BF “Hands-On” – Exercises

Tutorial – July 25, QRS 2017

Irena Bojanova
National Institute of Standards and Technology (NIST)

https://samate.nist.gov/BF/
BF: BOF Exercises

Use BF to describe known software vulnerabilities or to identify gaps in existing repositories:

1) Ghost: BOF $\rightarrow$ CVE-2015-0235
2) Chrome: BOF $\rightarrow$ CVE-2010-1773
3) CWE gaps: BOF $\rightarrow$ Refactoring CWEs
BOF: Causes, Attributes, and Consequences

**FOP**

**Result fault:**
- ✓ Int Overflow
- ✓ Int Underflow
- ✓ Int Coercion
- ✓ etc.

**Operator:**
- ✓ Incorrect Calculation
- ✓ Missing Factor
- ✓ Off By One
- ✓ Incorrect Argument
- ✓ Incorrect Conversion

**Operand error:**
- ✓ Incorrect Result

**Data type:**
- ✓ Input Not Properly Checked
- ✓ No NULL Termination
- ✓ Wrong Index / Pointer Out of Range
- ✓ Array Too Small
- ✓ Too Much Data

**Access:**
- ✓ Read
- ✓ Write

**Boundary:**
- ✓ Below
- ✓ Above

**Location:**
- ✓ Heap
- ✓ Stack

**Magnitude:**
- ✓ Small
- ✓ Moderate
- ✓ Far

**Data Size:**
- ✓ Little
- ✓ Some
- ✓ Huge

**Reach:**
- ✓ Continuous
- ✓ Discrete

**Consequences:**
- ✓ Altered Control Flow
- ✓ Incorrect Results
- ✓ Program Crash
- ✓ System Crash
- ✓ Resource Exhaustion
- ✓ Information Exposure/Change/Loss
- ✓ Denial Of Service
- ✓ Arbitrary Code Execution
- ✓ Admin Server Access/Complete Host Takeover
- ✓ Account Access
- ✓ Credentials Compromise
- ✓ Confidentiality/Authentication/Authorization/Integrity Loss

**Access:**
- ✓ Altered Control Flow
- ✓ Incorrect Results
- ✓ Program Crash
- ✓ System Crash
- ✓ Resource Exhaustion
- ✓ Information Exposure/Change/Loss
- ✓ Denial Of Service
- ✓ Arbitrary Code Execution
- ✓ Admin Server Access/Complete Host Takeover
- ✓ Account Access
- ✓ Credentials Compromise
- ✓ Confidentiality/Authentication/Authorization/Integrity Loss
BOF: Exercise 1 (Ghost)

Ghost: CVE-2015-0235
BOF: Exercise 1 (Ghost) – CVE-2015-0235

Create a BF description of CVE-2015-0235:

1. Examine the listed below CVE description, references [1,2,3], and source code excerpts with the bug and the fix.
2. Analyze the gathered information and come up with a BF description utilizing the BOF taxonomy (causes, attributes, and consequences).

**CVE-2015-0235 (Ghost):** “Heap-based buffer overflow in the _nss_hostname_digits_dots function in glibc 2.2, and other 2.x versions before 2.18, allows context-dependent attackers to execute arbitrary code via vectors related to the (1) gethostbyname or (2) gethostbyname2 function, aka GHOST.” [1]

BOF: Exercise 1 – Source Code

Code With Bug

1 /* calculate size incorrectly*/
2 size_needed = (sizeof (*host_addr)+ sizeof (*h_addr_ptrs)
   + strlen (name) + 1);
3
4 host_addr = (host_addr_t *) *buffer;
5 h_addr_ptrs = (host_addr_list_t (*)(char *) host_addr
   + sizeof (*host_addr));
6 hostname = (char *) h_addr_ptrs + sizeof (*h_addr_ptrs);
7 resbuf->h_name = strcpy (hostname, name);

Code With Fix

1 /* calculate size incorrectly*/
2 size_needed = (sizeof (*host_addr) + sizeof (*h_addr_ptrs)
   + sizeof (*h_alias_ptr) + strlen (name) + 1);
3
4 host_addr = (host_addr_t *) *buffer;
5 h_addr_ptrs = (host_addr_list_t (*)(char*) host_addr
   + sizeof (*host_addr));
6 hostname = (char*) h_addr_ptrs + sizeof (*h_addr_ptrs);
7 resbuf->h_name = strcpy (hostname, name);
Chrome: CVE-2010-1773
Create a BF description of CVE-2010-1773:
1. Examine the listed below CVE description, references [1-8], and source code excerpts with bug and fix.
2. Analyze the gathered information and come up with a BF description utilizing the BOF taxonomy.

**CVE-2010-1773 (Chrome WebCore):** “Off-by-one error in the toAlphabetic function in rendering/RenderListMarker.cpp in WebCore in WebKit before r59950, as used in Google Chrome before 5.0.375.70, allows remote attackers to obtain sensitive information, cause a denial of service (memory corruption and application crash), or possibly execute arbitrary code via vectors related to list markers for HTML lists, aka rdar problem 8009118.” [1]

BOF: Exercise 2 – Source Code

Code With Bug

```java
1 if (type == AlphabeticSequence) {
2     while ((numberShadow /= sequenceSize) > 0) {
3         letters[lettersSize - ++length] = sequence[numberShadow % sequenceSize - 1];
4     }
5 }
6 }
7 }
```

Code With Fix

```java
1 if (type == AlphabeticSequence) {
2     while ((numberShadow /= sequenceSize) > 0) {
3         --numberShadow;
4         letters[lettersSize - ++length] = sequence[numberShadow % sequenceSize];
5     }
6 }
7 }
8 }
```
CWE Gaps: Refactoring BOF CWEs
CWE-120: Buffer Copy without Checking Size of Input: The program copies an input buffer to an output buffer without verifying that the size of the input buffer is less than the size of the output buffer, leading to a buffer overflow.
CWE-121: Stack-based Buffer Overflow
CWE-122: Heap-based Buffer Overflow
CWE-123: Write-what-where Condition
CWE-124: Buffer Underwrite (‘Buffer Underflow’) 
CWE-125: Out-of-bounds Read
CWE-126: Buffer Over-read
CWE-127: Buffer Under-read
CWE-786: Access of Memory Location Before Start of Buffer
CWE-787: Out-of-bounds Write
CWE-788: Access of Memory Location After End of Buffer

Applying our definition and attributes, Buffer Overflow CWEs can be categorized as follows.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Before</th>
<th>After</th>
<th>Either End</th>
<th>Stack</th>
<th>Heap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>127</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Either R/W</td>
<td></td>
<td>788</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use BF to describe known software vulnerabilities:

INJ → CVE-2008-5817
INJ: Causes, Attributes, and Consequences

**Causes**
- Input Not Checked Properly
  - Permissive Whitelist
  - Incomplete Blacklist
  - White/Black List Not Checked Properly
- Input Not Sanitized Properly
  - Failure to Reject Input Altogether
  - Failure to Remove Offending Characters
  - Failure to “Escape” Offending Characters

**Attributes**

**Language:**
- SQL, Bash,
- regex, XML/Xscript
- PHP, CGI, etc.

**Special Element:**
- Query Elements
- Header Separators
- Scripting Elements
- Format Parameters
- Path Traversals
- Wildcards
- Metacharacters, etc.

**Entry Point:**
- Data Entry Field
- Markup Tag
- Function Call Parameter
- Program Call Argument, etc.

**Invalid Construct:**
- Database Query
- Command
- Regular Expression
- Markup
- Script, etc.

**Consequences**

- Incorrect Results
  - Add Command
  - Computer Worms Propagation
  - Denial Of Service
- Altered Control Flow
- Information Exposure/Change/Loss
  - Arbitrary Code Execution
  - ACI
  - Admin Server Access/Complete Host Takeover
- Confidentiality/Authentication/Authorization/Integrity Loss
- Credentials Compromise
  - Account Access
- Mask Legitimate Commands/Information
Create a BF description of CVE-2008-5817:

1. Examine the listed below CVE description, references [1,2,3,4].
2. Analyze the gathered information and come up with a BF description utilizing the INJ taxonomy (causes, attributes, and consequences).

CVE-2008-5817: “Multiple SQL injection vulnerabilities in index.php in Web Scribble Solutions webClassifieds 2005 allow remote attackers to execute arbitrary SQL commands via the (1) user and (2) password fields in a sign_in action.” [1]

BF: ENC Exercise

Use BF to describe known software vulnerabilities:

ENC → CVE-2002-1697
ENC: Causes, Attributes, and Consequences

**Causes**
- Modification of Encryption Algorithm
- Improper Encryption Algorithm/Step
  - Missing
  - Inadequate
  - Risky/Broken
- Insecure Mode of Operation
- Exposed Private/Secret Key
- KMN Fault

**Attributes**
- **Sensitive Data:**
  - Credentials
  - System, State
  - Cryptographic
  - Digital Documents
- **Data State:**
  - Stored
  - Transferred
- **Algorithm:**
  - Symmetric
  - Asymmetric
- **Security Service:**
  - Confidentiality
  - Integrity, Identity

**Consequences**

IEX of Sensitive Data
Create a BF description of CVE-2002-1697:

1. Examine the listed below CVE description, as well as references [1,2,3,4].
2. Analyze the gathered information and come up with a BF description utilizing the ENC taxonomy (causes, attributes, and consequences).

**CVE-2002-1697**: “Electronic Code Book (ECB) mode in VTun 2.0 through 2.5 uses a weak encryption algorithm that produces the same ciphertext from the same plaintext blocks, which could allow remote attackers to gain sensitive information.” [1]

BF: VRF Exercise – CVE-2015-2141

Use BF to describe known software vulnerabilities:

VRF → CVE-2015-2141
VRF: Causes, Attributes, and Consequences

**Causes**
- Modification of Verification Algorithm
- Improper Verification Algorithm/ Step
- Missing
- Inadequate
- Risky/Broken
- Weak
- RND> Inadequate/Predictable
- ENC/KMN/VRF Fault

**Attributes**

**Verified Data:**
- Secret
- Public

**Algorithm:**
- Hash Function + RND
- MAC
- Digital Signature

**Security Service:**
- Integrity Authentication
- Identity Authentication
- Origin Non-Repudiation

**Consequences**
- Unverified Keying Material
- Unverified/ Wrongly Verified Data/Identity/Origin

IEX
VRF: Exercise — CVE-CVE-2015-2141

Create a BF description of CVE-CVE-2015-2141:
1. Examine the listed below CVE description, as well as references [1,2,3,4].
2. Analyze the gathered information and come up with a BF description utilizing the ENC taxonomy (causes, attributes, and consequences).

CVE-2015-2141: “The InvertibleRWFunction::CalculateInverse function in rw.cpp in libcrypt++ 5.6.2 does not properly blind private key operations for the Rabin-Williams digital signature algorithm, which allows remote attackers to obtain private keys via a timing attack.” [1]

BF: KMN Exercise (FREAK)

Use BF to describe known software vulnerabilities:

**FREAK**: CRY $\rightarrow$ CVE-2015-0204, CVE-2015-1637, CVE-2015-1067
KMN: Causes, Attributes, and Consequences

**Causes**
- Improper Algorithm/Step
  - Missing
  - Inadequate
  - Risky/Broken
  - Weak
- Improper Offer/Use of Weak Protocol
- Hardcoded Key
- Wrong Key Selection
- RND->Inadequate/Predictable
- ENC/KMN/VRF Fault

**Attributes**

**Cryptographic Data:**
- Hashes
- Keying Material
- Digital Certificate

**Algorithm:**
- Hash Function + RND
- MAC
- Digital Signature

**Operation:**
- Generate/Select
- Store
- Distribute
- Use
- Destroy

**Consequences**
- Weak Public Key
- IEX of Private Key
- Weak Secret Key
- Unverified Keying Material
- IEX of Keying Material

Create a BF description for FREAK – CVE-2015-0204, CVE-2015-1637, CVE-2015-1067:
1. Examine the listed below CVE descriptions, references [1,2,3,4,5,6,7], and source code with bug and fix.
2. Analyze the gathered information and come up with a BF description utilizing the CRY taxonomy.

CVE-2015-0204: “The ssl3_get_key_exchange function in s3_clnt.c in OpenSSL before 0.9.8zd, 1.0.0 before 1.0.0p, and 1.0.1 before 1.0.1k allows remote SSL servers to conduct RSA-to-EXPORT_RSA downgrade attacks and facilitate brute-force decryption by offering a weak ephemeral RSA key in a noncompliant role, related to the "FREAK" issue. NOTE: the scope of this CVE is only client code based on OpenSSL, not EXPORT_RSA issues associated with servers or other TLS implementations.” [1]


CVE-2015-1067: “Secure Transport in Apple iOS before 8.2, Apple OS X through 10.10.2, and Apple TV before 7.1 does not properly restrict TLS state transitions, which makes it easier for remote attackers to conduct cipher-downgrade attacks to EXPORT_RSA ciphers via crafted TLS traffic, related to the "FREAK" issue, a different vulnerability than CVE-2015-0204 and CVE-2015-1637.” [3]

[7] GitHub, openssl, Only allow ephemeral RSA keys in export ciphersuites, https://github.com/openssl/openssl/commit/ce325c60c74b0fa784f5872404b722e120e5cab0?diff=split.
BF: KMN Exercise (FREAK) – Source Code

Client

```c
#ifdef OPENSSL_NO_RSA
if (alg_k & SSL_kRSA) {
    if (!SSL_C_IS_EXPORT(s->s3->tmp.new_cipher)) {
        al=SSL_AD_UNEXPECTED_MESSAGE;
        SSLerr(SSL_F_SSL3_GET_SERVER_CERTIFICATE,SSL_R_UNEXPECTED_MESSAGE);
        goto f_err;
    }
}
#endif

if ((rsa=RSA_new()) == NULL) {
    SSLerr(SSL_F_SSL3_GET_KEY_EXCHANGE,ERR_R_MALLOC_FAILURE);
}
else
    s->s3->tmp.use_rsa_tmp=0;
else
    s->s3->tmp.use_rsa_tmp=0;
if ((rsa=RSA_new()) == NULL) {
SSLerr(SSL_F_SSL3_GET_KEY_EXCHANGE,ERR_R_MALLOC_FAILURE);
```

Server

```c
case SSL3_ST_SW_KEY_EXCH_B: if (alg_k & SSL_kRSA) {
    if (!SSL_C_IS_EXPORT(s->s3->tmp.new_cipher->algorithm_mkey) {
        al=SSL_AD_UNEXPECTED_MESSAGE;
        SSLerr(SSL_F_SSL3_GET_SERVER_CERTIFICATE,SSL_R_UNEXPECTED_MESSAGE);
        goto f_err;
    }
}
#endif

if ((s->options & SSL_OP_EPHEMERAL_RSA)
  if ((rsa=RSA_new()) == NULL) {
SSLerr(SSL_F_SSL3_GET_KEY_EXCHANGE,ERR_R_MALLOC_FAILURE);
```

If client ciphersuit is non-export then returned by server RSA keys should be also non-export.

Therefore, handshake that offers export RSA key (512 bits, which is weak) should be abandoned by client.

The buggy code includes a handshake that enables accepting a 512-bit RSA key.

The fix is adding code that checks whether client ciphersuit is non-export and for abandoning the handshake if this is the case.