PatchRNN: A Deep Learning-Based System for Security Patch Identification

Xinda Wang*, Shu Wang*, Pengbin Feng*, Kun Sun*, Sushil Jajodia*,

Sanae Benchaaboun†, and Frank Geck†

*Center for Secure Information Systems, George Mason University
†CSIA Division, C5ISR Center, Space and Terrestrial Communications Directorate,
U.S. Army Combat Capabilities Development Command (DEVCOM)

What is a patch?

- A software patch is *a set of changes between two versions* of source code to improve security, resolve functionality issues, and add new features.
  - Generated using `diff` command.
  - On version control platform like GitHub, a commit can be regarded as a patch with some description comments.

Security vs. Non-Security Patch

Security patches:

- address specific security vulnerabilities.

Non-security patches:

- correct the software bugs.
- add/update functionality.

```
From f58c25069cf4a986fe17a80c5b38687e31feb539 Mon Sep 17 00:00:00 2001
From: Sebastian Pipping <sebastian@pipping.org>
Date: Wed, 10 Oct 2018 14:49:51 +0200

    ResetUri: Protect against NULL

diff --git a/src/UriCommon.c b/src/UriCommon.c
index 3775306..039beda 100644
--- a/src/UriCommon.c
+++ b/src/UriCommon.c
@@ -75,6 +75,9 @@

void URI_FUNC(ResetUri)(URI_TYPE(Uri) * uri) {
+   if (uri == NULL) {
+       return;
+   }
    memset(uri, 0, sizeof(URI_TYPE(Uri)));
 }
 }
```

Listing 1: An example of security patch for NULL pointer dereference vulnerability (CVE-2018-19200).

```
commit ac367d7a2884aa150cdfc0495348fd886d3bd228
Author: Embedthis Software <dev@embedthis.com>
Date:   Thu Nov 12 10:59:07 2015 -0800

    FIX: don't try to catch SIGKILL

diff --git a/src/goahead.c b/src/goahead.c
index 6e6c806a..aa66d292 100644
--- a/src/goahead.c
+++ b/src/goahead.c
@@ -204,7 +204,6 @@ static void initPlatform()
 {
 #if ME_UNIX_LIKE
     signal(SIGTERM, sigHandler);
-    signal(SIGKILL, sigHandler);
     #ifdef SIGPIPE
         signal(SIGPIPE, SIG_IGN);
     #endif
```

Listing 2: An example of non-security patch in *GoAhead*

Why do we need identify security patch?

- Software maintainers are struggling with OSS patches.
  - 96% of Apps contain OSS components that account for 57% of the code base on average[1].
  - Applying all the new patches increases the system downtime and introduces extra workload.
  - Postponing security patches could cause more damages.
    - Examples: Equifax breach, GitLab DDoS, ...
- Therefore, *security patches should have high priority to be applied.*

[1] Synopsys, “Open Source Software and Risk Analysis Report, ”https://www.synopsys.com/content/dam/synopsys/sigassets/reports/2018-ossra.pdf, 2018.

Traditional Approaches

- CVE advisory monitor
  - Rely on the CVE advisories to alert maintainers.
  - Problem: 70% of patches are not timely disclosed in the CVE[2].
- Text mining
  - Analyze textual information to find security related keywords.
  - Problem: changelog is not well-documented.

CVE-2019-10131 Off-by-One Read

```
From cb1214c124e1bd61f7dd551b94a794864861592e
From: Cristy <urban-warrior@imagemagick.org>
Date: Sat, 24 Mar 2018 15:33:39 -0400
Subject: [PATCH] ...

---
 coders/meta.c | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

[2] Li, Frank, and Vern Paxson. "A Large-Scale Empirical Study of Security Patches."
Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.

Traditional Approaches (Cont.)

- Using human defined features.
  - Manually define a set of features on code metrics.
  - Problem
    - Require lots of expertise.
    - Still incur high true positive/negative rate.

Motivation

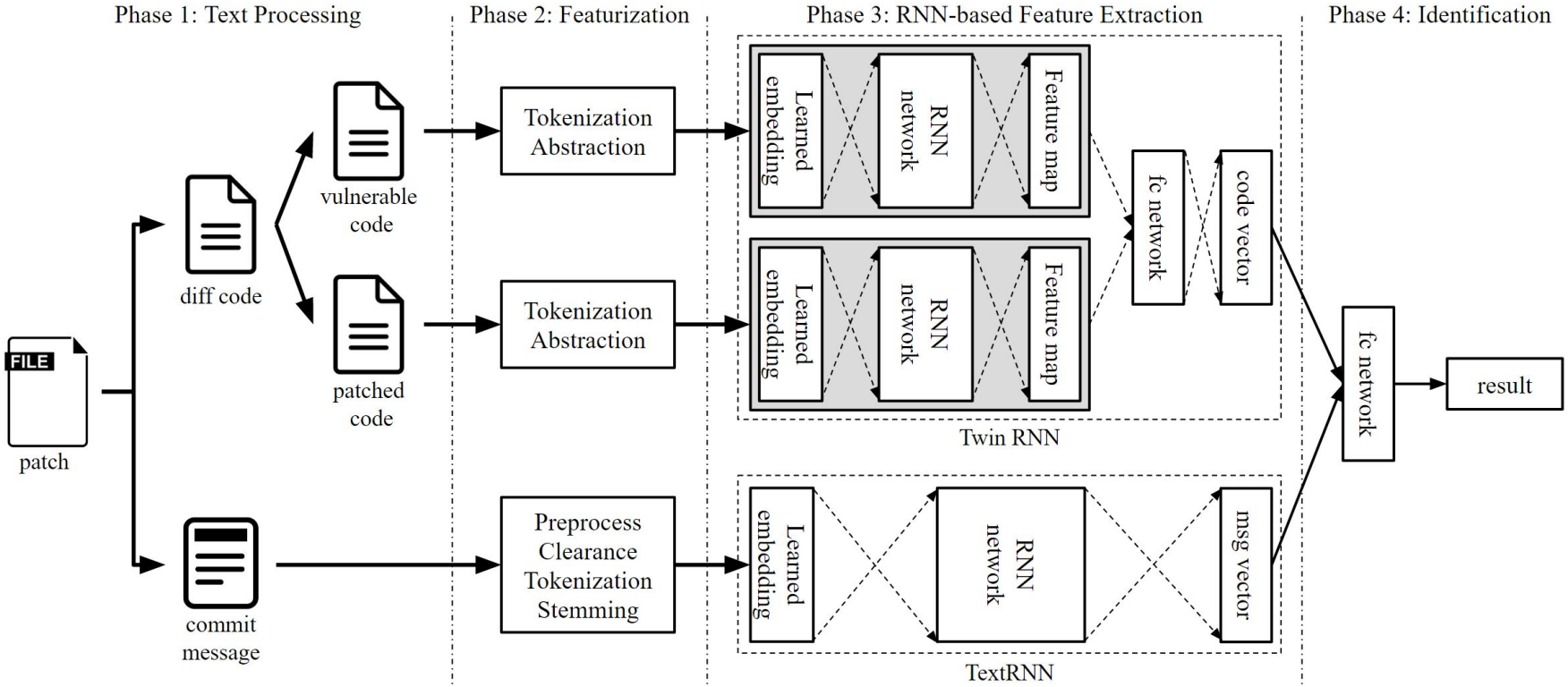
- Source code provides rich syntactic and semantic information.
- Neural networks have shown effectiveness in processing NLP.
    - Program language is also sequential and context-sensitive.

Our Work

To effectively identify security patches, we propose a deep learning based system called ***PatchRNN*** that utilizes both two parts of a commit:

- Commit message
- Source code difference

PatchRNN Overview



Parsing the Commit

Commit Message: Subject + Description

```
From 6d444c273da5499a4cd72f21cb6d4c9a5256807d Mon Sep 17 00:00:00 2001
From: Chris Liddell <chris.liddell@artifex.com>
Date: Wed, 5 Oct 2016 09:55:55 +0100
Subject: [PATCH] Bug 697178: Add a file permissions callback

For the rare occasions when the graphics library directly opens a file
(currently for reading), this allows us to apply any restrictions on
file access normally applied in the interpteter.
```

Source Code Difference

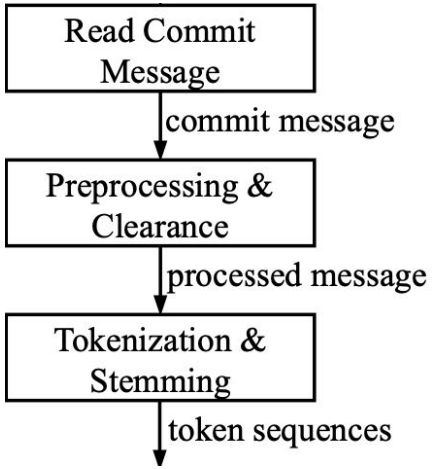
```
diff --git a/base/gsicc_manage.c b/base/gsicc_manage.c
index 931c2a6..e9c09c3 100644
--- a/base/gsicc_manage.c
+++ b/base/gsicc_manage.c
@@ -1124,10 +1124,12 @@ gsicc_open_search(const char* pname, int
namelen, gs_memory_t *mem_gc,
     }

     /* First just try it like it is */
-    str = sfopen(pname, "r", mem_gc);
-    if (str != NULL) {
-        *strp = str;
-        return 0;
+    if (gs_check_file_permission(mem_gc, pname, namelen, "r") >= 0) {
+        str = sfopen(pname, "r", mem_gc);
+        if (str != NULL) {
+            *strp = str;
+            return 0;
+        }
     }

     /* If that fails, try %rom% */ /* FIXME: Not sure this is needed
     or correct */
```

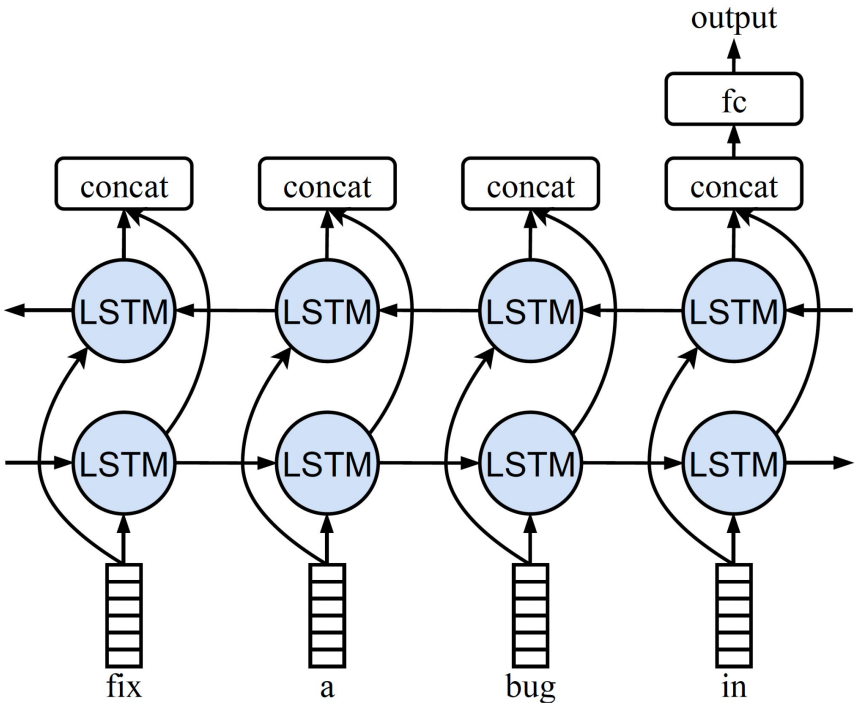
Feature Extraction from Commit Message

- Pre-processing: case lowering, data cleaning, and stopword removal.
- Tokenization and stemming.
- Transforming tokens into word embeddings via *word2vec.*



Feature Extraction from Commit Message (Cont.)

- Then, we develop a *TextRNN* model to generate the message vector.



Feature Extraction from Source Code Difference

- Retrieve the vulnerable and unpatched code.
- Perform the abstraction respectively.

```
    if (snprintf(spath, sizeof(spath), var, iface)
        >= sizeof (spath))
        return -1;

+   /* No path traversal */
+   if (strstr(name, "..") || strchr(name, '/'))
+       return -1;
+
    if (access(spatch, F_OK) != 0)
        return -1;
```

(a) original diff code.

```
    if (FUNC0(VAR0, sizeof(VAR0), VAR1, VAR2) >=
        sizeof (VAR0))
        return -1;

+
+   if (FUNC1(VAR3, LITERAL) || FUNC2(VAR3, LITERAL))
+       return -1;
+
    if (FUNC3(VAR0, VAR4) != 0)
        return -1;
```

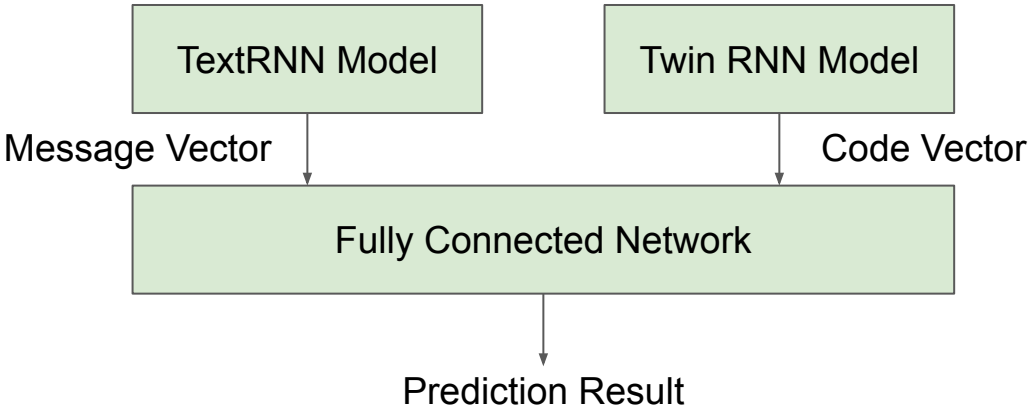
(b) abstracted diff code.

Feature Extraction from Source Code Diff (Cont.)

- Normalize to a fixed length respectively.
- Convert to two vectors via *word2vec*.
- Input in a twin RNN-based model and get the code vector.

Model Learning

- Finally, we concatenate the message and code vectors and then feed them to the prediction model.



Evaluation

- **Dataset**:
  - *PatchDB*[3]: 12,476 security patches and 25,565 non-security collected from NVD and popular GitHub projects).
  - Randomly choose 80% for training and remaining 20% for testing.
- **Implementation**: 3K LoC in Python 3 and Pytorch 1.6.
- **Environment**: Ubuntu 20.04.1 LTS, Intel Xeon Gold 5122, 3.60-GHz CPU with 64-GB RAM and 2 NVIDIA RTX 2080 Ti GPUs of 11 GB memory.

[3] Wang, Xinda, Shu Wang, Pengbin Feng, Kun Sun, and Sushil Jajodia. "PatchDB: A Large-Scale Security Patch Dataset." In 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 149-160. IEEE, 2021.

Evaluation (Cont.)

- **Performance**: 83.57% accuracy with 0.75 F1 score.
- **Overhead**
  - Preprocessing: 4.4 sec/patch.
  - Prediction: 1.2 sec/patch.

Case Study on Nginx

	Official Doc.	Ground Truth		Inference Results	
Changes with	Security	Security	Non-Sec.	T.P.	F.P.
1.19.1	0	8	11	4	0
1.19.2	0	8	7	3	0
1.19.3	0	7	12	3	0
Sum.	0	23	30	10	0

We identifies 10 security patches that are silently released by NGINX with no false positives.

Conclusion

- We initiate the study of using deep learning based approach to identify security patch.
- The evaluation on large-scale real-world dataset and Nginx shows its effectiveness with low false positives.

Thank you!

Q & A