Where is my Code Vulnerable: Matching CVEs and Source Code

David A. Barrett & Peter Shin, Canvass Labs

david.barrett@canvasslabs.com https://www.canvasslabs.com

peter@canvasslabs.com

THE LINUX FOUNDATION

Motivation

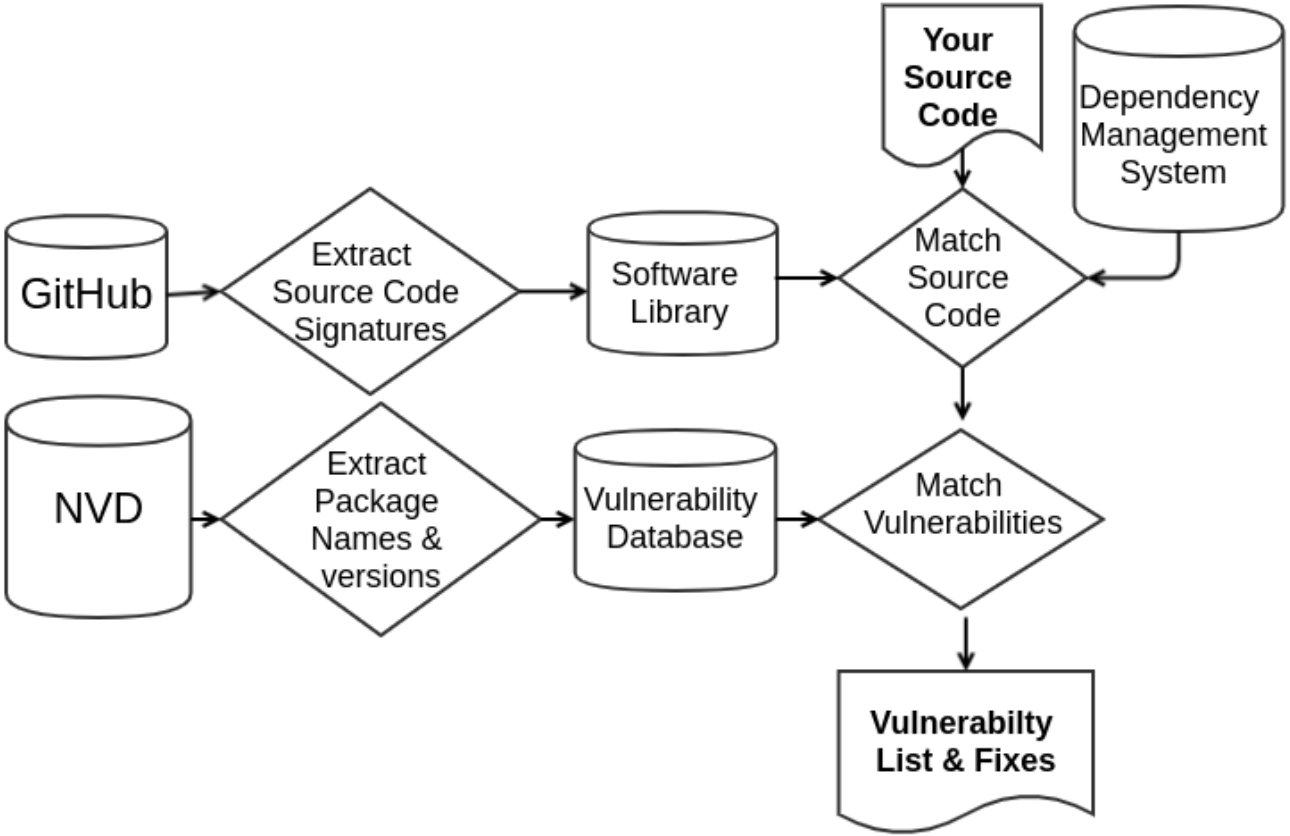
- Modern software depends upon third-party or open source software (or prior versions of their own proprietary software)

- Introduces maintenance costs for developers (or security officers)
  - Track dependencies
  - Monitor dependencies for newly-discovered vulnerabilities
  - Apply patches for updates

Solution

- Create Databases:
  - Find vulnerable code in open source
  - Find package names/versions in NVD
- Scan your source code to find vulnerabilities
- Report the vulnerable code and its fix.

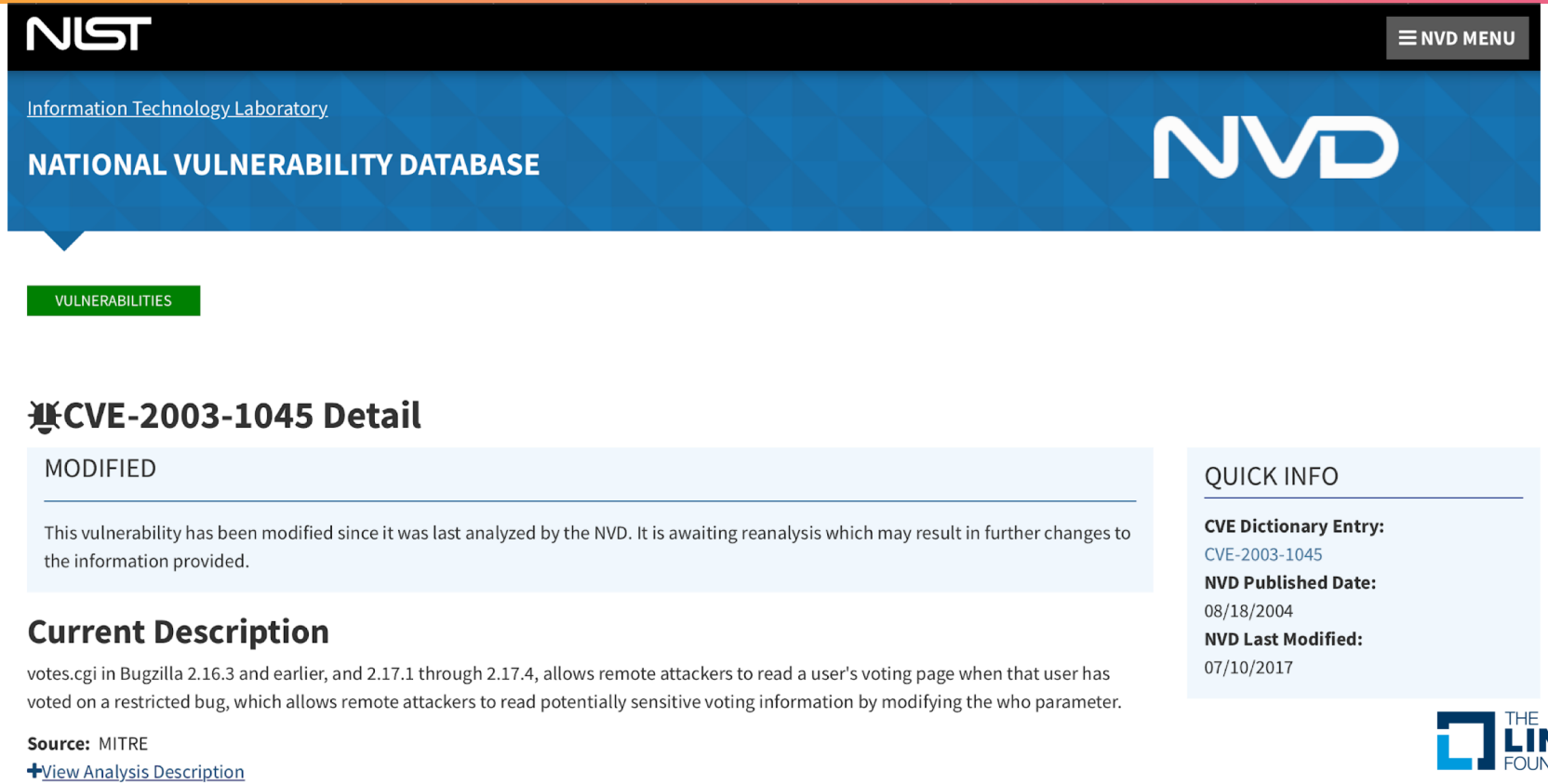
Flow Diagram: Finding Vulnerable Code



Map NVD Description to Package Names and Versions

THE LINUX FOUNDATION

Mapping NVD Description (example)



CVE-2003-1045 Description

votes.cgi in **Bugzilla 2.16.3 and earlier, and 2.17.1 through 2.17.4**, allows remote attackers to read a user's voting page when that user has voted on a restricted bug, which allows remote attackers to read potentially sensitive voting information by modifying the who parameter.

Keyword Tagging

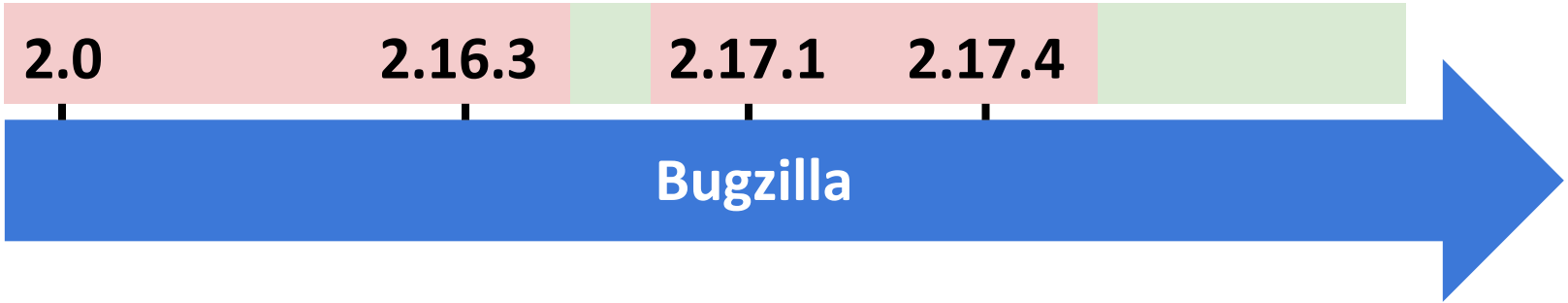
Bugzilla	product name (pn)
2.16.3	version range end (vre)
and	=
earlier	<
,	separator (sp)
and	separator (sp)
2.17.1	version range start (vrs)
through	<=
2.17.4	version range end (vre)

Keyword Interpretation

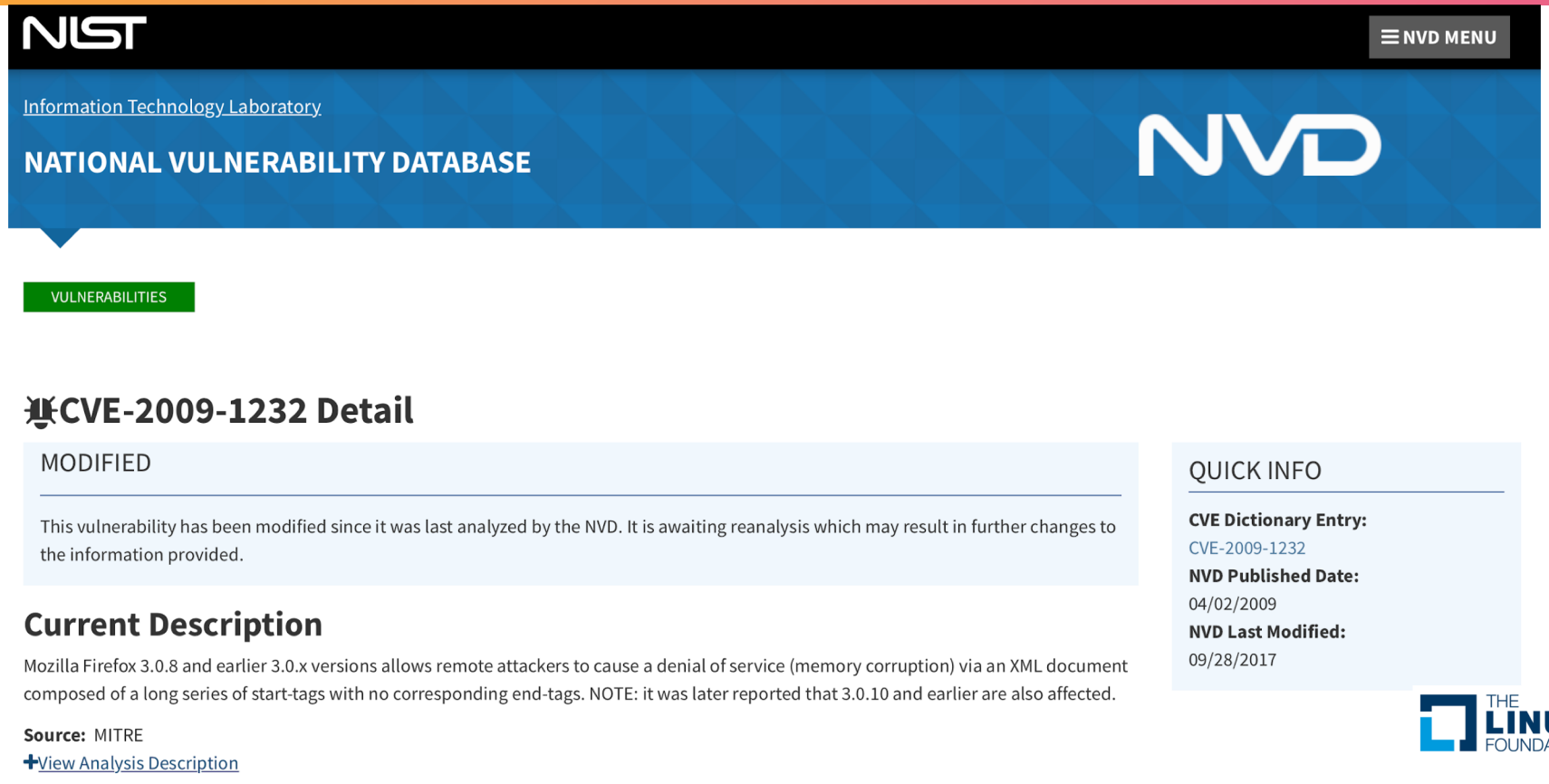
Bugzilla <= 2.16.3

and

2.17.1 <= Bugzilla <= 2.17.4



Example 2


NIST
NVD MENU
Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NVD
VULNERABILITIES
CVE-2009-1232 Detail
MODIFIED
This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.
Current Description
Mozilla Firefox 3.0.8 and earlier 3.0.x versions allows remote attackers to cause a denial of service (memory corruption) via an XML document composed of a long series of start-tags with no corresponding end-tags. NOTE: it was later reported that 3.0.10 and earlier are also affected.
Source: MITRE
View Analysis Description
QUICK INFO
CVE Dictionary Entry:
CVE-2009-1232
NVD Published Date:
04/02/2009
NVD Last Modified:
09/28/2017

CVE-2009-1232 Description

Mozilla Firefox 3.0.8 and earlier 3.0.x versions allows remote attackers to cause a denial of service (memory corruption) via an XML document composed of a long series of start-tags with no corresponding end-tags. NOTE: it was later reported that **3.0.10 and earlier** are also affected.

THE LINUX FOUNDATION

Keyword Tagging (example 2)

Mozilla Firefox 3.0.8 and earlier 3.0.x

pn pn vre = < vr

. . .

3.0.10 and earlier

vre = <

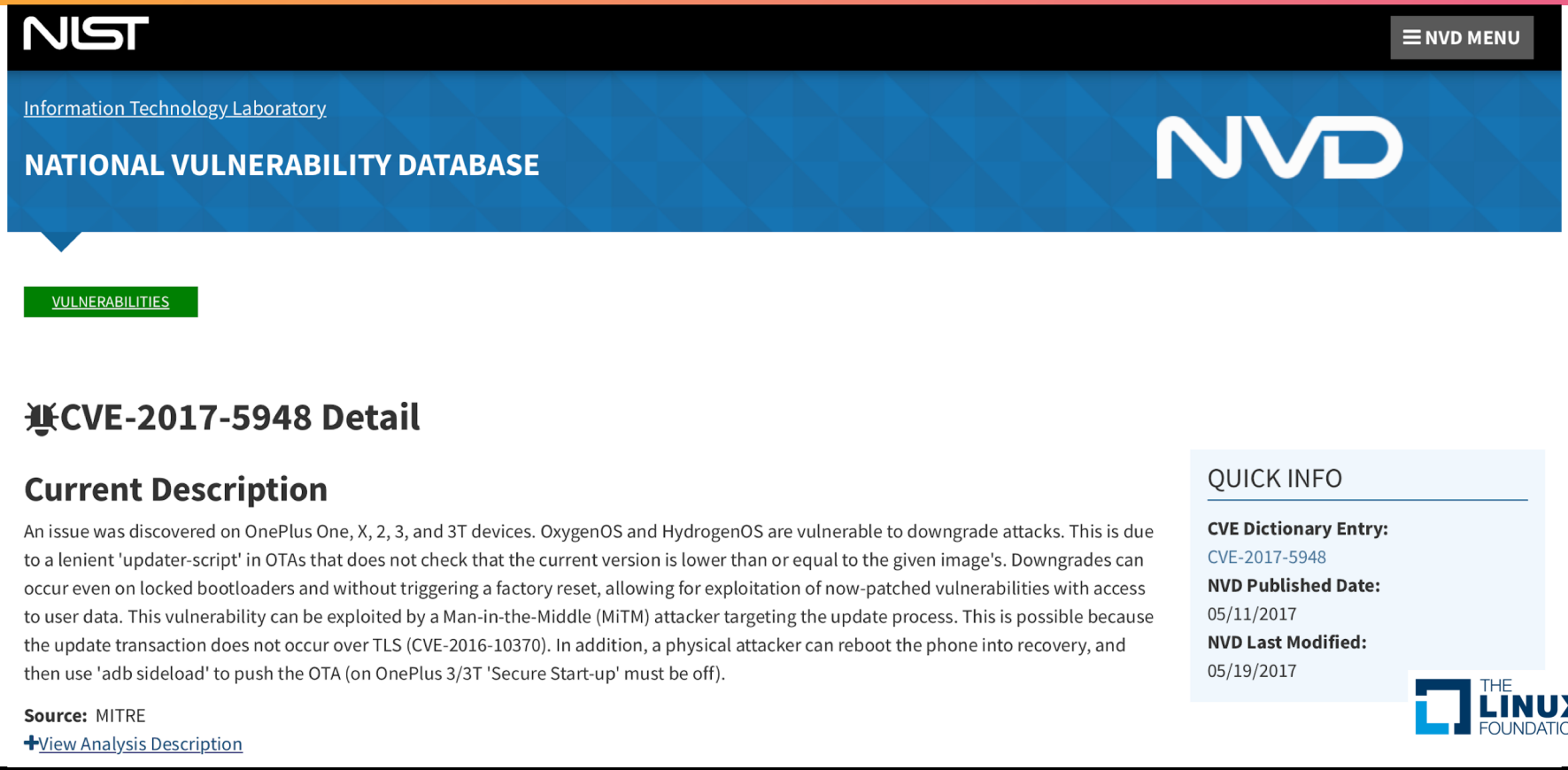
Keyword Interpretation (example 2)

3.0.x <= Mozilla Firefox <= 3.0.8

and

Mozilla Firefox <= 3.0.1

Example 3


NIST
NVD MENU
Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NVD
VULNERABILITIES
CVE-2017-5948 Detail
Current Description
An issue was discovered on OnePlus One, X, 2, 3, and 3T devices. OxygenOS and HydrogenOS are vulnerable to downgrade attacks. This is due to a lenient 'updater-script' in OTAs that does not check that the current version is lower than or equal to the given image's. Downgrades can occur even on locked bootloaders and without triggering a factory reset, allowing for exploitation of now-patched vulnerabilities with access to user data. This vulnerability can be exploited by a Man-in-the-Middle (MiTM) attacker targeting the update process. This is possible because the update transaction does not occur over TLS (CVE-2016-10370). In addition, a physical attacker can reboot the phone into recovery, and then use 'adb sideload' to push the OTA (on OnePlus 3/3T 'Secure Start-up' must be off).
Source: MITRE
View Analysis Description
QUICK INFO
CVE Dictionary Entry:
CVE-2017-5948
NVD Published Date:
05/11/2017
NVD Last Modified:
05/19/2017

CVE-2017-5948 Description

An issue was discovered on **OnePlus One, X, 2, 3, and 3T** devices. **OxygenOS and HydrogenOS** are vulnerable to downgrade attacks. This is due to a lenient 'updater-script' in OTAs that does not check that the current version is lower than or equal to the given image's. Downgrades can occur even on locked bootloaders and without triggering a factory reset, allowing for exploitation of now-patched vulnerabilities with access to user data. This vulnerability can be exploited by a Man-in-the-Middle (MiTM) attacker targeting the update process. This is possible because the update transaction does not occur over TLS (CVE-2016-10370). In addition, a physical attacker can reboot the phone into recovery, and then use 'adb sideload' to push the OTA (on OnePlus 3/3T 'Secure Start-up' must be off).

Keyword Tagging (example 3)

OnePlus	One	,	X	,	2	,	3	,	and	,	3T
pn	pn	sp	pn	sp	pn	sp	pn	sp	sp	sp	pn

. . .

OxygenOS	and	HydrogenOS
pn	sp	pn

Keyword Interpretation (example 3)

Vulnerable products:

- **OnePlus One X**
- **OnePlus One 2**
- **OnePlus One 3**
- **OnePlus One 3T**
- **OxygenOS**
- **HydrogenOS**

Example 4

NIST

NVD MENU

Information Technology Laboratory

NATIONAL VULNERABILITY DATABASE

NVD

VULNERABILITIES

CVE-2012-1463 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

The ELF file parser in AhnLab V3 Internet Security 2011.01.18.00, Bitdefender 7.2, Quick Heal (aka Cat QuickHeal) 11.00, Command Antivirus 5.2.11.5, Comodo Antivirus 7424, eSafe 7.0.17.0, F-Prot Antivirus 4.6.2.117, F-Secure Anti-Virus 9.0.16160.0, McAfee Anti-Virus Scanning Engine 5.400.0.1158, Norman Antivirus 6.06.12, nProtect Anti-Virus 2011-01-17.01, and Panda Antivirus 10.0.2.7 allows remote attackers to bypass malware detection via an ELF file with a modified endianness field. NOTE: this may later be SPLIT into multiple CVEs if additional information is published showing that the error occurred independently in different ELF parser implementations.

Source: MITRE

+View Analysis Description

QUICK INFO

CVE Dictionary Entry:
CVE-2012-1463
NVD Published Date:
03/21/2012
NVD Last Modified:
08/28/2017

THE LINUX FOUNDATION

CVE-2012-1463 Description

The ELF file parser in . . . **Command Antivirus 5.2.11.5, Comodo Antivirus 7424, eSafe 7.0.17.0, F-Prot Antivirus 4.6.2.117, F-Secure Anti-Virus 9.0.16160.0, McAfee Anti-Virus Scanning Engine 5.400.0.1158, Norman Antivirus 6.06.12, nProtect Anti-Virus 2011-01-17.01, and Panda Antivirus 10.0.2.7** allows remote attackers to bypass malware detection via . . .

Keyword Tagging (example 4)

Command Antivirus 5.2.11.5 , Comodo Antivirus
pn pn version sp pn pn

7424 , eSafe 7.0.17.0 , F-Prot Antivirus 4.6.2.117 ,
version sp pn version sp pn pn version sp

F-Secure Anti-Virus 9.0.16160.0 , McAfee Anti-Virus
pn pn version sp pn pn

Scanning Engine 5.400.0.1158 , Norman Antivirus
pn pn version sp pn pn

6.06.12 , nProtect Anti-Virus 2011-01-17.01 , and
version sp pn pn version sp sp

Panda Antivirus 10.0.2.7
pn pn version

Keyword Interpretation (example 4)

- Command Antivirus 5.2.11.5
- Comodo Antivirus 7424
- eSafe 7.0.17.0
- F-Prot Antivirus 4.6.2.117
- F-Secure Anti-Virus 9.0.16160.0
- McAfee Anti-Virus Scanning Engine 5.400.0.1158
- Norman Antivirus 6.06.12
- nProtect Anti-Virus 2011-01-17.01
- Panda Antivirus 10.0.2.7

CVE-2006-0219 Description

The original distribution of MyBulletinBoard (MyBB) to update from older versions to 1.0.2 omits or includes older versions of certain critical files, which allows attackers to conduct (1) SQL injection attacks via an attachment name that is not properly handled by inc/functions_upload.php (CVE-2005-4602), and possibly (2) other attacks related to threadmode in usercp.php.

Finding Vulnerable Code

THE LINUX FOUNDATION

A Vulnerability and its Fix

Patch code:

```
     Hunk: Lines 88-94 (previously 88-91)
88 88  /* remember the value we stored into this reg */

89 89   regs[insn->dst_reg].type = SCALAR_VALUE;

90    -  __mark_reg_known(
           regs + insn->dst_reg, insn->imm);
   90 +  if (BPF_C+LASS(insn->code) == BPF_ALU64) {
   91 +     __mark_reg_known(
              regs + insn->dst_reg, insn->imm);
   92 +  } else {
   93 +     __mark_reg_known(
              regs + insn->dst_reg, (u32)insn ->imm);
91 94    }
```

Package:	linux
File:	*kernel*/bpf/verifier.c
CVE ID:	CVE-2017-16995
Repository:	/kernel
Function:	check_alu_op
CWE_ID:	CWD-119

Vulnerable Source Code

```
int check_alu_op(struct bpf_verifier_env* env ,
      struct bpf_insn*_env insn) {
      struct bpf_reg_state *regs = cur_regs(env)
      ...
      regs[insn->dst_reg].type = SCALAR_VALUE;
      __mark_reg_known(
            regs + insn->dst_reg, insn->imm);
```

We Use its Abstract Syntax

```
TYPE check_alu_op(TYPE PARM,
    TYPE PARM) {
    TYPE LOCAL_VAR = FUNC_CALL(PARM)
    ...
    LOCAL_VAR[PARM] = SCALAR_VALUE;
    FUNC_CALL(
        LOCAL_VAR + PARM, PARM);
```

Vulnerable Source Code

```
int check_alu_op(struct bpf_verifier_env* env ,
      struct bpf_insn*_env insn) {
      struct bpf_reg_state *regs = cur_regs(env)
      ...
      regs[insn->dst_reg].type = SCALAR_VALUE;
      __mark_reg_known(
            regs + insn->dst_reg, insn->imm);
```

To Match Your Altered Source Code:

```
int check_alu_op(struct bpf_verifier_env* env ,
        struct bpf_insn*_env my_insn) {
        struct bpf_reg_state *regs = cur_regs(env)
        ...
        regs[my_insn->dst_reg].type = SCALAR_VALUE;
        __mark_reg_known(
              regs + my_insn->dst_reg, my_insn->imm);
```

And show you it is vulnerable and how to fix it

Example Result

Example Result: Code Matched 3 Vulnerabilities

Souce Code:

```
   /* ssl/t1_lib.c */
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
```

Process

This code has 28 function(s) to review.

NVD database 3

Code repository 29

Package name: openssl
Repository path:
File name: /ssl/t1_lib.c
Function name: tls1_process_ticket
CVE ID: CVE-2016-2177
CVSS Score: 7.5
CWE ID: CWE-190
Patch Code: Show

Package name: android
Repository path: platform/external/openssl
File name: /ssl/t1_lib.c
Function name: tls_decrypt_ticket
CVE ID: CVE-2013-0169
CVSS Score: 2.6
CWE ID: CWE-310
Patch Code: Show

Package name: openssl
Repository path:
File name: /ssl/t1_lib.c
Function name: tls1_process_heartbeat
CVE ID: CVE-2014-0160
CVSS Score: 5.0
CWE ID: CWE-119
Patch Code: Show

THE LINUX FOUNDATION

Example Result: A Matched Vulnerability And Its Fix

Package name: openssl
Repository path:
File name: /ssl/t1_lib.c
Function name: tls1_process_heartbeat
CVE ID: CVE-2014-0160
CVSS Score: 5.0
CWE ID: CWE-119

Patch Code: Hide

Hunk : Lines 6-25 (previously 6-21)			
6	6		unsigned int payload;
7	7		unsigned int padding = 16; /* Use minimum padding */
8	8		
9		-	/* Read type and payload length first */
10		-	hbtype = *p++;
11		-	n2s(p, payload);
12		-	pl = p;
13		-	
14	9		if (s->msg_callback)
15	10		s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
16	11		&s->s3->rrec.data[0], s->s3->rrec.length,
17	12		s, s->msg_callback_arg);
18	13		
	14	+	/* Read type and payload length first */
	15	+	if (1 + 2 + 16 > s->s3->rrec.length)
	16	+	return 0; /* silently discard */
	17	+	hbtype = *p++;
	18	+	n2s(p, payload);
	19	+	if (1 + 2 + payload + 16 > s->s3->rrec.length)
	20	+	return 0; /* silently discard per RFC 6520 sec. 4 */
	21	+	pl = p;
	22	+	
19	23		if (hbtype == TLS1_HB_REQUEST)

THE LINUX FOUNDATION

Example Result: Versions That Are Vulnerable

Package name: openssl
Repository path:
File name: /ssl/t1_lib.c
Function name: tls1_process_heartbeat
CVE ID: CVE-2014-0160
CVSS Score: 5.0
CWE ID: CWE-119

This code has 28 function(s) to review.

NVD database 3

Code repository 29

Package: openssl
Repository path: /
Function name: tls1_process_heartbeat
File name: ssl/t1_lib.c **Versions:** Hide

Version	Function lines
OpenSSL_1_0_1c	2436 2503
OpenSSL_1_0_2-beta1	3789 3856
OpenSSL_1_0_1b	2436 2503
OpenSSL_1_0_1f	2553 2620
OpenSSL_1_0_1e	2481 2548
OpenSSL_1_0_1	2436 2503
OpenSSL_1_0_1d	2473 2540
OpenSSL_1_0_1a	2436 2503

https://github.com/canvasslabs

canvasslabs

Canvass Labs Inc.
La Jolla, CA
Sign in to view email
https://www.canvasslabs.com/

Block or report user

Popular repositories

maven_demo
maven demo project
Java

canvass_for_security-dependency_checker
Finds vulnerabilities in dependent software packages
Python

npm_demo
NPM demo project
JavaScript

canvass_for_security-sample_vuln_db

Bert-on-CVE
Jupyter Notebook

Where is my Code Vulnerable: Matching CVEs and Source Code

David A. Barrett & Peter Shin, Canvass Labs

david.barrett@canvasslabs.com https://www.canvasslabs.com

peter@canvasslabs.com

THE LINUX FOUNDATION