[13]2020-Towards a Deep Learning Model for Vulnerability Detection on Web Application Variants-ICSTW-PHP--56篇参考文献

- 总结
  - 数据集----SARD，File，PHP，随机，7：3
  - 代码特征表示
    - 通过VLD转换为Opcode
  - **向量化**
    - 通过Vocabularycomposed of VLD opcodes来生产整数，也即numeric vector
    - i*5+j+1
    - 表1

TABLE I: Vocabulary composed of VLD opcodes. The index of the opcode in position i, j is given by $i * 5 + j + 1$.

i \ j	0	1	2	3	4
0	OOV	NOP	ADD	SUB	MULT
1	DIV	MOD	SL	SR	CONCAT
2	BW_OR	BW_AND	BW_XOR	BW_NOT	BOOL_NOT
3	BOOL_XOR	IS_IDENTICAL	IS_NOT_IDENTICAL	IS_EQUAL	IS_NOT_EQUAL
4	IS_SMALLER	IS_SMALLER_OR_EQUAL	CAST	QM_ASSIGN	ASSIGN_ADD
5	ASSIGN_SUB	ASSIGN_MUL	ASSIGN_DIV	ASSIGN_MOD	ASSIGN_SL
6	ASSIGN_SR	ASSIGN_CONCAT	ASSIGN_BW_OR	ASSIGN_BW_AND	ASSIGN_BW_XOR
7	PRE_INC	PRE_DEC	POST_INC	POST_DEC	ASSIGN
8	ASSIGN_REF	ECHO	GENERATOR_CREATE	JMP	JMPZ
9	JMPNZ	JMPZNZ	JMPZ_EX	JMPNZ_EX	CASE
10	CHECK_VAR	SEND_VAR_NO_REF_EX	MAKE_REF	BOOL	FAST_CONCAT
11	ROPE_INIT	ROPE_ADD	ROPE_END	BEGIN_SILENCE	END_SILENCE
12	INIT_FCALL_BY_NAME	DO_FCALL	INIT_FCALL	RETURN	RECV
13	RECV_INIT	SEND_VAL	SEND_VAR_EX	SEND_REF	NEW
14	INIT_NS_FCALL_BY_NAME	FREE	INIT_ARRAY	ADD_ARRAY_ELEMENT	INCLUDE_OR_EVAL
15	UNSET_VAR	UNSET_DIM	UNSET_OBJ	FE_RESET_R	FE_FETCH_R
16	EXIT	FETCH_R	FETCH_DIM_R	FETCH_OBJ_R	FETCH_W
17	FETCH_DIM_W	FETCH_OBJ_W	FETCH_RW	FETCH_DIM_RW	FETCH_OBJ_RW
18	FETCH_IS	FETCH_DIM_IS	FETCH_OBJ_IS	FETCH_FUNC_ARG	FETCH_DIM_FUNC_ARG
19	FETCH_OBJ_FUNC_ARG	FETCH_UNSET	FETCH_DIM_UNSET	FETCH_OBJ_UNSET	FETCH_LIST
20	FETCH_CONSTANT	GOTO	EXIT_STMT	EXT_FCALL_BEGIN	EXT_FCALL_END
21	EXT_NOP	TICKS	SEND_VAR_NO_REF	CATCH	THROW
22	FETCH_CLASS	CLONE	RETURN_BY_REF	INIT_METHOD_CALL	INIT_STATIC_METHOD_CALL
23	ISSET_ISEMTY_VAR	ISSET_ISEMPTY_DIM_OBJ	SEND_VAL_EX	SEND_VAR	INIT_USER_CALL
24	UNKNOWN[119]	SEND_USER	STRLEN	DEFINED	TYPE_CHECK
25	VERIFY_RETURN_TYPE	FE_RESET_RW	FE_FETCH_RW	FE_FREE	INIT_DYNAMIC_CALL
26	DO_ICALL	DO_UCALL	DO_FCALL_BY_NAME	PRE_INC_OBJ	PRE_DEC_OBJ
27	POST_INC_OBJ	POST_DEC_OBJ	ASSIGN_OBJ	OP_DATA	INSTANCEOF
28	DECLARE_CLASS	DECLARE_INHERITED_CLASS	DECLARE_FUNCTION	RAISE_ABSTRACT_ERROR	DECLARE_CONST
29	ADD_INTERFACE	VERIFY_INSTANCEOF	VERIFY_ABSTRACT_CLASS	ASSIGN_DIM	ISSET_ISEMPTY_PROP_OBJ
30	HADLE_EXCEPTION	USER_OPCODE	ASSERT_CHECK	JMP_SET	DECLARE_LAMBDA_FUNCTION
31	ADD_TRAIT	BINDTRAIS	SEPARATE	FETCH_CLASS_NAME	JMP_SET_VAR
32	DISCARD_EXCEPTION	YIELD	GENERATOR_RETURN	FAST_CALL	FAST_RET
33	RECV_VARIADIC	SEND_UNPACK	POW	ASSIGN_POW	BIND_GLOBAL
34	COALESCE	SPACESHIP	DECLARE_ANON_CLASS	DECLARE_ANON_INHERITED_CLASS	FETCH_STATIC_PROP_R
35	FETCH_STATIC_PROP_W	FETCH_STATIC_PROP_RW	FETCH_STATIC_PROP_IS	FETCH_STATIC_PROP_FUC_ARG	FETCH_STATIC_PROP_UNSET
36	UNSET_STATIC_PROP	ISSET_ISEMPTY_STATIC_PROP	FETCH_CLASSICAL_CONSTANT	BIND_LEXICAL	BIND_STATIC
37	FETCH_THIS	UNKNOWN[185]	ISSET_ISEMPTY_THIS	SWITCH_LONG	SWITCH_STRING
38	IN_ARRAY	COUNT	GET_CLASS	GET_CALLED_CLASS	GET_TYPE
39	FUNC_NUM_ARGS	FUNC_GET_ARGS	ISSET_EMPTY		

  - 模型
    - 总
      - DL 模型由多层构成。每一层都接收前一层的输出作为输入，并对输入应用一些转换。通过具有多个层，模型可以基于更简单和更广泛的概念（来自前几层）[24]学习更复杂和更有用的概念。
    - Embeddding
      - 将正整数，也即索引值，转换为固定大小的向量
    - N LSTM+N Dropout
      - 考虑每个操作码的上下文，也即保存顺序性
      - 减少过拟合
    - Dense
      - 1
        - 学习the slice和label之间的关系，将输入转换成一个具有相同形状的向量
      - 2
        - 对the slice进行分类并存在SQLi漏洞的概率，sigmoid函数[44]
  - 优化器+调参
    - 优化器

        - 三种常用的基于 SGD 的优化器
        - ADADELTA[25]
            - 它使用动态学习率 (lr) 计算每个维度
        - RMSProp[26]
            - 它使用重新缩放的梯度生成更新
        - ADAM[27]
            - 它使用梯度的运行平均值生成更新。
    - 调参
        - HS - HIDDEN SIZE,---学习能力相关
            - LSTM中HIDDEN SIZE的单元数
            - 拥有的单元越多，学习的内容也越多
        - δ - dropout rate,----减少过拟合
            - 它表示dropout rate，一个与dropout Layer相关的0到1之间的值，并对应于输入向量的每个条目变为0的概率。
        - NE - number of epochs---提高学习效果
            - 定义优化器更新参数的次数
            - 一般来说，该参数越高，模型学习效果越好。但是，我们需要平衡学习和泛化能力，不要让模型过拟合。
- 评价
    - 在训练集中对每个模型进行了3次10倍交叉验证
- 结果
    - 最佳模型达到了95%以上的准确率，正确率和召回率，且RMSProp优化器最佳
    - 表 5

TABLE V: Results of the accuracy, precision and recall for the various configurations analysed.

Optimizer	HS	δ	NE	Accuracy	Precision	Recall
ADADELTA	80	0.30	160	0.9487	0.9837	0.9344
RMSProp	80	0.15	70	**0.9535**	0.9651	**0.9614**
ADAM	**70**	0.30	**35**	0.9413	**0.9876**	0.9189

HS - HIDDEN_SIZE, δ - dropout rate, NE - number of epochs

- 注意的点
    - 数据集问题
        - 输入可能被净化，但仍会危及应用程序；-----过滤未全面，可以绕过。
        - 恶意输入可能会在对其他变量的赋值过程中，在代码片之间进行传播----恶意代码存在恶意传递，数据流分析
    - 优化器超参数保留问题--默认值
        - 简化第一个实验并想获得对该问题的一些直觉
    - 加入n LSTM layers的原因---[44]

        - 在漏洞检测任务中，操作码在切片中的顺序是非常相关的。
        - 产生输出向量，编码之前的操作码及其顺序的信息
    - 加入Dropout layer的原因
        - 位置，在每个LSTM的后面
        - 可以进一步减少过拟合--概率--设置为零--噪声-防止学习到无关模式
            - 该层根据给定的概率(δ)，将其输入的一些条目随机设置为零。该方法的目标是在模型中引入噪声，以防止它记忆由LSTM层偶然学到的无关模式。
    - Opcode的优势
        - 类似于C/C++的汇编操作码的操作码格式处理PHP代码片，被认为是一种中间语言。
        - 更接近于语言的内部结构---有助于解放分类任务
        - 由于切片可以属于任何web应用程序变体，使用中间语言来表示它们可以促进变体之间的分析横向性。
        - 这种中间语言尚未用于解决这个任务
    - 嵌入层的作用
        - CNN 和 RNN 层之前通常有一个嵌入层，它**将离散符号映射到连续向量，解决自然语言数据的稀疏性问题。**此外，通常将这些组件的输出提供给前馈组件，该组件学习执行所需的任务，如分类[28]。
        - one-hot vectors and embedding vectors
        - embedding vectors
            - CBOW
            - skip-gram