1 EXAMPLE VULNERABILITY

This supplemental material gives the details of the running example in our paper.

The running example is a vulnerability item (CVE-2023-50723), this CVE suggests that missing escaping in the code for displaying sections in the administration interface allowed those who could edit arbitrary wiki pages to gain programming rights, threatening the system's confidentiality, integrity, and availability. The description is as follows:

> *XWiki Platform is a generic wiki platform. Starting in 2.3 and prior to versions 14.10.15, 15.5.2, and 15.7-rc-1, anyone who can edit an arbitrary wiki page in an XWiki installation can gain programming right through several cases of missing escaping in the code for displaying sections in the administration interface. This impacts the confidentiality, integrity and availability of the whole XWiki installation. Normally, all users are allowed to edit their own user profile so this should be exploitable by all users of the XWiki instance. This has been fixed in XWiki 14.10.15, 15.5.2 and 15.7RC1. The patches can be manually applied to the 'XWiki.ConfigurableClassMacros' and 'XWiki.ConfigurableClass' pages.*

Its four patches (positive samples) are as follows:

- ***0f367aa****ae4e0696f61cf5a67a75edd27d1d16db6*,
- ***1157c1e****cea395aac7f64cd8a6f484b1225416dc7*,
- ***749f6ae****e1bfbcf191c3734ea0aa9eba3aa63240e*,
- ***bd82be9****36c21b65dee367d558e3050b9b6995713*.

We also randomly select 1,500 code commits that are not relevant to the fixing of the vulnerability in the same repository as negative samples. These 1,504 code commits are candidate code commits.

2 RESULT OF SHIP

As mentioned in the original paper, SHIP involves three phases, i.e., initial ranking, commits linkage prediction, and commit group forming and ranking. Here we give a step-by-step elaboration on these three phases based on the example vulnerability item.

2.1 Phase-1: Initial Ranking

SHIP extracts rule-based and semantic features for all candidate code commits, and predicts their relevance scores to the vulnerability, to produce an initial ranking list of candidate code commits. The result is shown in Table 1 in this document, in which "Label: 1" denotes that the code commit is the patch of CVE-2023-50723 and "Label: 0" otherwise, columns "Prediction" and "Rank" give the predicted relevance scores and the ranking, respectively. Here we provide the result of code commits "*0f367aa*", "*1157c1e*", "*749f6ae*", and "*bd82be9*" (i.e., the four patches of this vulnerability). The complete initial ranking list is given in **"Phase1-Output.csv"** (in the same directory as this PDF file).

Table 1: Phase-1 Output (Partial)

Commit	Prediction	Label	Rank
bd82be9	1	1	1
0f367aa	0.9999845	1	5
1157c1e	0.99850816	1	8
749f6ae	0.002294838	1	22
……			

2.2 Phase-2: Commits Linkage Prediction

Top-*k* candidate code commits (here we take *k* as 50 as an example) in the initial ranking list will be input to this phase to predict their interrelationship. The result is shown in Table 2 in this document, in which "Label1: 1" and "Labe2: 1" denote that Commit1 and Commit2 are the patches of CVE-2023-50723, respectively, and "0" otherwise. "Label: 1" denotes that this pair of code commits is regarded as relevant in training, and "0" otherwise. Column "Prediction" is the predicted relevance score of this pair. Here we give the result of pairs "*bd82be9-0f367aa*", "*0f367aa-749f6ae*", "*0f367aa-1157c1e*", "*bd82be9-1157c1e*", "*bd82be9-749f6ae*", and "*1157c1e-749f6ae*" (the interrelationships between four real patches of CVE-2023-50723) in Table 2 in this document. The complete result of the C_2^{50} pairs is given in **"Phase2-Output.csv"** (in the same directory as this PDF file).

Table 2: Phase-2 Output (Partial)

Commit1	Commit2	Prediction	Label	Label1	Label2
bd82be9	*0f367aa*	0.9829762	1	1	1
0f367aa	*749f6ae*	0.9986405	1	1	1
0f367aa	*1157c1e*	0.998857	1	1	1
bd82be9	*1157c1e*	0.99911064	1	1	1
bd82be9	*749f6ae*	0.9992873	1	1	1
1157c1e	*749f6ae*	0.9998568	1	1	1
.....					

2.3 Phase-3: Commit Group Forming and Ranking

This phase aims to generate groups of candidate code commits by determining whether a pair of commits should be connected. According to the description of Section III-C of the original paper, the parameter θ determines whether two code commits can be connected, here we take the value of θ as 0.9 as an example. That is to say, the pairs of code commits with a relevance score higher than 0.9 will be connected, and thus one or more maximal connected subgraphs (MCS) can be formed. Based on the relevance scores of pairs of code commits given in Phase-2, a total of 45 MCSs are formed, as shown in column "Group" in Table 3 in this document (those code commits having the same "Group" number denote they are in the same MCS). In this table, "Label: 1" denotes that the code commit is the patch of CVE-2023-50723 and "Label: 0" otherwise. Here we give the result of the first 10 MCS, the complete result of all 45 groups is given in **"Phase3-Output.csv"** (in the same directory as this PDF file).

It can be seen that code commits *0f367aa*, *1157c1e*, *749f6ae*, and *bd82be9* (i.e., the four real patches of CVE-2023-50723) form an MCS because no other code commit has a relevance score greater than 0.9 with any of them. SHIP then produces proxy vectors for all MCSs, gets their relevance scores with the vulnerability, and accordingly ranks them, as shown in columns "Prediction" and "Rank" in Table 3 in this document. The ninth MCS containing and only containing the four real patches is ranked at the first position, which will be determined as the *patch group* and output.

Table 3: Phase-3 Output (Partial)

Commit	Label	Group	Prediction	Rank
e995f4c	0	1	1.932448e-06	23
1314958	0	2	7.781783e-13	43
5ae7a0e	0	3	4.111797e-07	30
79418dd	0	4	1.0217896e-05	17
9efa051	0	5	4.7453623e-06	19
b2c2a55	0	6	1.6026002e-06	24
728e4b7	0	7	4.3778805e-06	21
499902c	0	8	3.2412118e-11	40
749f6ae	1	9	0.99423003	1
0f367aa	1	9	0.99423003	1
1157c1e	1	9	0.99423003	1
bd82be9	1	9	0.99423003	1
bb462c2	0	10	2.2149186e-05	15
		.....		