BF “Hands-On” – Exercises

Tutorial – July 25, QRS 2017

Irena Bojanova

National Institute of Standards and Technology (NIST)

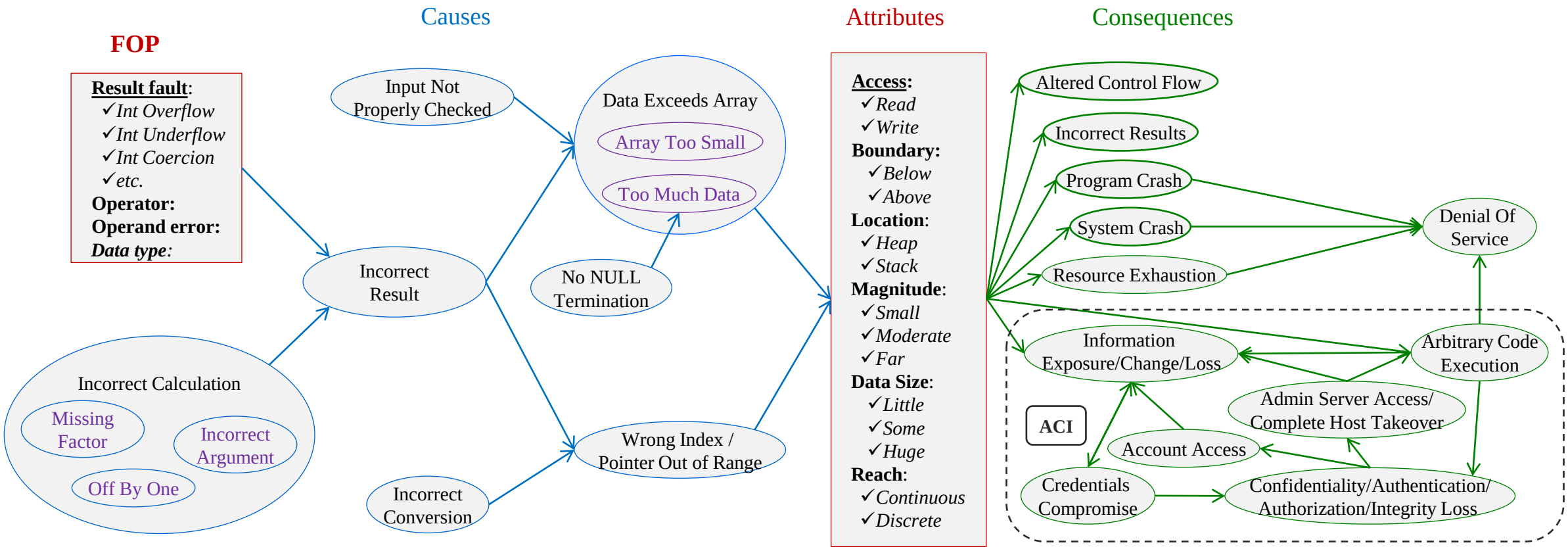
https://samate.nist.gov/BF/

BF: BOF Exercises

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

1) Ghost: BOF → CVE-2015-0235
2) Chrome: BOF → CVE-2010-1773
3) CWE gaps: BOF → Refactoring CWEs

BOF: Causes, Attributes, and Consequences



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]

[1] The MITRE Corporation, CVE Common Vulnerabilities and Exposures, CVE-2015-0235.

[2] Openwall, bringing security into open environment, Qualys Security Advisory CVE-2015-0235.

[3] Qualys Security Advisory CVE-2015-0235.

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);
```

BOF: Exercise 2 (Chrome)

Chrome: CVE-2010-1773

BOF: Exercise 2 (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]

[1] The MITRE Corporation, CVE Common Vulnerabilities and Exposures, CVE-2010-1773.
[2] Robin Gandhi, Buffer Overflow Semantic template CVE-2010-1773.
[3] Tracker, Issue 44955.
[4] chromium, Diff of /branches/WebKit/375/WebCore/rendering/RenderListMarker.cpp. Revision 48099.
[5] chromium, Contents of /branches/WebKit/375/WebCore/rendering/RenderListMarker.cpp. Revision 44321.
[6] chromium, Contents of /branches/WebKit/375/WebCore/rendering/RenderListMarker.cpp. Revision 48100.
[7] webkit, Fix for Crash in WebCore::toAlphabetic() while running MangleMe -and corresponding- https://bugs.webkit.org/show_bug.cgi?id=39508. Reviewed by Darin Adler.
[8] Hat Bugzilla – Bug 596500- (CVE-2010-1773) CVE-2010-1773 WebKit: off-by-one memory read out of bounds vulnerability in handling of HTML lists.

BOF: Exercise 2 – Source Code

Code With Bug

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

Code With Fix

```
1  if (type == AlphabeticSequence)
2  {
3      while ((numberShadow /= sequenceSize) > 0)
4      {
5          --numberShadow;
6          letters[lettersSize - ++length] = sequence[numberShadow % sequenceSize];
7      }
8  }
```

BOF: Exercise 3

CWE Gaps: Refactoring BOF CWEs

BOF: Exercise 3 (Refactoring 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.

Buffer Overflow CWEs Organized by Attribute:

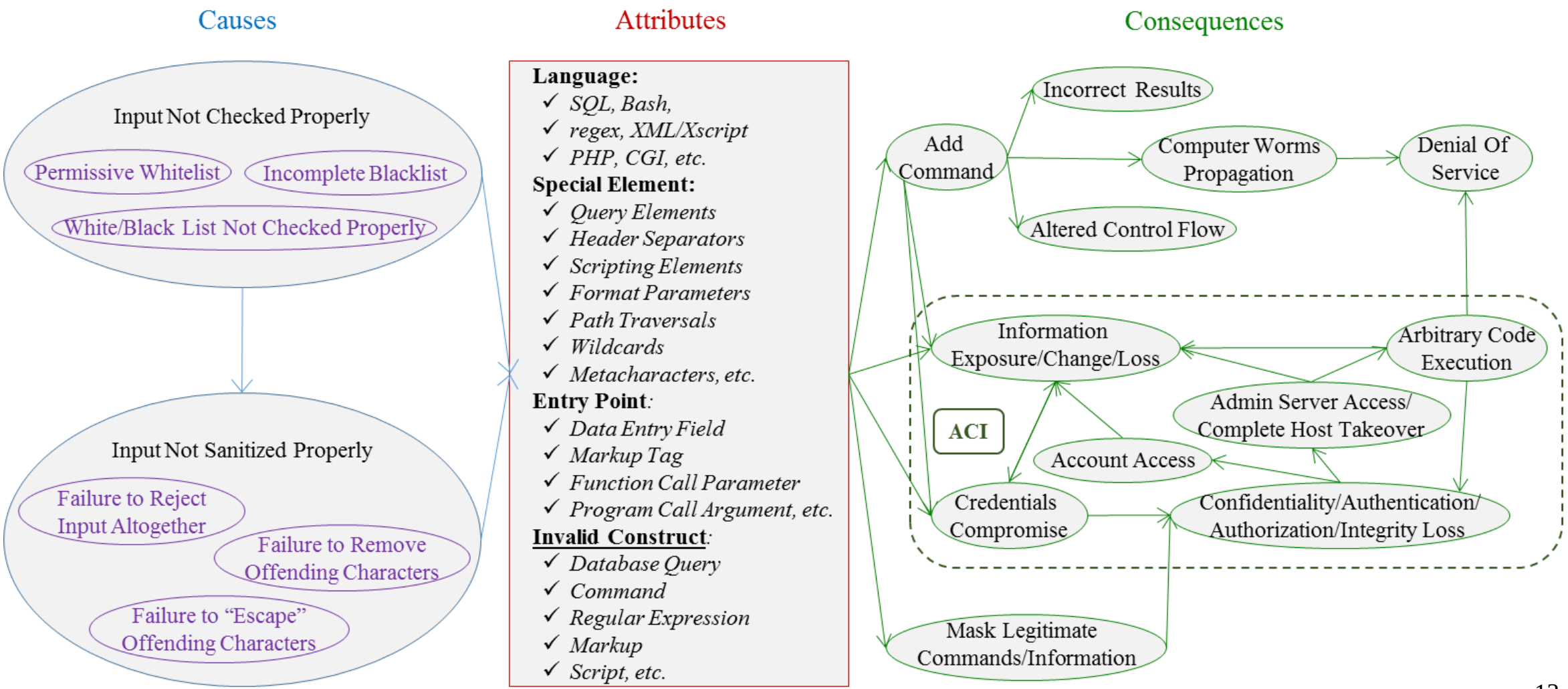
	Before	After	Either End		Stack	Heap
Read	127					
Write						
Either R/W		788				

BF: INJ: Exercise – CVE-2008-5817

Use BF to describe known software vulnerabilities:

INJ → CVE-2008-5817

INJ: Causes, Attributes, and Consequences



INJ: Exercise – CVE-2008-5817

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]

[1] The MITRE Corporation, CVE Common Vulnerabilities and Exposures, CVE-2008-5817.

[2] CXSESECURITY, webClassifieds 2005 (Auth Bypass) SQL Injection Vulnerability CWE-89 CVE-2008-5817.

[3] The MITRE Corporation, CWE Common Weakness Enumeration, CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection').

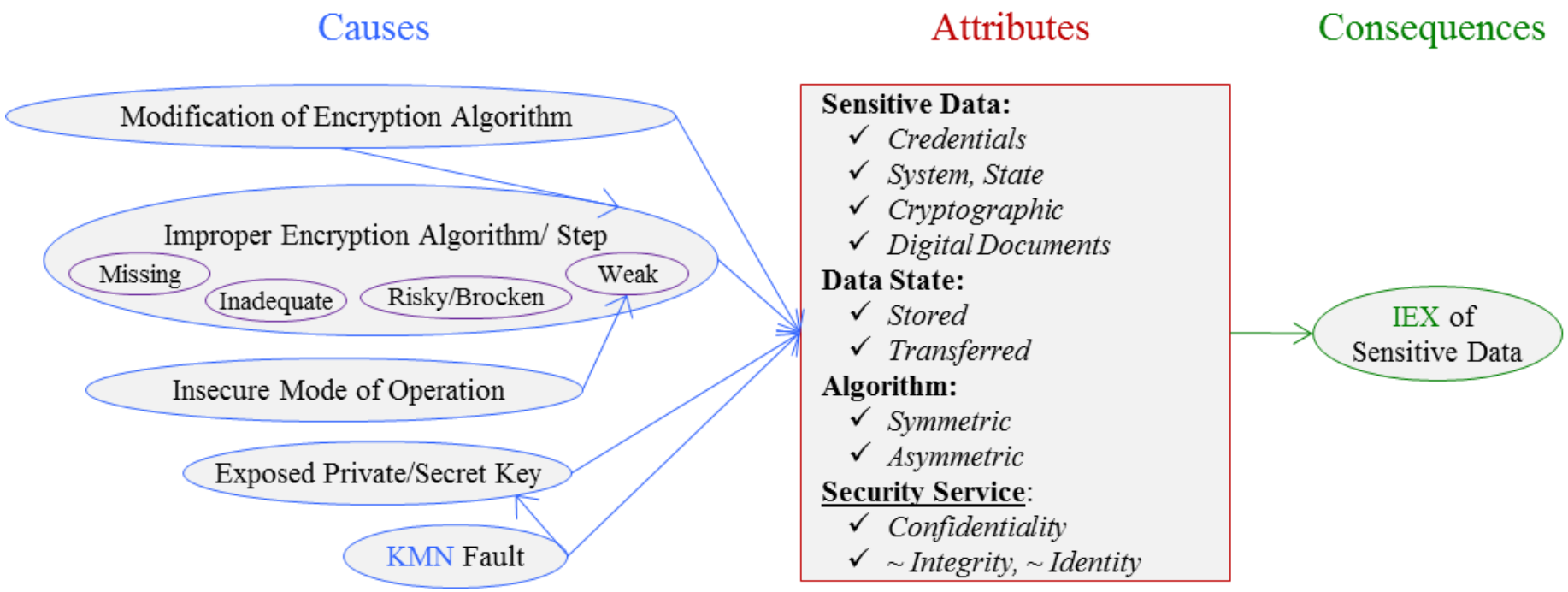
[4] Bricks, SQL injection.

BF: ENC Exercise

Use BF to describe known software vulnerabilities:

ENC → CVE-2002-1697

ENC: Causes, Attributes, and Consequences



ENC: Exercise – CVE-2002-1697

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]

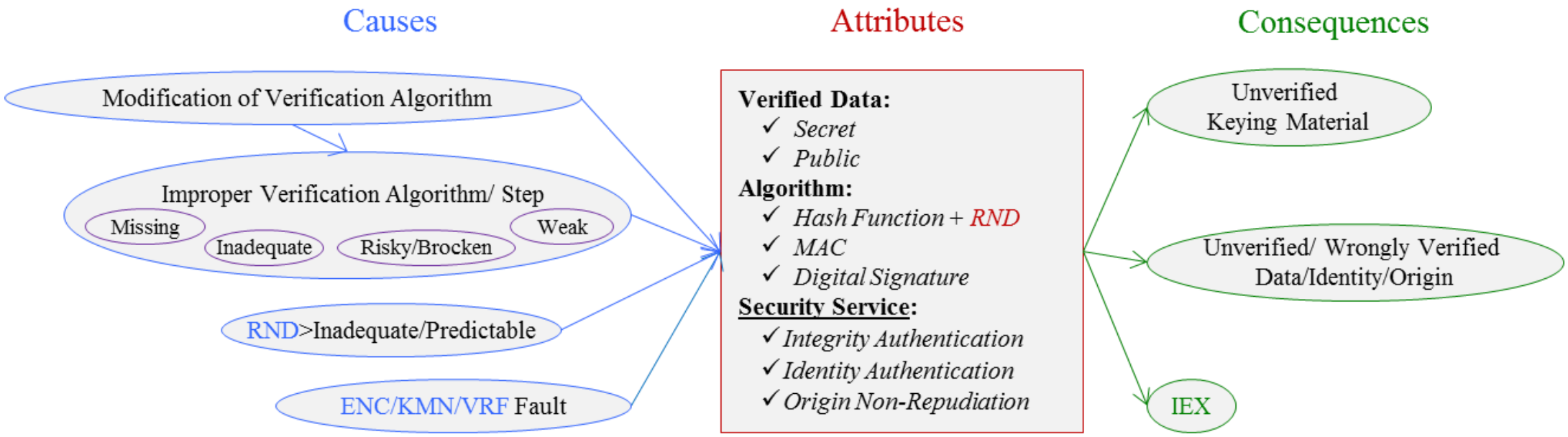
[1] The MITRE Corporation, CVE-2002-1697, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1697.
[2] Wikipedia, RSA (cryptosystem), http://en.wikipedia.org/wiki/RSA_(cryptosystem).
[3] Seclists, Security weaknesses of VTun, http://seclists.org/bugtraq/2002/Jan/119.
[4] Wikipedia, Deterministic encryption, https://en.wikipedia.org/wiki/Deterministic_encryption.

BF: VRF Exercise – CVE-2015-2141

Use BF to describe known software vulnerabilities:

VRF → CVE-2015-2141

VRF: Causes, Attributes, and Consequences



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]

[1] The MITRE Corporation, CVE-2141, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2015-2141

[2] Bugzilla – Bug 936435, VUL-0: CVE-2015-2141: libcryptopp: libcrypto++ -- security update, https://bugzilla.suse.com/show_bug.cgi?id=936435.

[3] E. Sidorov, "Breaking the Rabin-Williams digital signature system implementation in the Crypto++ library," 2015, http://eprint.iacr.org/2015/368.pdf.

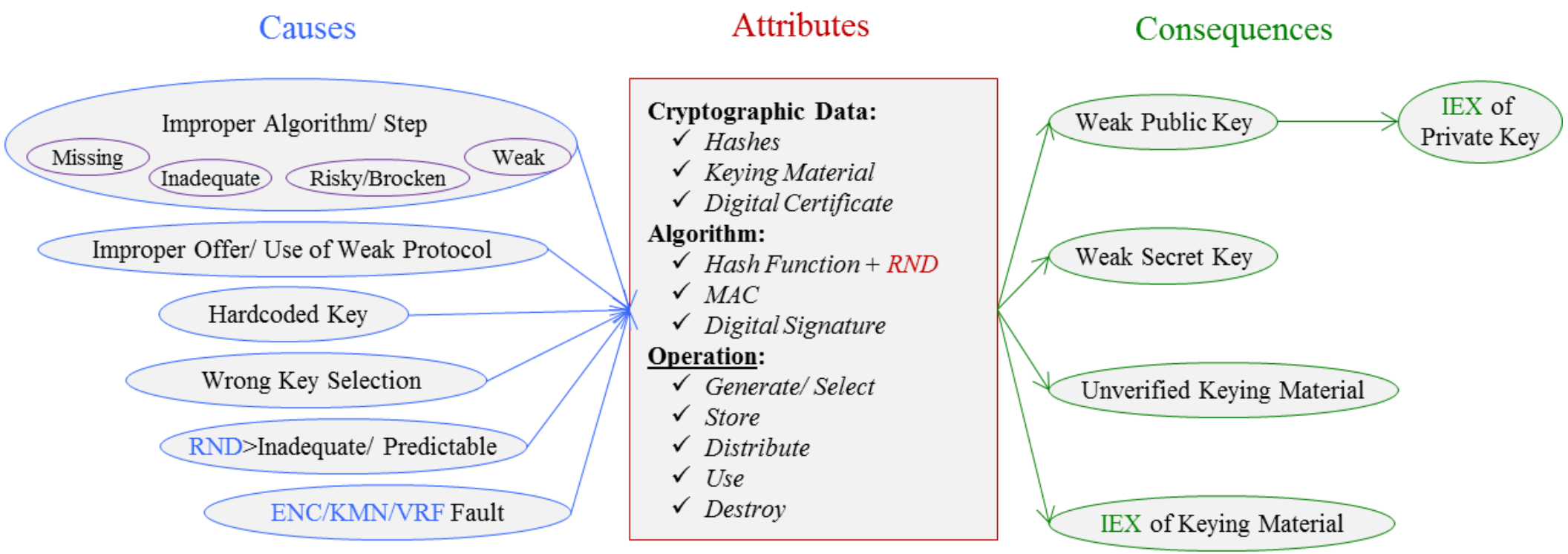
[4] Wikipedia, "Blinding Cryptography," https://en.wikipedia.org/wiki/Blinding_(cryptography).

BF: KMN Exercise (FREAK)

Use BF to describe known software vulnerabilities:

FREAK: CRY → CVE-2015-0204, CVE-2015-1637, CVE-2015-1067

KMN: Causes, Attributes, and Consequences



BF: KMN Exercise (FREAK)– CVE-2015-0204, CVE-2015-1637, CVE-2015-1067

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-2015-1637: "Schannel (aka Secure Channel) in Microsoft Windows Server 2003 SP2, Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 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-2015-1067." [2]

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]

[1] The MITRE Corporation, CVE--2015-0204, https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-0204
[2] The MITRE Corporation, CVE--2015-1637, https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-1637.
[3] The MITRE Corporation, CVE--2015-1067, https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-1067.
[4] R. Heaton, The SSL FREAK vulnerability explained, http://robertheaton.com/2015/04/06/the-ssl-freak-vulnerability.
[5] Censys, The FREAK Attack. https://censys.io/blog/freak
[6] StackExchange, Protecting phone from the FREAK bug, http://android.stackexchange.com/questions/101929/protecting-phone-from-the-freak-bug/101966.
[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

```
#ifndef OPENSSL_NO_RSA
if (alg_k & SSL_kRSA) {
if ((rsa=RSA_new()) == NULL) {
SSLerr(SSL_F_SSL3_GET_KEY_EXCHANGE,ERR_R_MALLOC_FAILURE);
```

```
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;
  }
if ((rsa=RSA_new()) == NULL) {
SSLerr(SSL_F_SSL3_GET_KEY_EXCHANGE,ERR_R_MALLOC_FAILURE);
```

Server

```
case SSL3_ST_SW_KEY_EXCH_B:
alg_k = s->s3->tmp.new_cipher->algorithm_mkey;
if ((s->options & SSL_OP_EPHEMERAL_RSA)
#ifndef OPENSSL_NO_KRB5
    && !(alg_k & SSL_kKRB5)
#endif )
    s->s3->tmp.use_rsa_tmp=1;
else
    s->s3->tmp.use_rsa_tmp=0;
if (s->s3->tmp.use_rsa_tmp
```

```
case SSL3_ST_SW_KEY_EXCH_B:
alg_k = s->s3->tmp.new_cipher->algorithm_mkey;
s->s3->tmp.use_rsa_tmp=0;
if (
```

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.