Testovanie zraniteľností aplikácií

Lucia Kokuľová

Vedúci práce: RNDr. JUDr. Pavol Sokol, PhD.
Konzultant: Mgr. Terézia Mézešová

Motivácia

- narastajúca potreba vývojárov vytvoriť nielen atraktívne aplikácie, ale takisto dostatočne zabezpečené

Zdroj: https://www.informationsecuritybuzz.com/study-research/organization-requires-endpoint-threat-detection-response/

Ciele práce

- analyzovať možnosti testovania zraniteľností aplikácií
- porovnať aktuálne prístupy a implementácie testovania zraniteľností aplikácií
- navrhnúť a implementovať rozhranie pre vybrané metódy testovania zraniteľnosti aplikácií.

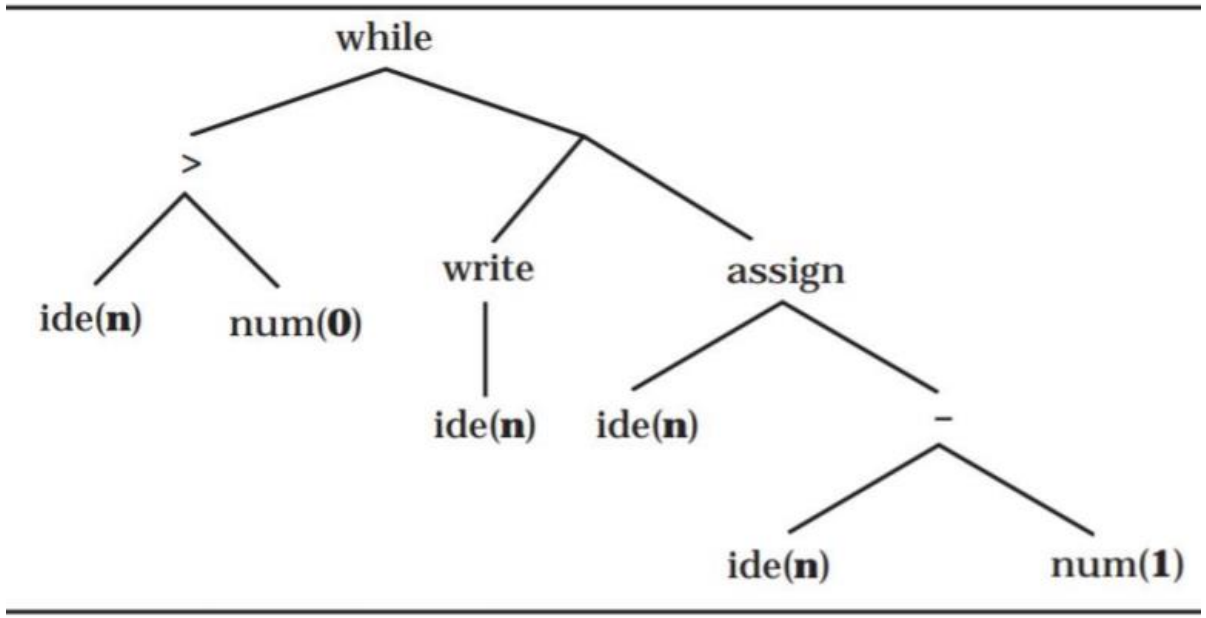
Testovanie zraniteľností aplikácií

- zraniteľnosť - slabina či nedostatok systému, ktorý umožňuje uplatnenie hrozby, je to vlastnosť aktíva
- **penetračné testy**
  - simulácia útoku hackera, pokus o neoprávnený prienik do systému
  - interný a externý penetračný test
- **statická analýza kódu**
  - hľadanie chýb v zdrojovom kóde aplikácií (správnosť podmienok, cyklov, výnimiek, vylúčenie možnosti SQL injection, ...)

Statická analýza kódu

- taint analýza
- data flow diagram
- code property graf
- abstraktný syntaktický strom

```
while n>0 do write n; n:=n-1 end while.
```



Požiadavky na nástroj do našej infraštruktúry

- Nástroje vo všeobecnosti:
  - delenie podľa programovacích jazykov
  - špecializované na konkrétnu metódu statickej analýzy
  - pluginy do vývojových prostredí
  - ...
- open source
- spoľahlivosť existujúcej analýzy
- možnosť dopĺňať vlastné pravidlá na analýzu
- podpora operačných systémov Windows a Linux
- podpora open source databáz

- open source platforma na priebežnú kontrolu kvality (= bezpečnosti) kódu
- vykonáva kontrolu kódu statickou analýzou
- deteguje chyby v kóde a bezpečnostné zraniteľnosti vo viac než 20 jazykoch
  - Java, C#, PHP, JavaScript, Python, XML,...

a implementácia pravidiel

Využitie abstraktného syntaktického stromu

- Využíva sa pri tvorbe a uplatňovaní pravidiel
- Kódovacie pravidlo hrá úlohu návštevníka, ktorý je schopný navštíviť všetky uzly daného abstraktného syntaktického stromu

Tvorba pravidiel

- Pridanie závislostí
- Implementácia pravidiel
  - Výber *jazyka*, pre ktorý sa pravidlo implementuje
  - Výber *zraniteľnosti*, pre ktorú sa pravidlo implementuje

OWASP - Top 10 bezpečnostných rizík 2017

A1 - Injection
A2 - Broken Authentication and Session Management
A3 - Sensitive Data Exposure
A4 - XML External Entity (XXE)
A5 - Broken Access Control
A6 - Security Misconfiguration
A7 - Cross-Site Scripting (XSS)
A8 - Insecure Deserialization
A9 - Using Components with Known Vulnerabilities
A10 - Insufficient Logging & Monitoring

OWASP - Top 10 bezpečnostných rizík 2017

A1 - Injection
A2 - Broken Authentication and Session Management
A3 - Sensitive Data Exposure
A4 - XML External Entity (XXE)
A5 - Broken Access Control
A6 - Security Misconfiguration
A7 - Cross-Site Scripting (XSS)
A8 - Insecure Deserialization
A9 - Using Components with Known Vulnerabilities
A10 - Insufficient Logging & Monitoring

+ 1 pravidlo mimo OWASP Top 10

Implementácia pravidiel

- Abstraktný syntaktický strom vytvorený nástrojom SonarQube delí jednotlivé podstromy do kategórií
  - *IF_STATEMENT*
  - *WHILE_LOOP*
  - *...*
  - *METHOD_INVOCATION* → objekt = metóda(iný objekt)
    → objekt . metóda

Implementácia pravidiel

```
nodesToVisit() {
for (Kind method : AST.Kind1, AST.Kind2, ...)
   visitNode(method);
}

visitNode(Kind method) {
   search for vulnerabilities
   if (found)
      reportIssue();
}
```

Výsledky testovania

A1 - Injection

```
String query = "SELECT data FROM user_data WHERE user_name = "
                    + request.getParameter("customerName");
```

```
String name = request.getParameter("customerName");
String query = "SELECT account_balance FROM user_data WHERE
                  user_name = ? ";
PreparedStatement pstmt = connection.prepareStatement(query);
pstmt.setString(1, name);
ResultSet results = pstmt.executeQuery();
```

Výsledky testovania

A4 - XML External Entity (XXE)

```
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setFeature("http://xml.org/sax/features/external-general-
                entities", true);

libxml_disable_entity_loader(false);
```

```
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setFeature("http://xml.org/sax/features/external-general-
                entities", false);

libxml_disable_entity_loader(true);
```

Výsledky testovania

A9 - Using Components with Known Vulnerabilities

```
$passwordDB = sha1($data[0]['login'].'*'.$password);
$passwordDB = sha1($bForm->get('email')->getData()
                   .'*'
                   .md5(rand(1,9999).microtime()));
```

```
$passwordDB = hash('sha256', '$data[0]['login'].'*'.$password');
$passwordDB = hash('sha256', '$bForm->get('email')->getData()
                   .'*'
                   .hash('sha256', 'rand(1,9999).microtime())');
```

Výsledky testovania

Generovanie bezpečných náhodných čísel

```
$passwordDB = sha1($bForm->get('email')->getData()
                .'*'
                .md5(rand(1,9999).microtime()));
```

```
$passwordDB = sha1($bForm->get('email')->getData()
                .'*'
                .md5(openssl_random_pseudo_bytes().microtime()));
```

Analýza projektov v nástroji SonarQube

Analýza testovacích projektov z našej univerzity

- testovanie všetkými pravidlami nástroja SonarQube
- 52 000 riadkov kódu v jazyku **PHP**
- 2 zraniteľnosti, 20 chýb, 4 000 code smells
- 6 000 riadkov kódu v jazyku **Java**
- 4 chyby, 21 code smells

Analýza projektov zo služby GitHub

- testovanie pomocou nami implementovaných pravidiel
- 594 000 riadkov kódu najobľúbenejších projektov v jazyku **Java**
- 58 zraniteľností

Použitá literatúra a zdroje

- YAMAGUCHI, Fabian, Nico GOLDE, Daniel ARP, Konrad RIECK, 2014. Modelling and Discovering Vulnerabilities with Code Property Graphs.
- WEIDMAN, Georgia, 2014. Penetration Testing: A Hands-On Introduction to Hacking. No Starch Press.
- https://www.sonarsource.com/products/codeanalyzers/sonarjava.html
- https://www.sonarqube.org/
- https://www.owasp.org/index.php/Top_10_2017-Top_10
- https://www.sec.cs.tu-bs.de/pubs/2014-ieeesp.pdf
- https://medium.com/
- http://www.julioauto.com/project/visual-data-tracer.html
- https://docs.oracle.com/javase/7/docs/api/

Ďakujem za pozornosť

Otázky?