Static Analysis and code review

A journey through time

Migchiel de Jong

Software Security Consultant @ HP ESP - Fortify

Audit this code

```
/* Make these globals rather than local to mapping_chdir to avoid stack overflow */
char pathspace[MAXPATHLEN];
char old_mapped_path[MAXPATHLEN];

void do_elem(char *dir)
{
    /* . */
    if (dir[0] == '.' && dir[1] == '\0') {
    /* ignore it */
    return;
    }

    /* .. */
    if (dir[0] == '.' && dir[1] == '.' && dir[2] == '\0') {
    char *last;
    /* lop the last directory off the path */
    if ((last = strrchr(mapped_path, '/'))) {
        /* If start of pathname leave the / */
        if (last == mapped_path)
        last++;
        *last = '\0';
    }
    return;
    }

    /* append the dir part with a leading / unless at root */
    if (!(mapped_path[0] == '/' && mapped_path[1] == '\0'))
    if (strlen(mapped_path) < sizeof(mapped_path) - 1)
        strcat(mapped_path, "/");
    if (sizeof(mapped_path) - strlen(mapped_path) > 1)
    strncat(mapped_path, dir, sizeof(mapped_path) - strlen(mapped_path) - 1);
}

int mapping_chdir(char *orig_path)
{
    int ret;
    char *sl, *path;

    strcpy(old_mapped_path, mapped_path);
    path = &pathspace[0];
    strcpy(path, orig_path);

    /* / at start of path, set the start of the mapped_path to / */
    if (path[0] == '/') {
    mapped_path[0] = '/';
    mapped_path[1] = '\0';
    path++;
    }
```

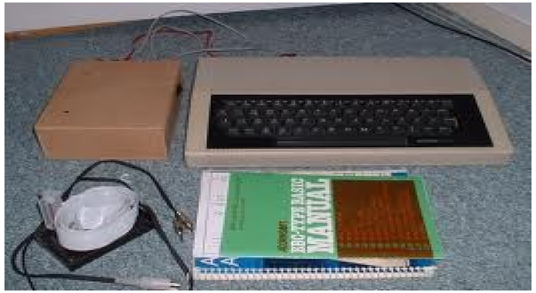
```
/* Helper function for sgetpwnam(). */
char *sgetsave(char *s)
{
    char *new;

    new = (char *) malloc(strlen(s) + 1);

    if (new == NULL) {
    perror_reply(421, "Local resource failure: malloc");
    dologout(1);
    /* NOTREACHED */
    }
    (void) strcpy(new, s);
    return (new);
}

/* Save the result of a getpwnam.  Used for USER command, since the data
 * returned must not be clobbered by any other command (e.g., globbing). */
struct passwd *sgetpwnam(char *name)
{
    static struct passwd save;
    register struct passwd *p;
#ifdef M_UNIX
    struct passwd *ret = (struct passwd *) NULL;
#endif
    char *sgetsave(char *s);
#ifdef KERBEROS
    register struct authorization *q;
#endif /* KERBEROS */

#if defined(SecureWare) || defined(HPUX_10_TRUSTED)
    struct pr_passwd *pr;
#endif

#ifdef KERBEROS
    init_krb();
    q = getauthuid(p->pw_uid);
    end_krb();
#endif /* KERBEROS */

#ifdef M_UNIX
#if defined(SecureWare) || defined(HPUX_10_TRUSTED)
    if ((pr = getprpwnam(name)) == NULL)
    goto DONE;
#endif /* SecureWare || HPUX_10_TRUSTED */
    if ((p = getpwnam(name)) == NULL)
    goto DONE;
#else /* M_UNIX */
#if defined(SecureWare) || defined(HPUX_10_TRUSTED)
    if ((pr = getprpwnam(name)) == NULL)
    return ((struct passwd *) pr);

#endif
```

A quick recap: Where we are today

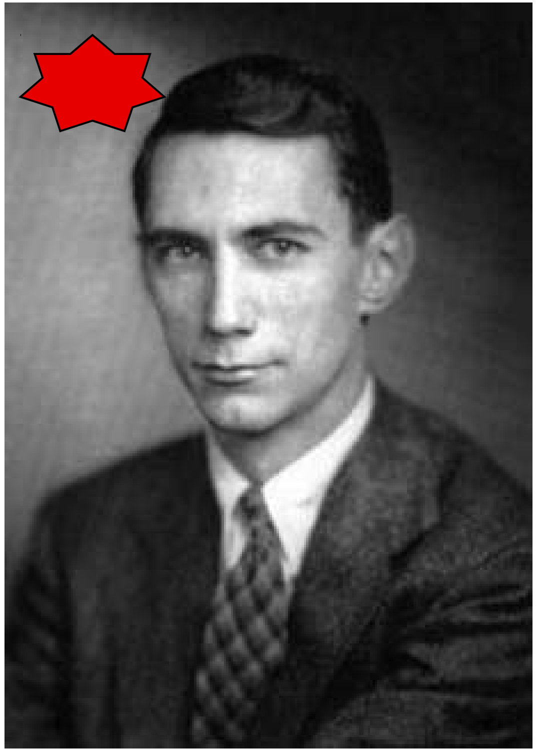
- Small coding errors can have a big effect on security.
- Typical software development practices don’t address the problem.
- As a group, developers tend to make the same security mistakes over and over.
- Static analysis tools can help identify common security errors early.

What to find?

- Web-based enterprise applications
  - ✓ SQLI (and other injection attacks), XSS
  - ✓ Missing data validation
  - ✓ Session management errors (length, identifier)
  - ✓ Information leakage between sessions (concurrency)
  - ✓ Bad error handling (allowing system probing)
  - ✓ Compliance errors (treatment of personal data)
  - ✓ Insecure configuration

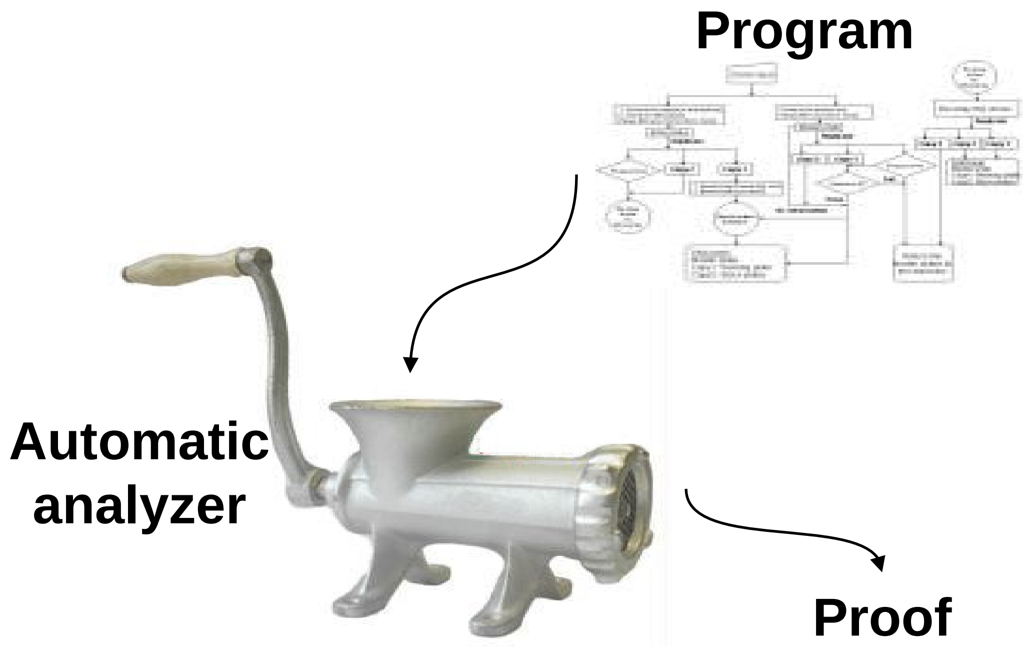

BBC-TYPE BASIC MANUAL

520ST

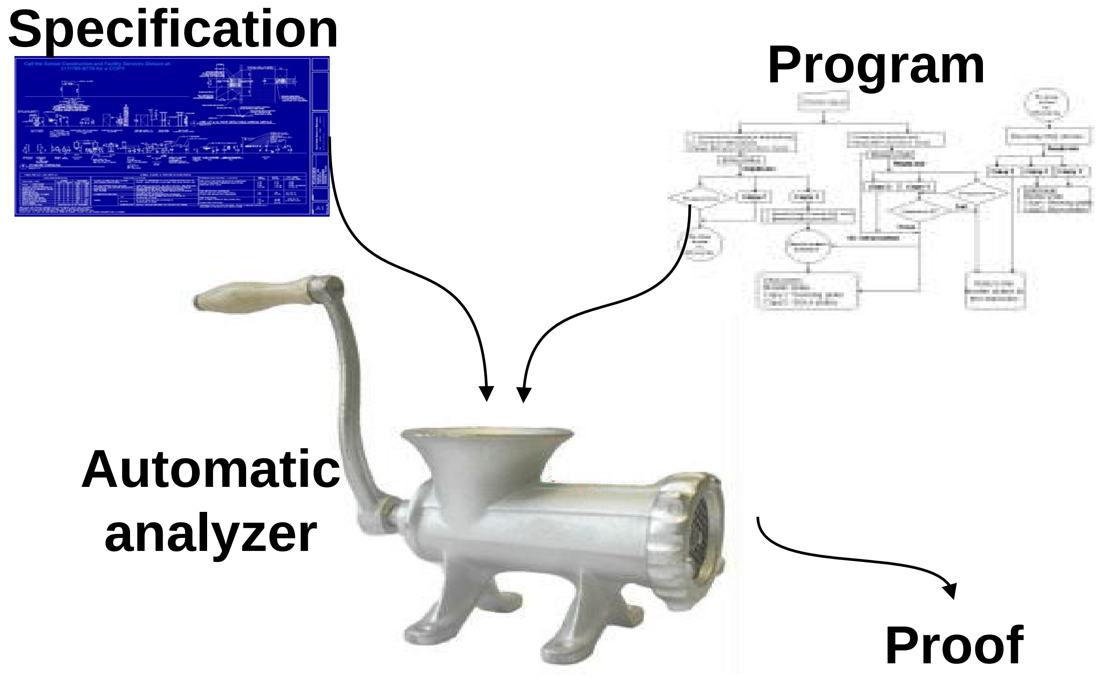


Proof

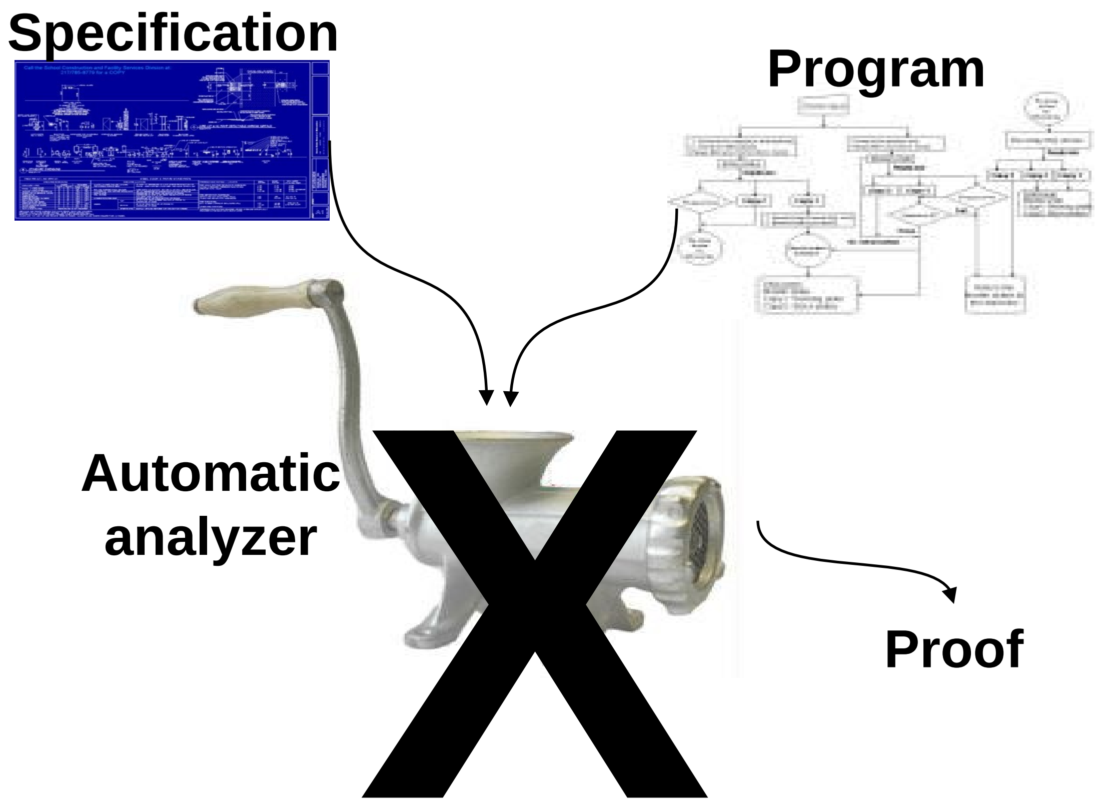
Program verification take #1



Program verification take #2

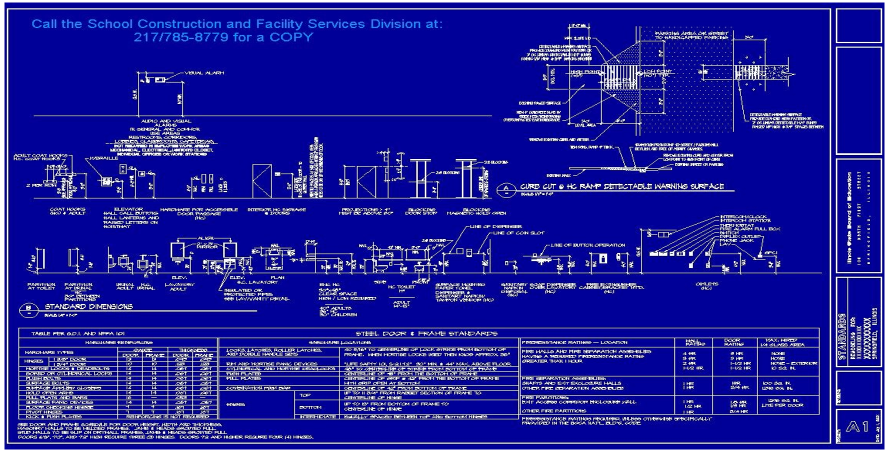


Program verification: a snag

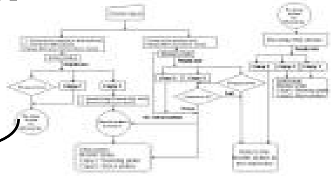


Program verification: a practical dilemma

Specification


Call the School Construction and Facility Services Division at:
217/785-8779 for a COPY
CURB CUT @ HC RAMP DETECTABLE WARNING SURFACE
STANDARD DIMENSIONS
STEEL DOOR & FRAME STANDARDS
A1

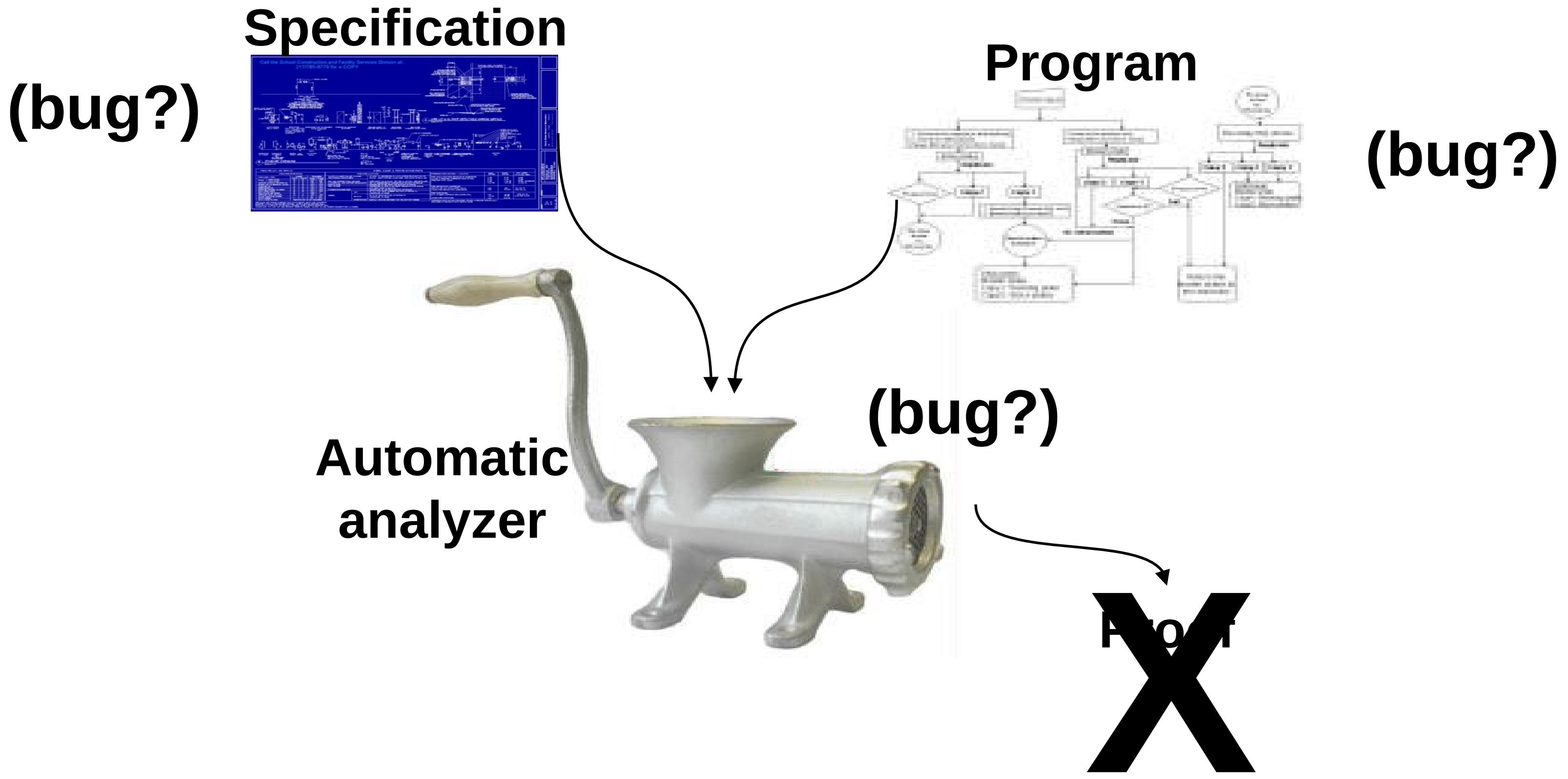
Program



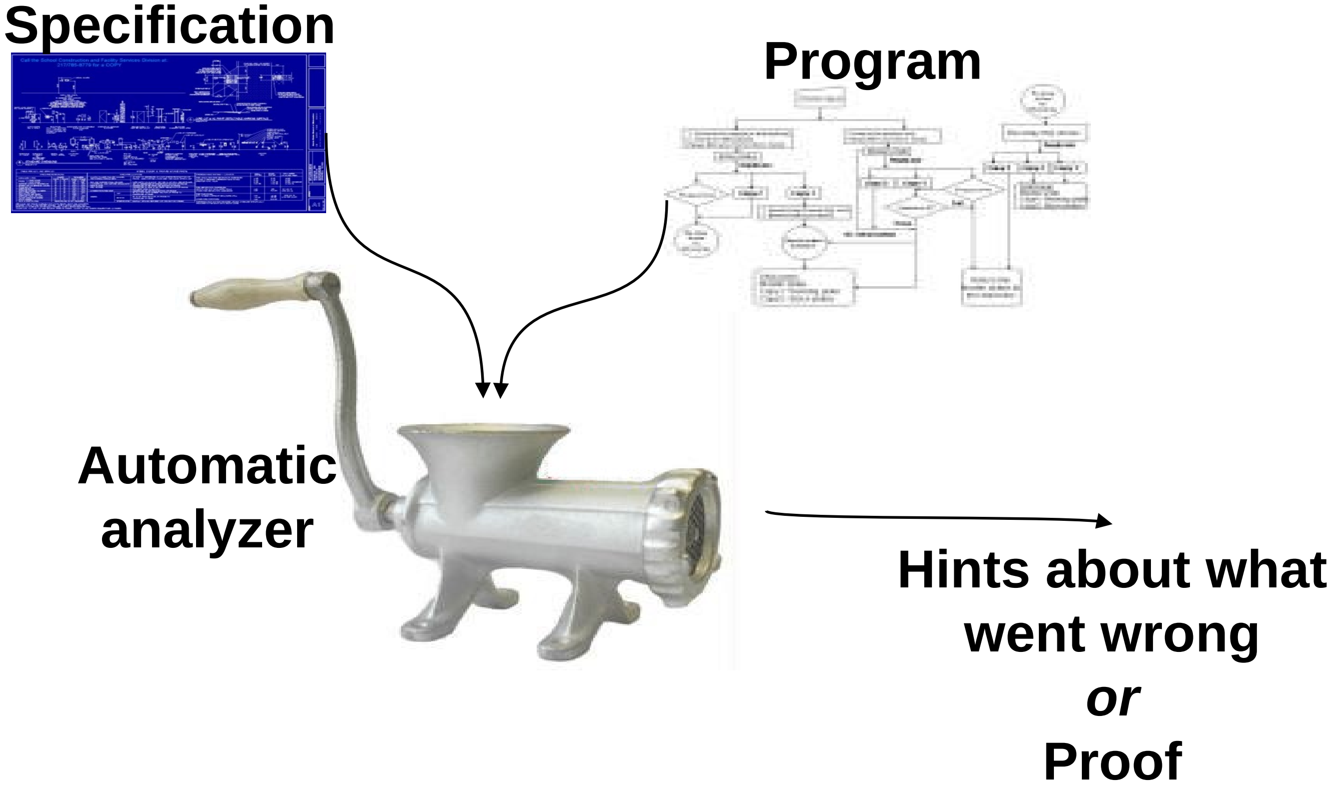
Automatic analyzer

Proof

Program verification: the maiden voyage

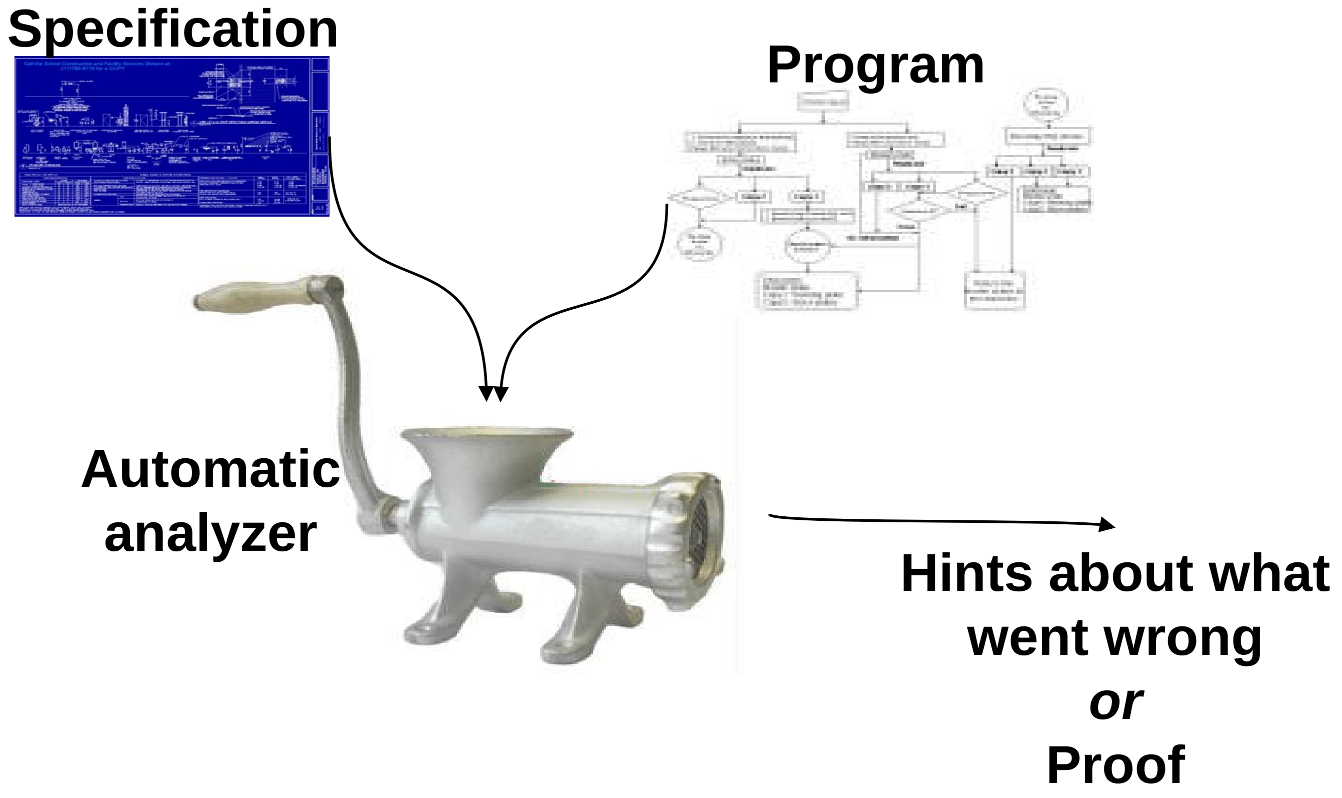


Program verification: one more tweak



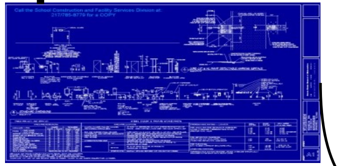
time passes …

Program verification today

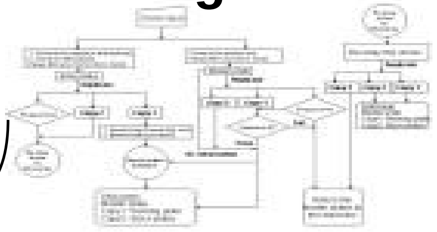


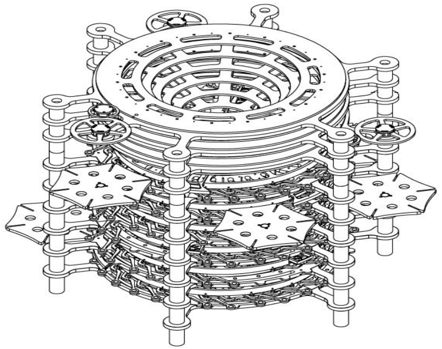
A less ambitious plan

Specification



Program

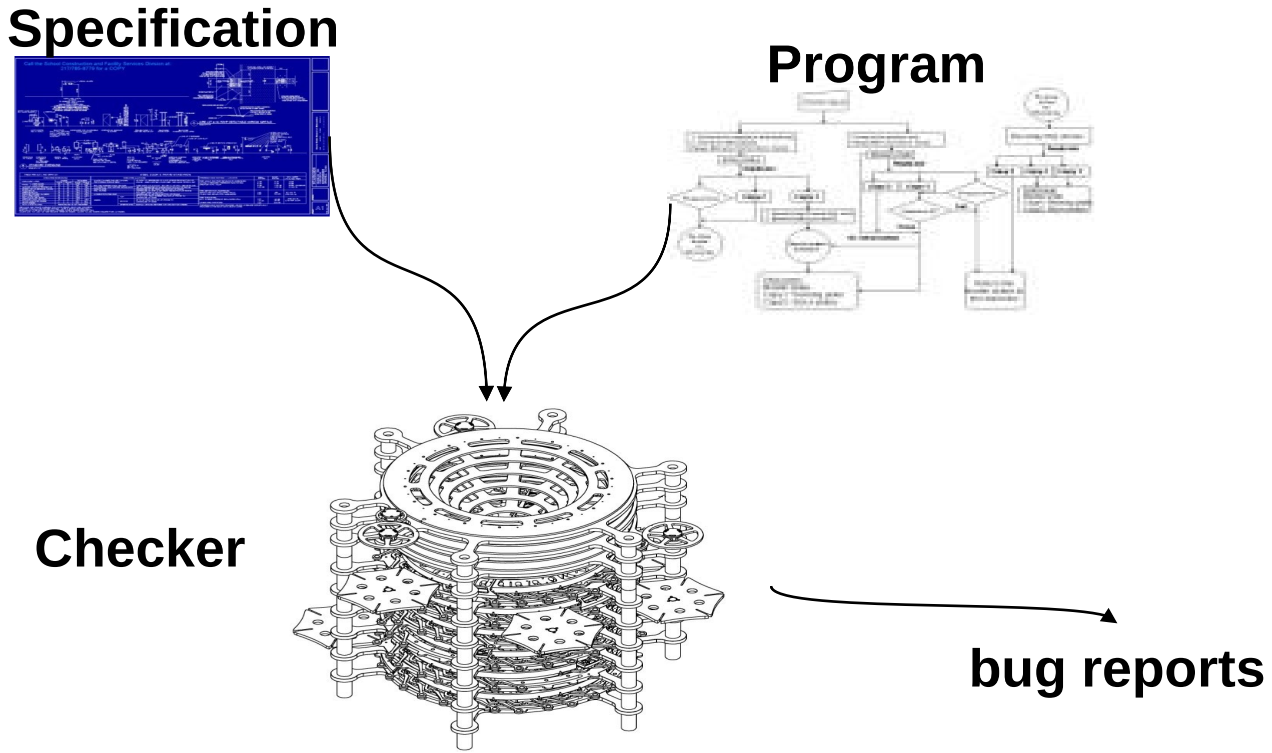




Checker

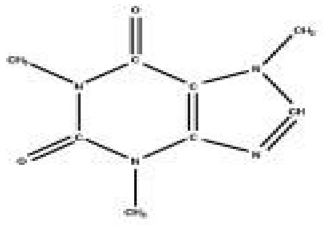
Hints about what went wrong
or
Proof

A less ambitious plan

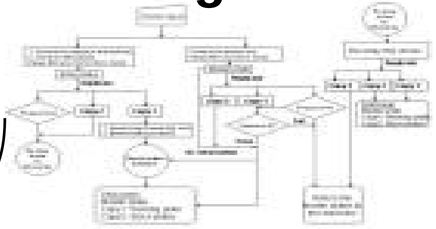


A less ambitious plan

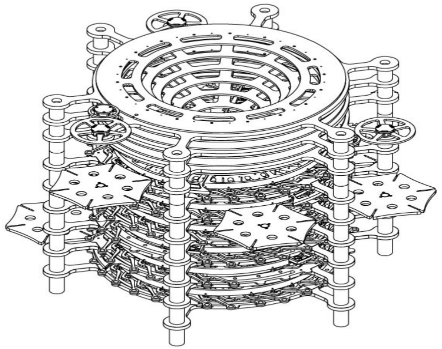
Property



Program

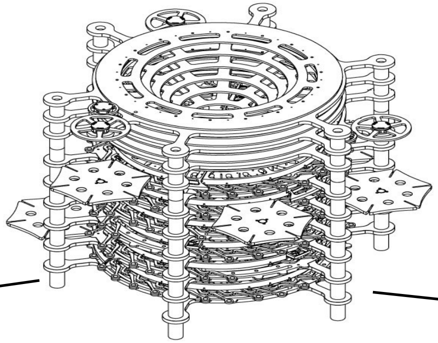


Checker



bug reports

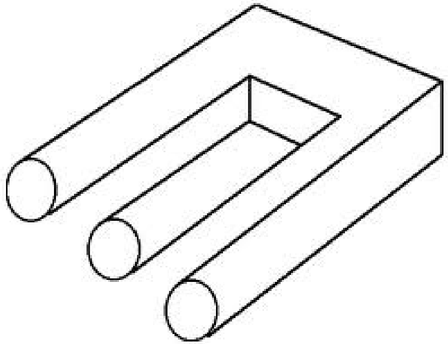
Three hard problems



$$\begin{tikzcd}
\text{Make sense of the program} \arrow[r] & \text{Compute findings} \arrow[r] & \text{Explain findings to user}
\end{tikzcd}$$

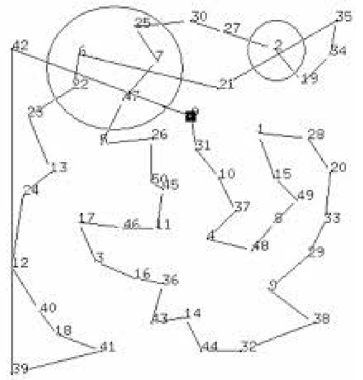
Make sense of the program

- Academic solutions typically target a limited number of languages (often tackling just a language subset).
- Enterprise applications use:
  - ✓ C/C++
  - ✓ Java / JSP / JS
  - ✓ PL/SQL
  - ✓ C#
  - ✓ Visual Basic



- Critical for success: robustness over precision

Compute findings



- Focus of most academic research
- Problem: No one-size fits all technique
- Solution: Build a flexible model, use multiple analyzers
- Gotchas: context, capacity
- Tricky:
  - ✓ pointers/pointer aliasing
  - ✓ function pointers/reflection/inversion of control (IOC)
  - ✓ loops

Context is King: <u>token</u>

`read`

Context is King: line

```
read(f,buf+len1,len-len1);
```

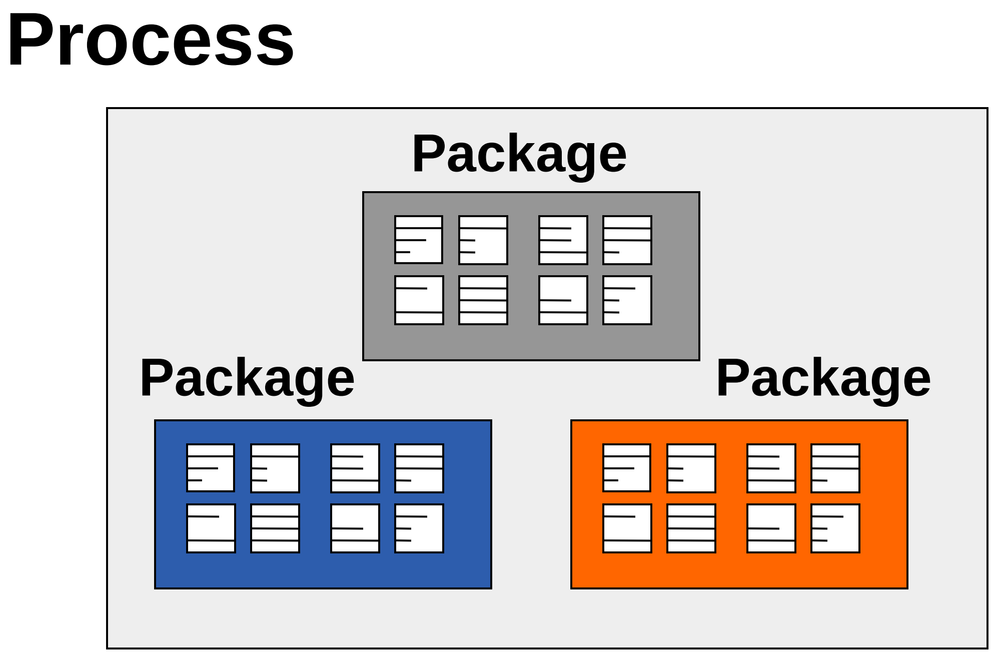
Context is King: <u>function</u>

```
static int my_read(int f,char *buf,int len)
{
    int len1 = 0;
    int ret;

    while(len1 < len) {
        ret = read(f,buf+len1,len-len1);
        if(ret < 0)
            return -1;
        len1 += ret;
    }
    return len;
}
```

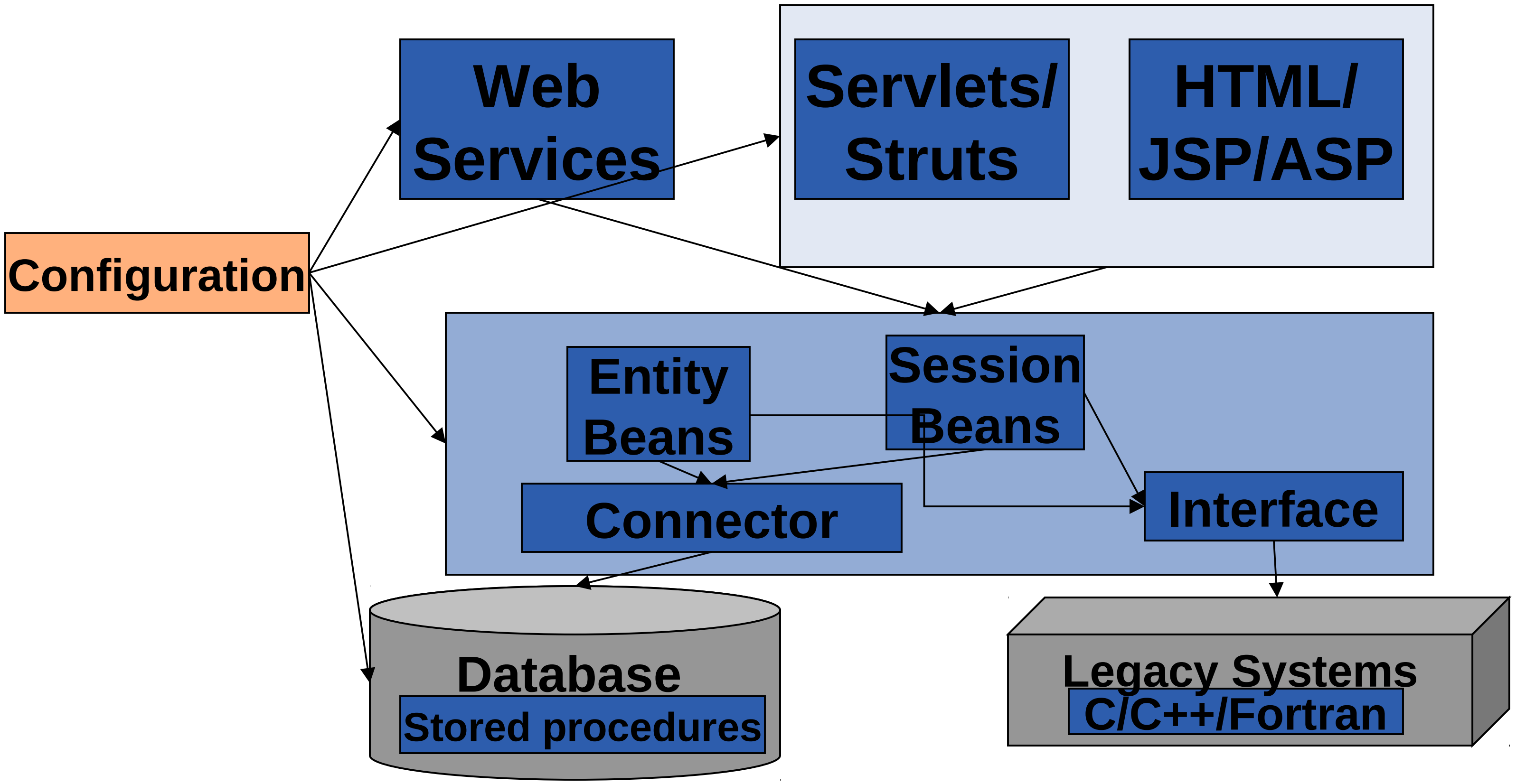
Context is King: file

```
/* Make these globals rather than local to mapping_chdir to avoid stack overflow */
char pathspace[MAXPATHLEN];
char old_mapped_path[MAXPATHLEN];

void do_elem(char *dir)
{
    /* . */
    if (dir[0] == '.' && dir[1] == '\0') {
    /* ignore it */
    return;
    }

    /* .. */
    if (dir[0] == '.' && dir[1] == '.' && dir[2] == '\0') {
    char *last;
    /* lop the last directory off the path */
    if ((last = strrchr(mapped_path, '/'))) {
        /* If start of pathname leave the / */
        if (last == mapped_path)
        last++;
        *last = '\0';
    }
    return;
    }

    /* append the dir part with a leading / unless at root */
    if (!(mapped_path[0] == '/' && mapped_path[1] == '\0'))
    if (strlen(mapped_path) < sizeof(mapped_path) - 1)
        strcat(mapped_path, "/");
    if (sizeof(mapped_path) - strlen(mapped_path) > 1)
    strncat(mapped_path, dir, sizeof(mapped_path) - strlen(mapped_path) - 1);
}
```

Context is King: process

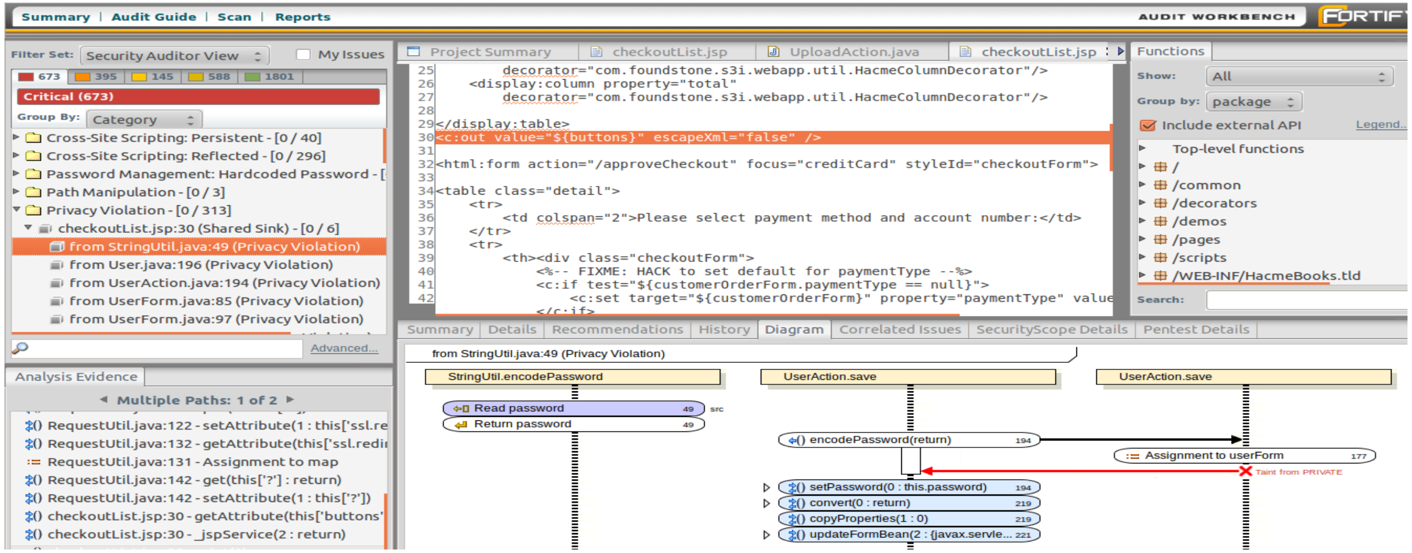


Context is King: system



Explain output to user

- **Less studied in academia**
- **Most of the perceived value is actually here**



Filter Set: Security Auditor View   My Issues

673 | 395 | 145 | 588 | 1801

Critical (673)

Group By: Category

- Cross-Site Scripting: Persistent - [0 / 40]
- Cross-Site Scripting: Reflected - [0 / 296]
- Password Management: Hardcoded Password - [
- Path Manipulation - [0 / 3]
- Privacy Violation - [0 / 313]
  - checkoutList.jsp:30 (Shared Sink) - [0 / 6]
    - from StringUtil.java:49 (Privacy Violation)
    - from User.java:196 (Privacy Violation)
    - from UserAction.java:194 (Privacy Violation)
    - from UserForm.java:85 (Privacy Violation)
    - from UserForm.java:97 (Privacy Violation)

Advanced...

Analysis Evidence

Multiple Paths: 1 of 2

```
RequestUtil.java:122 - setAttribute(1 : this['ssl.re
RequestUtil.java:132 - getAttribute(this['ssl.redi
RequestUtil.java:131 - Assignment to map
RequestUtil.java:142 - get(this['?'] : return)
RequestUtil.java:142 - setAttribute(1 : this['?'])
checkoutList.jsp:30 - getAttribute(this['buttons'
checkoutList.jsp:30 - _jspService(2 : return)
```

Project Summary | checkoutList.jsp | UploadAction.java | checkoutList.jsp

```
        decorator="com.foundstone.s3i.webapp.util.HacmeColumnDecorator"/>
    <display:column property="total"
        decorator="com.foundstone.s3i.webapp.util.HacmeColumnDecorator"/>

</display:table>
<c:out value="${buttons}" escapeXml="false" />

<html:form action="/approveCheckout" focus="creditCard" styleId="checkoutForm">

<table class="detail">
    <tr>
        <td colspan="2">Please select payment method and account number:</td>
    </tr>
    <tr>
        <th><div class="checkoutForm">
            <%-- FIXME: HACK to set default for paymentType --%>
            <c:if test="${customerOrderForm.paymentType == null}">
                <c:set target="${customerOrderForm}" property="paymentType" value
            </c:if>
```

Functions

Show: All

Group by: package

Include external API   Legend...

Top-level functions
- /
- /common
- /decorators
- /demos
- /pages
- /scripts
- /WEB-INF/HacmeBooks.tld

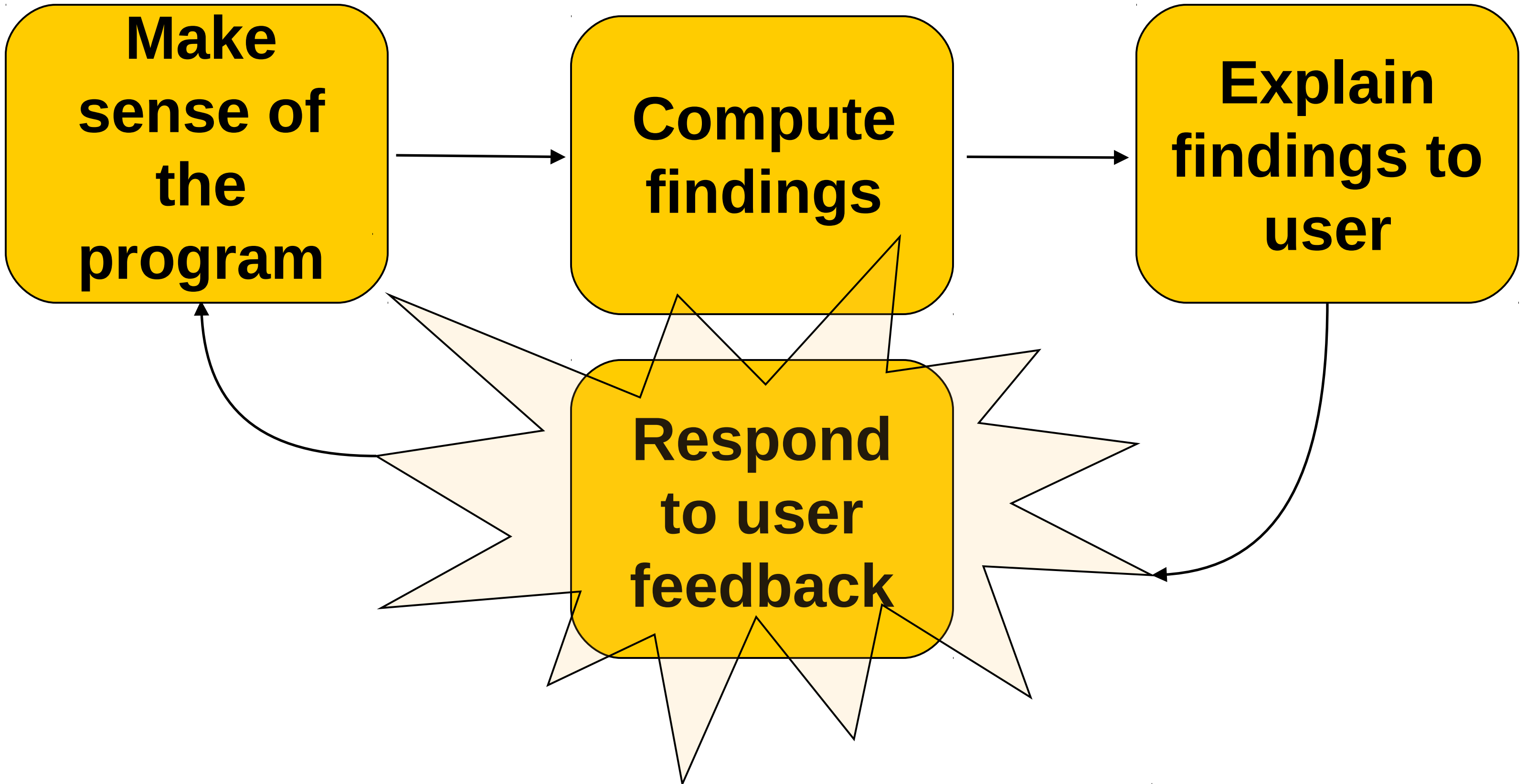
Search:

Summary | Details | Recommendations | History | Diagram | Correlated Issues | SecurityScope Details | Pentest Details

from StringUtil.java:49 (Privacy Violation)

StringUtil.encodePassword
UserAction.save
UserAction.save
Read password 49
src
Return password 49
encodePassword(return) 194
Assignment to userForm 177
Taint from PRIVATE
setPassword(0 : this.password) 194
convert(0 : return) 219
copyProperties(1 : 0) 219
updateFormBean(2 : {javax.servle... 221

A Critical Fourth Problem



Future developments

- Use context to infer intent from the code

    Weakly typed languages

- Speed up analysis utilizing multi core multi CPU setups

    Use the ‘time gained’ to do more analysis (breadth and depth)

- Incremental analysis

THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
GRAZIE
MEHRBANI
BÏYAN
SHUKRIA
BOLZÏN
MERCI
JUSPAXAR
GOZAIMASHITA
EFCHARISTO
PALDIES