for each CVE of the host, until a combination of CVEs that gives admin access rights was not found (such host is considered vulnerable). If such a combination of CVEs was not found, then the host is not considered vulnerable. For the categories approach (analysis of CVE categories), firstly, the lists of CVEs was transformed into the list of CVE categories (time required for this operation is included into the total time of the analysis). After that, those categories were analysed until a combination of CVE categories that gives admin access rights was not found (such host is considered vulnerable). If such a combination of CVE categories was not found, then the host is not considered vulnerable.

All experiments were done in the PyCharm 2022.3.3 environment on PC with the following specifications: Windows 11 21H2 OS, Intel Core i7-10700KF 3.80GHz CPU and 32.0 GB of RAM. Received results are presented in Table IV. A visual representation of the approaches' comparison using a bar chart is shown in Fig. 5.

TABLE I. EXPERIMENTAL EVALUATION: 10 RANDOM CVES AND CPES

Hosts	Approach		Difference	
	Default	Categories	In seconds	In percentages
10	0.7667	0.0485	0.7182	1580.8247
50	2.5270	0.1876	2.3394	1347.0149
100	5.0194	0.3777	4.6417	1328.9383
150	7.9008	0.5961	7.3047	1325.4152
200	10.2307	0.7692	9.4615	1330.0442
250	13.0961	0.9628	12.1333	1360.2098

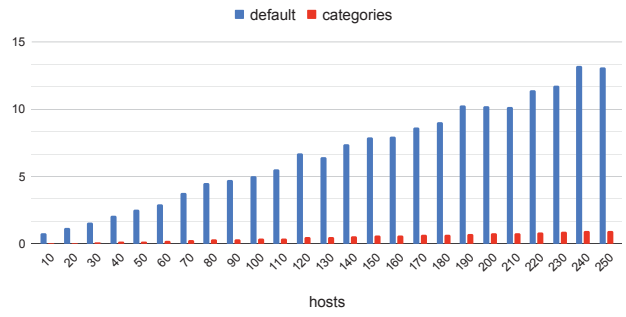

Fig. 5. Approaches comparison: 10 CVEs and CPEs

After that, it was decided to check how the number of random CVEs and CPEs per host affects the testing results. For the first experiment, hosts with 10 random CVEs and CPEs were generated, thus it was decided to increase these numbers to 50. The obtained results are available in Table V. A visual representation of the approaches' comparison using a bar chart is shown in Fig. 6.

TABLE V. EXPERIMENTAL EVALUATION: 50 RANDOM CVES AND CPES

Hosts	Approach		Difference	
	Default	Categories	In seconds	In percentages
10	2.6163	0.1310	2.4853	1997.1756
50	12.8356	0.5517	12.2839	2326.5543
100	24.5391	1.0802	23.4589	2271.7182
150	38.1961	1.6711	36.5250	2285.6861
200	51.6923	2.2646	49.4277	2282.6239
250	59.2813	2.5890	56.6923	2289.7374

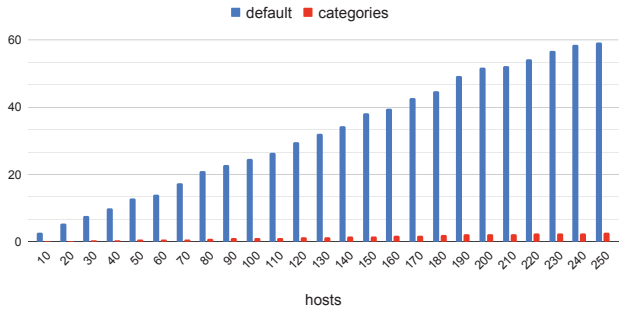

Fig. 6. Approaches comparison: 50 CVEs and CPEs

The analysis of the obtained results allowed us to put forward an assumption about the linear nature of the relationship between the time spent on the analysis of hosts in accordance with their number, as well as the number of CVEs and CPEs available for each host. Thus, it was decided to implement an additional experiment, where the same calculations are done for the fixed number of hosts equal to 100, but the number of random CVEs and CPEs for each host was changing from 10 to 100 with the step equal to 10, see Table VI.

TABLE VI. EXPERIMENTAL EVALUATION: LINEAR RELATIONSHIP

CVEs and CPEs	Approach		Difference	
	Default	Categories	In seconds	In percentages
10	5.0194	0.3777	4.6417	1328.9383
20	16.6813	0.8306	15.8507	2008.3434
30	20.5712	0.9482	19.6230	2169.5001
40	23.2357	1.0555	22.1802	2201.3927
50	24.5391	1.0802	23.4589	2271.7182
60	26.1366	1.1347	25.0019	2303.3930
70	32.0206	1.3029	30.7177	2457.6406
80	31.6881	1.2979	30.3902	2441.4901
90	36.2152	1.4419	34.7733	2511.6305
100	38.3429	1.5002	36.8427	2555.8526

The analysis of the obtained time values allowed us to calculate a linear function for time required for experiment and for time difference between them corresponding to the obtained values using the least squares method, see Table VII. Note that the linear function is represented as $f(x) = mx + b$, thus m and b are representing the corresponding coefficients.

TABLE VII. EXPERIMENTAL EVALUATION: LINEAR COEFFICIENTS

Hosts	CVEs and CPEs	Approach						Default / Categories ratio
		Default		Categories		Time difference		
		m	*b*	*m*	*b*	*m*	*b*	
100	10..100	0.3153	8.1055	0.0103	0.5328	0.0477	0.0497	30.7315
10..250	10	0.0516	0.0528	0.0038	0.0031	0.2353	0.3113	13.4081
10..250	50	0.2460	0.3370	0.0107	0.0256	0.3050	7.5727	23.0083

Analysis of the experiment results showed that the categories-based approach shows constant growth of the consumed time difference with increase of hosts and vulnerability amounts (the time difference function has positive m). It proves that if the numbers of vulnerabilities and hosts continue to grow, the performance gains using the proposed approach will also steadily increase.

VI. DISCUSSION

As the result of vulnerabilities categorization, each CVE from NVD received a category and none of CVEs received more than one category. It means that developed categorization is complete, while adjustments will be required only after the announcement of the new version CVSS metrics.

The distribution of CVEs between categories is shown in Table VIII. The value without brackets represents the number of vulnerabilities that were categorized in accordance with the assumptions introduced in Section III. The number of vulnerabilities that were categorized without those assumptions is shown in brackets. Note that those numbers were last updated 03/11/2023.

TABLE VIII. DISTRIBUTION OF CVES BETWEEN CATEGORIES

C111 1042 (931)	**C112** 17880 (9643)	**C113** 3792 (1606)	**C114** 116894 (53015)
C121 0 (0)	**C122** 370 (0)	**C123** 8 (0)	**C124** 2026 (4)
C221 132 (125)	**C222** 14183 (12199)	**C223** 303 (269)	**C224** 18128 (15211)
C131 2 (2)	**C132** 1908 (6)	**C133** 11 (0)	**C134** 3224 (51)
C231 0 (0)	**C232** 47 (47)	**C233** 2 (2)	**C234** 30 (30)
C331 74 (68)	**C332** 2424 (1982)	**C333** 310 (293)	**C334** 6232 (5097)

It can be noted that in accordance with the assumptions made, it became possible to increase the number of CVEs for some categories, while other categories stayed with the same number of CVEs. It is understandable because of the nature of the assumptions, where the *privileges required* metric was set equal to NONE for CVEs with CVSS v2 metrics only, and the *obtained privileges* metric was set equal to OTHER for CVEs with CVSS v3 metrics only. In future work, it is planned to use a more complex approach for mapping CVSS v2 and v3 metrics, so their distribution between those categories would be more justified.

The colour of the cells highlights the categories of CVEs that do not have any CVEs (red) and those with only a tiny number of CVEs (orange). It means that some transitions between intruder states in the host attack graph are impossible or have a low probability, see Fig. 7.

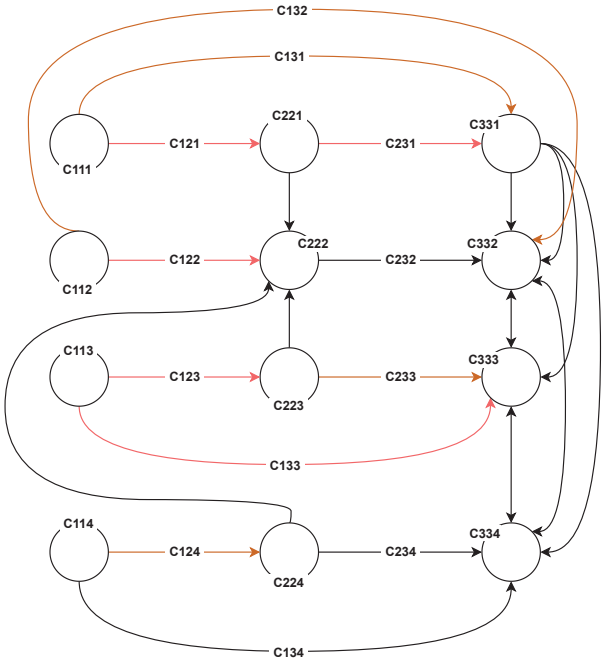

Fig. 7. Possibility of transitions between intruder states

Red lines show transitions that are not possible – there are no CVEs in C121 and C231 categories, while orange lines show transitions with low probability – C131, C123, C233, and C133 categories have low numbers of CVEs.

VII. CONCLUSION

The new approach to vulnerability categorization for fast multistep attack modelling is presented in the paper. Its novelty lies in the categorization of all available CVEs into 24 categories based on their access vector, initial and obtained access rights. After that, instead of vulnerabilities, only their categories are used for constructing attack graphs of each host of the analysed system. It helps to improve the efficiency of this process for every host of the network, while those computations can be done for each host separately. Moreover, some assumptions were introduced to integrate second and third versions of CVSS vulnerabilities descriptions and allow transitions of the attacker between different access vectors.

In the end, 189022 CVEs with either CVSS version 2, 3, or both were categorized. Those categories define transitions between 12 states of the host, 4 of which define availability of administrator access rights. Achievement of such access rights by the intruder is considered as a possibility to attack other hosts and, thus, to continue performing a malicious activity in the analysed network.

Several experiments were implemented to compare the suggested approach with the default one. Firstly, it was decided to generate hosts with 10 random CVEs and CPEs, from 10 hosts to 250, while 10 hosts are added on each step. Moreover, each step of the host generation was done for 5 times and average time consumption was taken as a result to limit the effect of randomness. After that, the same experiment was redone, but with 50 random CVEs and CPEs for each host. The results showed that the suggested approach is 13.4 times faster at average for 10 CVEs and CPEs, while 23.0 times faster for 50 CVEs and CPEs.

In addition, it was decided to implement an additional experiment, where the same calculations are done for the fixed number of hosts equal to 100, but the number of random CVEs and CPEs for each host was changed from 10 to 100 with the step equal to 10. Once again, each step of the host generation was done for 5 times and average time consumption was taken as a result to limit the effect of randomness. This experiment showed that the suggested approach is 30.7 times faster at average for 100 hosts.

Summarising the experiments, the approximate function as a linear equation for consumed time difference was calculated. For all experiments, the linear function coefficients are positive. It proves the superiority of the categories-based approach over the default one for the analysis of vulnerable computer networks of any size.

During the future work, we plan to use the presented results in the prototype to analyse the possibility and impact of cyber-physical attacks on critical infrastructure facilities that use Internet of Things devices. It should be noticed that the attack analysis has a dynamic nature because of the variable nature of the intruder's parameters, available network connections and security tools, as well as countermeasures.

ACKNOWLEDGMENT

The study was supported by the grant of the Russian Science Foundation No. 22-71-00107, https://rscf.ru/en/project/22-71-00107.

REFERENCES

[1] R. Ferdiana *et al.*, "A systematic literature review of intrusion detection system for network security: Research trends, datasets and methods," in *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE, 2020, pp. 1–6.

[2] D. S. Levshun, D. A. Gaifulina, A. A. Chechulin, and I. V. Kotenko, "Problematic issues of information security of cyber-physical systems," *Informatics and automation*, vol. 19, no. 5, pp. 1050–1088, 2020.

[3] Y. Li, G.-q. Huang, C.-z. Wang, and Y.-c. Li, "Analysis framework of network security situational awareness and comparison of implementation methods," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–32, 2019.

[4] E. Doynikova, E. Novikova, I. Murenin, M. Kolomeec, D. Gaifulina, O. Tushkanova, D. Levshun, A. Meleshko, and I. Kotenko, "Security measuring system for iot devices," in *Computer Security. ESORICS 2021 International Workshops: CyberICPS, SECPRE, ADIoT, SPOSE, CPS4CIP, and CDT&SECOMANE, Darmstadt, Germany, October 4–8, 2021, Revised Selected Papers*. Springer, 2022, pp. 256–275.

[5] K. Izrailov, D. Levshun, I. Kotenko, and A. Chechulin, "Classification and analysis of vulnerabilities in mobile device infrastructure interfaces," in *Mobile Internet Security: 5th International Symposium, MobiSec 2021, Jeju Island, South Korea, October 7–9, 2021, Revised Selected Papers*. Springer, 2022, pp. 301–319.

[6] M. U. Aksu, K. Bicakci, M. H. Dilek, A. M. Ozbayoglu, and E. ı. Tatli, "Automated generation of attack graphs using nvd," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, 2018, pp. 135–142.

[7] V. Pham and T. Dang, "Cvexplorer: Multidimensional visualization for common vulnerabilities and exposures," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1296–1301.

[8] C. Elbaz, L. Rilling, and C. Morin, "Fighting n-day vulnerabilities with automated cvss vector prediction at disclosure," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, pp. 1–10.

[9] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A survey of iiot protocols: A measure of vulnerability risk analysis based on cvss," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–53, 2020.

[10] D. Ivanov, M. Kalinin, V. Krundyshev, and E. Orel, "Automatic security management of smart infrastructures using attack graph and risk analysis," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020, pp. 295–300.

[11] D. Levshun, Y. Bakhtin, A. Chechulin, and I. Kotenko, "Analysis of attack actions on the railway infrastructure based on the integrated model," in *Mobile Internet Security: 4th International Symposium, MobiSec 2019, Taichung, Taiwan, October 17–19, 2019, Revised Selected Papers*. Springer, 2020, pp. 145–162.

[12] M. Fuentes-García, J. Camacho, and G. Maciá-Fernández, "Present and future of network security monitoring," *IEEE Access*, vol. 9, pp. 112 744–112 760, 2021.

[13] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Computer Science Review*, vol. 35, p. 100219, 2020.

[14] X. Liu, "A network attack path prediction method using attack graph," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–8, 2020.

[15] A. Fedorchenko, I. V. Kotenko, and A. Chechulin, "Integrated repository of security information for network security evaluation." *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 6, no. 2, pp. 41–57, 2015.

[16] I. Kotenko, E. Doynikova, and A. Chechulin, "Security metrics based on attack graphs for the olympic games scenario," in *2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. IEEE, 2014, pp. 561–568.

[17] M. Humayun, M. Niazi, N. Jhanjhi, M. Alshayeb, and S. Mahmood, "Cyber security threats and vulnerabilities: a systematic mapping study," *Arabian Journal for Science and Engineering*, vol. 45, pp. 3171–3189, 2020.

[18] E. R. Russo, A. Di Sorbo, C. A. Visaggio, and G. Canfora, "Summarizing vulnerabilities' descriptions to support experts during vulnerability assessment activities," *Journal of Systems and Software*, vol. 156, pp. 84–99, 2019.

[19] T. Giannakopoulos, "Threat categorization on cve descriptions using text classification," Master's thesis, Πανεπιστήμιο Πειραιώς, 2022.

[20] G. J. Blinowski and P. Piotrowski, "Cve based classification of vulnerable iot systems," in *Theory and Applications of Dependable Computer Systems: Proceedings of the Fifteenth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX, June 29–July 3, 2020, Brunów, Poland 15*. Springer, 2020, pp. 82–93.

[21] A. Ibrahim, S. Bozhinoski, and A. Pretschner, "Attack graph generation for microservice architecture," in *Proceedings of the 34th ACM/SIGAPP symposium on applied computing*, 2019, pp. 1235–1242.

[22] Y. Cui, J. Li, W. Zhao, and C. Luan, "Research on network security quantitative model based on probabilistic attack graph," in *ITM Web of Conferences*, vol. 24. EDP Sciences, 2019, p. 02003.

[23] G. Stergiopoulos, P. Dedousis, and D. Gritzalis, "Automatic analysis of attack graphs for risk mitigation and prioritization on large-scale and complex networks in industry 4.0," *International Journal of Information Security*, pp. 1–23, 2022.

[24] H. Hu, J. Liu, Y. Zhang, Y. Liu, X. Xu, and J. Tan, "Attack scenario reconstruction approach using attack graph and alert data mining," *Journal of Information Security and Applications*, vol. 54, p. 102522, 2020.

[25] B. Bezawada, I. Ray, and K. Tiwary, "Agbuilder: an ai tool for automated attack graph building, analysis, and refinement," in *Data and Applications Security and Privacy XXXIII: 33rd Annual IFIP WG 11.3 Conference, DBSec 2019, Charleston, SC, USA, July 15–17, 2019, Proceedings 33*. Springer, 2019, pp. 23–42.