CERIAS Tech Report 2000-17

CERIAS Classic Vulnerability Database User Manual

Guangfeng Song, Salvador Mandujano, Pascal Meunier
Center for Education and Research in
Information Assurance and Security
Purdue University, West Lafayette, IN 47907

1 Introduction

This document describes the vulnerability database (vdb) html, Java and command-line interfaces. For installation and configuration, please refer to the Vulnerability Database installation manual. There are three ways to use the vdb (Fig.1). For read access and searches, the web interface using Perl and Apache is the most convenient. It uses http authentication, and is comprised of a server script and a client script. For inserting records in the vdb or reading existing ones, the Java interface (requiring X-windows) is used. For deleting records, one must use a Perl script from a command-line interface. It is also possible to search from a command-line interface.

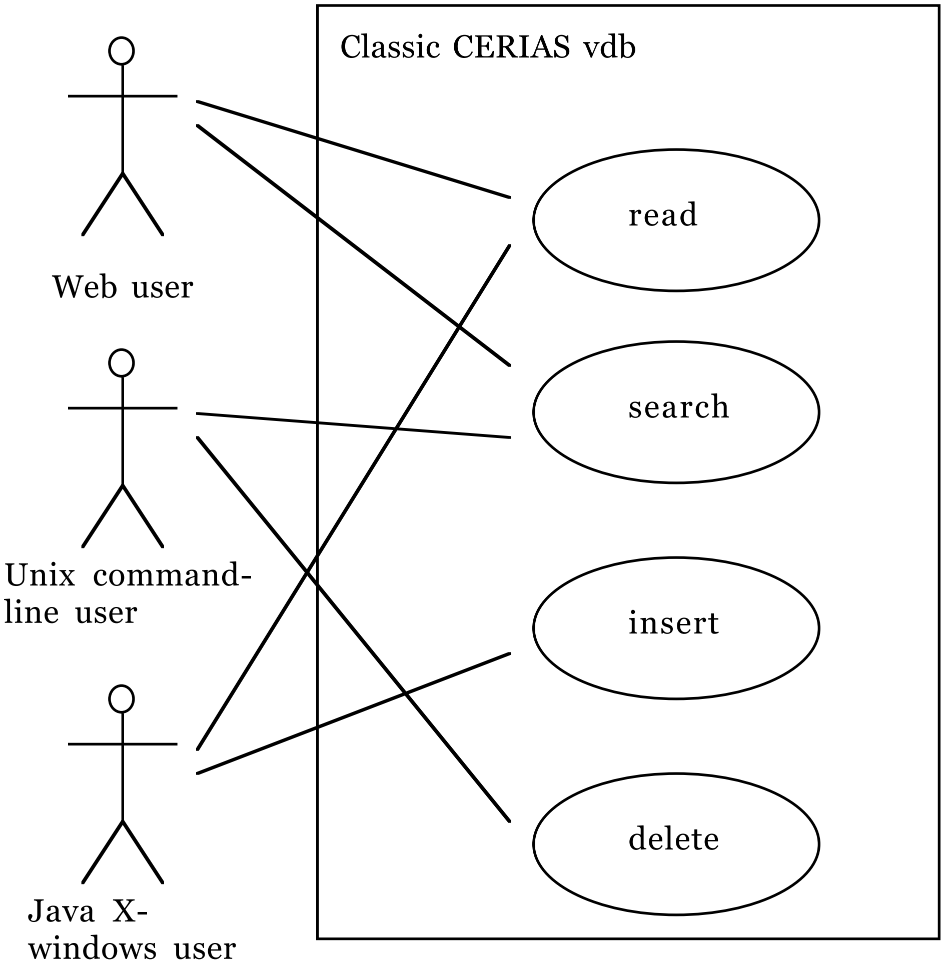


Figure 1. UML use case diagram for different types of users.

2 The WWW Interface

The preferred mechanism for searching the database is the WWW interface. This interface provides an easy-to-use browsing tool that displays records and provides comprehensive search mechanisms. Note that a search server written in Perl must be started before using this interface (see the installation instructions and the UNIX command line interface below). If the WWW interface stops working, the most likely reason is that the Perl script stopped functionning.

2.1 How to start

The WWW interface requires that you point a WWW browser to the URL of your database installation, such as http://amber.cerias.purdue.edu/cgi-bin/vdb. Machines must be authorized to connect to the service. Users will also need a user name and a password to access the database. Manuals are available from the Main page of the WWW interface.

2.2 How to find records

By clicking on the Search link on the Main page, you can perform the following types of searches:

Search Titles:
Simply input any string (including part of a word) to be matched against the title of vulnerabilities. The server will list alphabetically the matching vulnerabilities.

Search Records:
Type in any string to be searched in the content of database records. The matched results (records that contain this string) will be listed alphabetically with the related field partially displayed.

Search by Perl Regular Expression:
A Perl regular expression search will be performed on the database and prints in formatted HTML the search results. Note that special characters in the regular expression are escaped because they are a problem in the web server. Hence, the search server converts these characters back to normal before using the regular expression (with one notable exception: the back tick.).

Besides searching for specific strings, you can also click the List Records link on the Main page to display all the records in the database alphabetically.

2.3 Dump Records

Click the "Dump Records" link on the Main page to list all the records in raw text, which can be saved for further data processing.

3 The UNIX command-line interface

You will need to locate the Perl directory ~/vdbase/perl to run the following commands.

3.1 Searching the Database

The script pattern match.pl takes as arguments a series of words and searches the vulnerability database for matches on those keywords. It loads the database index and calls the fgrep program to search every file called V in the ~/vdbase/vdb/ directory. Searching the database this way can bring any machine to its knees (every search opens thousands of files) so we do not recommend that you use it unless you have no other choice.

The same directory contains two other programs that are particularly useful for searching the database. The first is a program called searchServer.pl and it essentially loads the entire database to memory and does a pattern search using Perl. The server will check to see if new records have been added or if records have been modified every thirty seconds and will load/reload as needed.

The second program is called searchClient.pl and it contacts the server and gives it a string to search for and displays whatever the search server returns.

The match is similar to fgrep in that the entire text passed has to be matched. Unlike fgrep, however, the search string is altered a little bit before used in the server. In its output the search server prints the results with very simple formatting so that what you see in the screen is not likely to be in the same format as what you see in the database.

To run the search server, type:

```
% ./searchServer.pl -p 6768
```

Where the -p option tells it which port to use.

To run the client, type:

```
% ./searchClient.pl -p 6768 -k "String to search for"
```

Where the -p option tells which port the server should use. Be warned that the server is not a full-blown daemon and should not be run in the background of an obscured window. Create a separate xterm window for it and look at it periodically to make sure that no errors are being ignored.

3.2 Deleting a record

The name of the script that you need to use is "removerecord".

i. Go to the ~/vdbase directory
ii. Execute the script passing as only parameter the name of the vulnerability you want to remove. For instance:

```
% removerecord lpr_buffer_overrun
```

The script will remove all the subdirectories and files related to that name.

iii. Finally, edit the /home/projects/proj-vdb/vdbase/vdb/Vulnerabilities file, and remove the line with the name of the vulnerability. The output of the program should look something like this:

```
earthsea 71 % removerecord mental_overflow_sam
Removing field:     access_required
Removing field:     category
Removing field:     class
Removing field:     complexity_of_exploit
Removing field:     dataentry
Removing field:     desc
Removing field:     direct_impact
Removing field:     indirect_impact
Removing field:     krsul_class
Removing field:     modifications
Removing field:     os_type
Removing field:     source_addres
Removing field:     system
Removing field:     system_verbatim
Removing field:     system_version
Removing field:     thac_cavail
Removing field:     thac_conf
Removing field:     thac_create
Removing field:     thac_destroy
Removing field:     thac_disclose
Removing field:     thac_exec
Removing field:     thac_integrity
Removing field:     thac_misrep
Removing field:     thac_modify
Removing field:     thac_observe
Removing field:     thac_repudiate
Removing field:     title
Removing field:     vendor
```

4 The Java Interface

4.1 Recovery from crash

The Java GUI may crash, especially in circumstances where the drop-down menus are used before the previous operation has completed. In addition, it implements record locking to allow multiple people access to the database at the same time. If the program crashes before it released the locks on the records you were editing (which happens when you save or when you exit), it is possible that the next time you use the program you may have to clean the locks by hand. The program will tell you how to do that, but be sure that the lock that you clean by hand is yours! It is possible that someone else may be editing that record. You can do that by checking the ownership of the lock-file it created.

4.1 Running the GUI Interface

There is a shell script that sets the Java class path and runs the GUI interface. We recommend that to run the interface by creating the following alias: "alias vdbJava $JAVAGUI/runvdbgui."

Please, do not run the database using another command! The script makes sure that your umask is set to 007 and hence the files created by the database will have the correct permissions. The script also makes sure that your classpath contains all the packages needed to run the system. If your umask is not 007 then it is possible that the files you create using the Java GUI will not be readable or writable by anyone else. This has the potential for breaking lots of things.

4.2 Java Interface Menu

In the Java interface, vulnerability entries are listed on the left of the screen. Double clicking on an item on a entry will display the contents of that vulnerability on the right of the screen.

There's a drop-down menu for the interface. The File menu has 5 options.

1) About: for displaying product information
2) Save Changes: for saving your changes to the database. You must invoke this menu items to have your changes saved after adding or editing database entries. Otherwise, your changes will be lost. Note that when performing any change or adding a new entry, changes
should be saved twice:

i. In the edition screen, click "Save Changes". This saves the
   changes to RAM memory.
ii. In the main screen, under the "File" menu, click "Save Changes"
   to actually write the changes to disk.

3) Print item to file: for exporting the current record to a pure text file.
4) MIME export record: for exporting the record as a multi-part MIME file.
5) Exit: for closing the java interface

The View menu has 4 options.
1) Display Fields: for indicating to the GUI interface to eliminate from the record display selected fields. This is particularly useful when fields such as patches and exploit scripts clutter the screen and the user wishes to view records without displaying these fields. When printing records the interface will also only print the fields as indicated by this menu.
2) Show classifiers: for displaying the classifiers used for the database
3) Classifier description: for explaining classifiers
4) Rating system description: for explaining the field rating system

The Edit menu has 3 options.
1) Edit Selected Entry: for editing the current record in the database
2) Add New Entry: for adding new records to the database. When you create a new record, a dialog is presented to the user requesting a record ID and title for the new vulnerability. Once this information is presented a new blank record is created for that vulnerability.

3) Deleting an entry
This is not implemented in the Java interface. One must use the command line interface (see 3.2).

4.3 Entering MIME parts

All text fields where you can type information can have MIME parts inserted within the text1 MIME parts are manipulated by using the following keyboard commands while in the text field:

<control-i>: Insert textual mime part. Opens a dialog that allows the user to type or paste text into the field and insert it as a MIME part. Important Note: The editor is not smart enough, nor it should be, to notice that you have inserted a MIME part and that it should remove the corresponding file if you decide to discard your changes to the record. Hence, if you add a MIME part and then discard your changes to the record you will have a MIME part file in the $MIMEINCLUDES directory that will not be referenced by any record. Hence, delete the MIME parts created manually before discarding your changes to the record if you want the MIME parts to be discarded too!

<control-d>: Delete MIME part. This option deletes the MIME part where the cursor is located. The MIME include directive is removed from the text and the MIME part file is deleted from the file system.

Fields that have associated classifiers can also have MIME parts. However, we don't recommend that this be done as there are utilities will not work correctly in this case.

<control-e>: Edit MIME part. If the MIME part is editable then this command allows the user to edit the part in a special MIME part editor.

<control-v>: View the MIME part. Displays the content of the MIME part in a special window. <control-x>: Export MIME part. Not implemented yet! Allows the user to export this part to a multi-part MIME file that can be viewed with an external viewer or that can be send via email.

<control-m>: View part with an external viewer. If the MIME part is not a textual part then it cannot be viewed using the control-v command. This command saves exports the part as a temporary file and calls an external MIME viewer to display the part.

<control-f>: MIME encode a file. This command opens a file dialog box and lets the user select an external file that must be MIME encoded and saved to a MIME part for the record. Once the file is selected, the interface will attempt to guess the MIME type and will open a dialog box to confirm that the type selected is indeed correct. If it is not, then select the correct type and proceed with the conversion.

MIME parts are highlighted in the main window and can be viewed by double clicking on the name of the included part. Bug Note: Under some window managers in UNIX, a double click is defined as two Fields that have associated classifiers display the classifier name in parenthesis under the field name. Click in the field name to display the allowable values for the field.

5 Field descriptions and classification procedures

This section describes the fields of the database. The value of the database depends on Objectivity, Determinism, Repeatability and Specificity.

Objectivity: the features must be identified from the object known and not from the subject knowing

Determinism: there must be a clear procedure that can be followed to extract the feature

Repeatability: several people extracting the same feature for the object must agree on the value observed

Specificity: the value for the feature must be unique and unambiguous.

In this database, decision trees are used to limit the ambiguity associated with certain fields. In addition, a rating system helps to quantify the certainty on the value of a field.

5.1 Classifiers and rating system

When editing a record the interface will open a new window, shown in Figure 2, that contains fields and pop-up menus for entering data. Fields that have classifiers are marked by including the name of the classifier in parenthesis under the field name and you can display the classifier by clicking on the name of the field. If the field has a classifier and is a text field, then you are not required to enter data that matches the classifiers. However, the GUI will complain about it and we strongly recommend that you do stick with pre-defined choices.

Some fields in the database have associated confidence ratings that give users an idea of how reliable is the data for that particular field. The rating system is as follows:

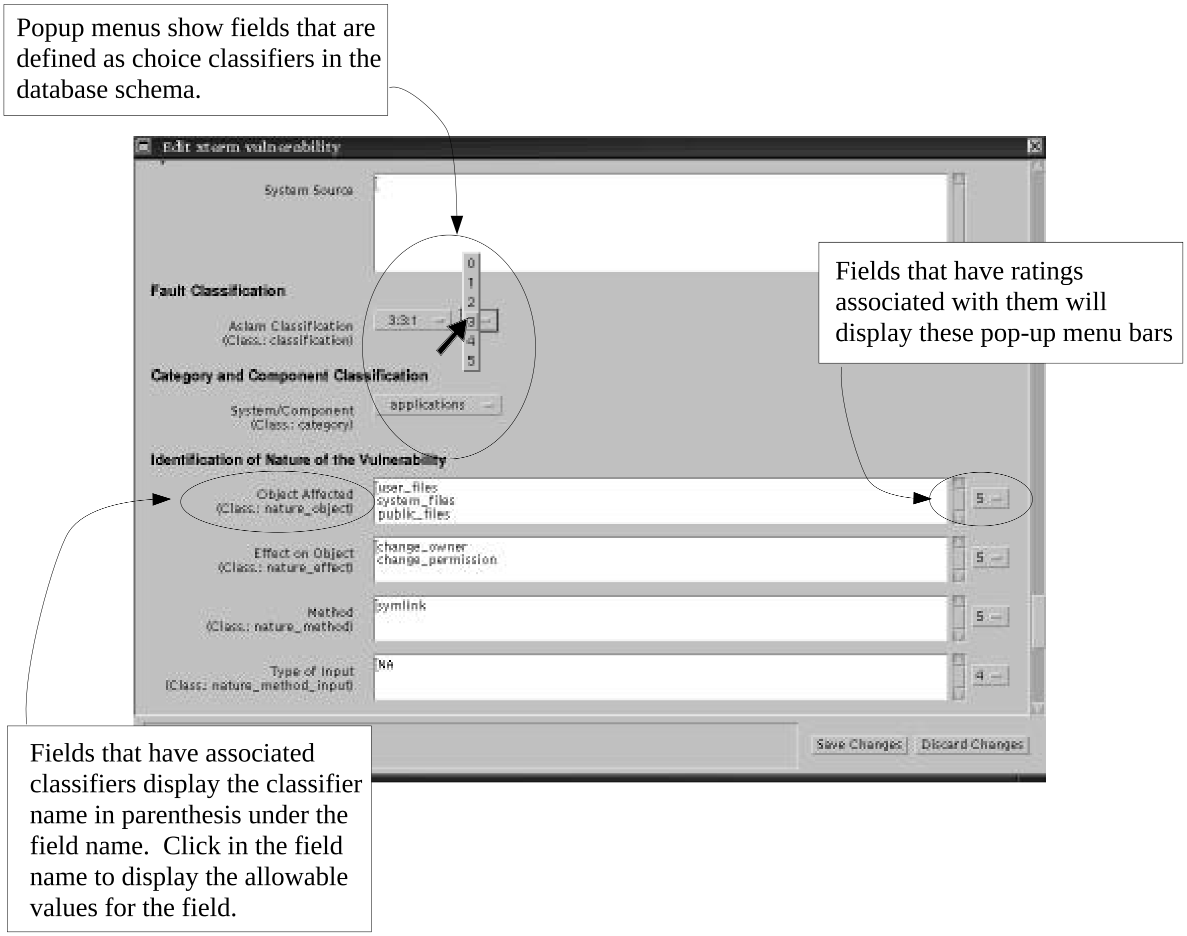
Value of 0: Item has not been rated. Users will generally make no assumptions about the information in this field.

Items with a rating of 0 should not be trusted or used to justify any results.

Value of 1: Item is likely to be a guess or speculation.

Value of 2: Item is not likely to be correct and limited trust should be put on it.

Value of 3: Item is likely to be only partially correct, may contain errors, may be incomplete, etc.



Popup menus show fields that are defined as choice classifiers in the database schema.

Fields that have ratings associated with them will display these pop-up menu bars

Fields that have associated classifiers display the classifier name in parenthesis under the field name. Click in the field name to display the allowable values for the field.

Edit xterm vulnerability

System Source

Fault Classification

Aslam Classification (Class.: classification): 3:3:1, rating 3

Category and Component Classification

System/Component (Class.: category): applications

Identification of Nature of the Vulnerability

Object Affected (Class.: nature_object): user_files, system_files, public_files; rating 5

Effect on Object (Class.: nature_effect): change_owner, change_permission; rating 5

Method (Class.: nature_method): symlink; rating 5

Type of Input (Class.: nature_method_input): NA; rating 4

Figure 2. Rating of classifier confidence

Value of 4: Item seems to be correct but has not been verified by a trusted party. The operator that entered this information, to the best of his knowledge, believes the information to be accurate.

Values of 5: Item is correct and has been verified by a trusted entity. The operator has evidence that the item is correct and can guarantee, with a high probability, that the item contains accurate and complete information.

When entering data you should be especially careful to enter the appropriate rating for the data that you are entering. Leaving that rating at its default value of zero will cause the data that you are entering to be ignored in automatic processes.

The following figures show the classifiers in the order that will be needed in the Java interface:

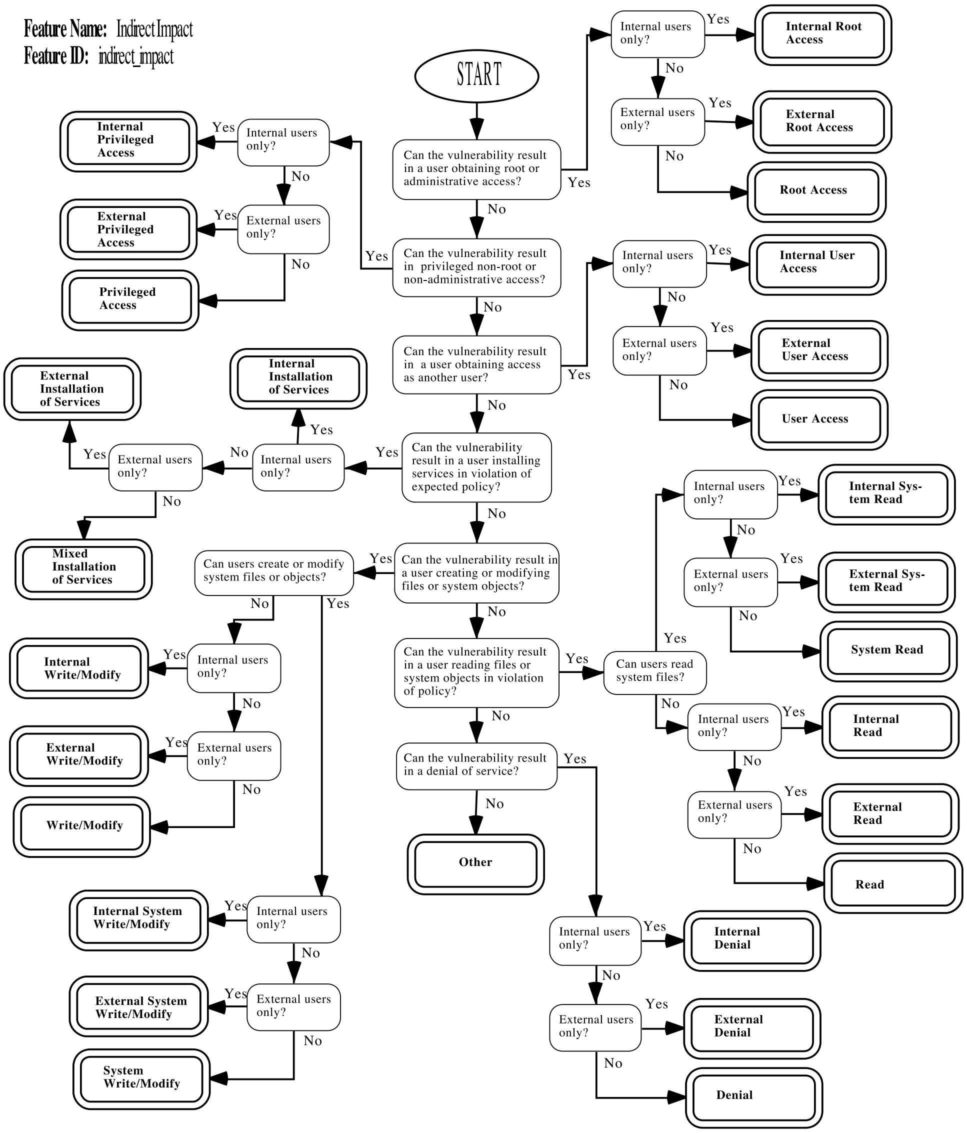
i. Indirect impact

This feature attempts to identify the indirect or ultimate impact of the vulnerability. This is the worst possible thing that can happen some time after the exploitation of the vulnerability, barring the exploitation of other vulnerabilities.

VULNERABILITY DATABASE FEATURE SELECTION CRITERIA

Feature Name: Indirect Impact

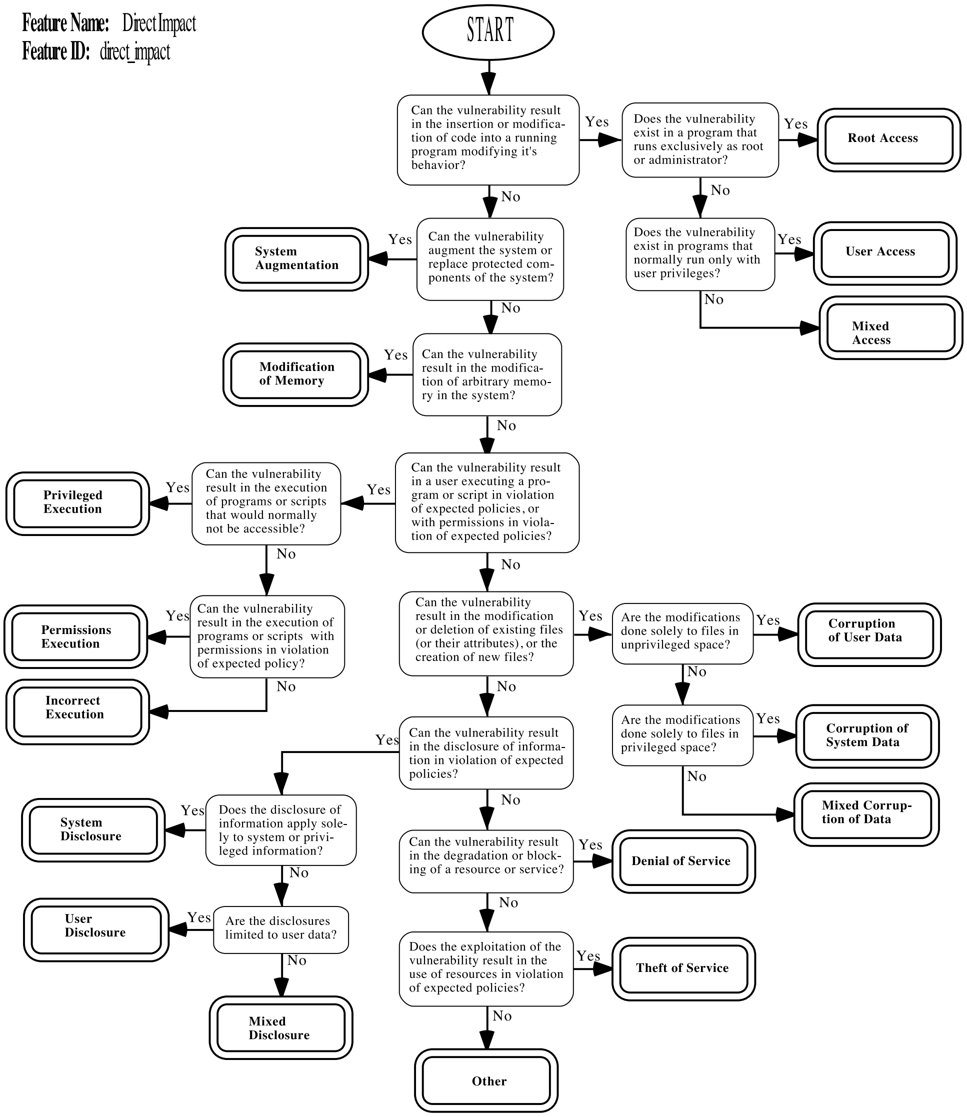
Feature ID: indirect_impact


START
Can the vulnerability result in a user obtaining root or administrative access?
Yes
Internal users only?
Yes
Internal Root Access
No
External users only?
Yes
External Root Access
No
Root Access
No
Can the vulnerability result in privileged non-root or non-administrative access?
Yes
Internal users only?
Yes
Internal Privileged Access
No
External users only?
Yes
External Privileged Access
No
Privileged Access
No
Can the vulnerability result in a user obtaining access as another user?
Yes
Internal users only?
Yes
Internal User Access
No
External users only?
Yes
External User Access
No
User Access
No
Can the vulnerability result in a user installing services in violation of expected policy?
Yes
Internal users only?
Yes
Internal Installation of Services
No
External users only?
Yes
External Installation of Services
No
Mixed Installation of Services
No
Can the vulnerability result in a user creating or modifying files or system objects?
Yes
Can users create or modify system files or objects?
No
Internal users only?
Yes
Internal Write/Modify
No
External users only?
Yes
External Write/Modify
No
Write/Modify
Yes
Internal users only?
Yes
Internal System Write/Modify
No
External users only?
Yes
External System Write/Modify
No
System Write/Modify
No
Can the vulnerability result in a user reading files or system objects in violation of policy?
Yes
Can users read system files?
Yes
Internal users only?
Yes
Internal System Read
No
External users only?
Yes
External System Read
No
System Read
No
Internal users only?
Yes
Internal Read
No
External users only?
Yes
External Read
No
Read
No
Can the vulnerability result in a denial of service?
Yes
Internal users only?
Yes
Internal Denial
No
External users only?
Yes
External Denial
No
Denial
No
Other

ii. Direct impact

This is the immediate threat posed by the atomic exploit of (i.e., the smallest action that exploits) the vulnerability. Direct impacts happen immediately upon the exploitation of the vulnerability.

Feature Name: Direct Impact
Feature ID: direct_impact


START
Can the vulnerability result in the insertion or modification of code into a running program modifying it's behavior?
Yes
Does the vulnerability exist in a program that runs exclusively as root or administrator?
Yes
Root Access
No
Does the vulnerability exist in programs that normally run only with user privileges?
Yes
User Access
No
Mixed Access
No
Can the vulnerability augment the system or replace protected components of the system?
Yes
System Augmentation
No
Can the vulnerability result in the modification of arbitrary memory in the system?
Yes
Modification of Memory
No
Can the vulnerability result in a user executing a program or script in violation of expected policies, or with permissions in violation of expected policies?
Yes
Can the vulnerability result in the execution of programs or scripts that would normally not be accessible?
Yes
Privileged Execution
No
Can the vulnerability result in the execution of programs or scripts with permissions in violation of expected policy?
Yes
Permissions Execution
No
Incorrect Execution
No
Can the vulnerability result in the modification or deletion of existing files (or their attributes), or the creation of new files?
Yes
Are the modifications done solely to files in unprivileged space?
Yes
Corruption of User Data
No
Are the modifications done solely to files in privileged space?
Yes
Corruption of System Data
No
Mixed Corruption of Data
No
Can the vulnerability result in the disclosure of information in violation of expected policies?
Yes
Does the disclosure of information apply solely to system or privileged information?
Yes
System Disclosure
No
Are the disclosures limited to user data?
Yes
User Disclosure
No
Mixed Disclosure
No
Can the vulnerability result in the degradation or blocking of a resource or service?
Yes
Denial of Service
No
Does the exploitation of the vulnerability result in the use of resources in violation of expected policies?
Yes
Theft of Service
No
Other

iii. Nature of threat. This refers to the immediate risks that the vulnerabilities present (much like the direct impact). Each of these features can take the values "Yes:' 'No ' "Does Not Apply" and "Unknown". Hence, each feature is a decision tree with a depth of one that has a single fundamental division. They are divided in two sets, actions and consequences:

Actions:

thac_observe: The vulnerability can result in a user observing objects, data, etc., in violation of expected policy.

thac_destroy: The vulnerability can result in a user destroying objects, data, etc., in violation of expected policy.

thac_modify: The vulnerability can result in a user modifying objects, data, etc., in violation of expected policy

thac_create: The vulnerability can result in a user creating objects in violation of expected policy.

thac_exec: The vulnerability can result in a user executing a program in violation of expected policy.

Consequences:

thac_cavail: The vulnerability can result in the change of availability of the system

thac_disclose: The vulnerability can result in the disclosure of information in violation of expected policy.

thac_misrep: The vulnerability can result in misrepresentation of information

thac_repudiate: The vulnerability can result in repudiation of information

thac_intergrity: The vulnerability can result in a change of integrity of the system

thac_config: The vulnerability can result in the loss of confidentiality of information

iv. System

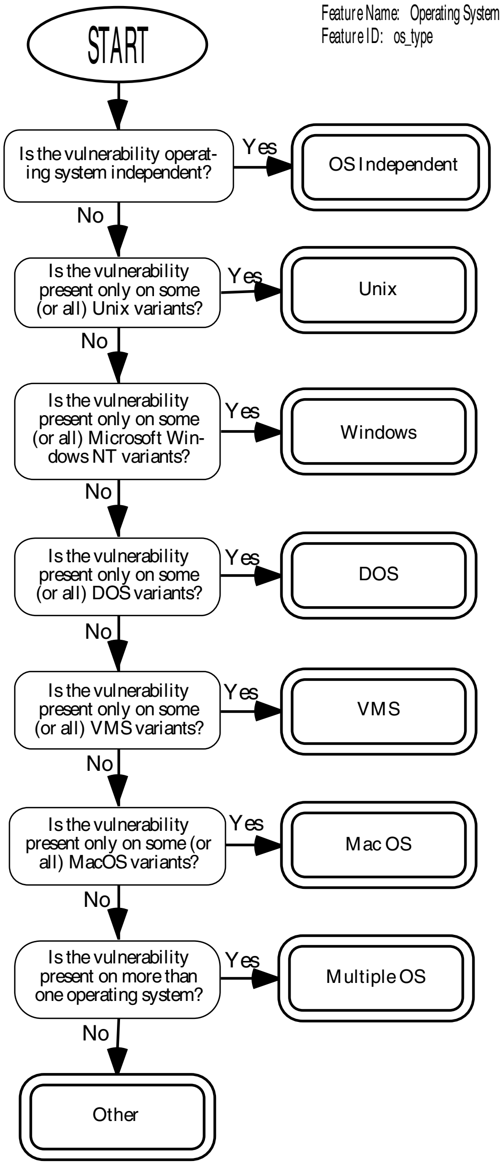
This classifier is used to indicate the systems that are known (to us!) to have the vulnerability. To date we have recorded vulnerabilities for the following operating system.

Solaris->SUN Solaris
SGIRIX -> SGI IRIX
SunOS-> SUN OS
DECOSF1-> Digital.OSF/1
DOS-> Microsoft DOS
NECUX-> NEC XX-UX
Windows 95->Microsoft Windows 95
HP-UX->Hewlett-Packard Unix
Windows NT->Microsoft Windows NT
AIX-> IBM's AIX
Windows WG->Microsoft Windows(pre-95)
OpenStep->OpenStep
Slackware->Linux Slackware
OSF->OSF
Redhat->Linux Redhat
Caldera->Caldera
Debian->Linux Debian
Goah->NEC's Goah
Mklinux->Linux Apple Distribution
Ultrix->Ultrix
OpenLinux->Linux Caldera distribution
DEC_UNIX->Digital Unix
OtherLinux->Unknown, unsupported or uncommon Linux
AUX->Apple's Unix
BSDI->BSDI Unix
DG->Data General
NovellUnix->Novel Unixware
unicos->Cray's UNICOS
NetBSD->NetBSD Unix
MacOS->Macitosh OS
FreeBSD->Free BSD Unix
Netware-> Novell Netware
Athena->MIT-distributed athena
OpenBSD->OpenBSD Unix
Cygnus->Cygnus Network security
VMS->DEC VMS
OpenVision->Open vision
NA->Does not apply

v. Vendors
SUN-> Sun Microsystems, Inc
Microsoft-> Microsoft
SGI ->Sillicon Graphics Inc
Netscape-> Netscape Corporation
BSDI-> Berkeley Software Design, Inc.
Slackware-> Walnut CreckCDROM
Redhat -> Redhat Software, Inc.
Debian -> Software in the Public Interest (SPI)
MkLinux -> Apple Computer
DGC -> Data General Corporation
FreeBSD -> FreeBSD, Inc
HP -> Hewlett-Packard Company
IBM ->IBM Corporation
SCO-> The Santa Cruz Operation, Inc
NeXT -> NeXE Software, Inc.
OpenGroup-> The Open Group
SantaCruz ->The Santa Cruz Operation (SCO)
Caldera-> Caldera
DEC -> Digital Equipment Corporation
Apple-> Apple Computer
OSF-> Open Software Foundation
CRAY -> Cray
NetBSD -> Ihe NetBSD Project
OpenBSD -> The OpenBSD Project
Novell-> Novell
NA -> Does not apply

vi. Type of OS

Feature Name: Operating System Type
Feature ID: os_type


START
Is the vulnerability operating system independent?
Yes
OS Independent
No
Is the vulnerability present only on some (or all) Unix variants?
Yes
Unix
No
Is the vulnerability present only on some (or all) Microsoft Windows NT variants?
Yes
Windows
No
Is the vulnerability present only on some (or all) DOS variants?
Yes
DOS
No
Is the vulnerability present only on some (or all) VMS variants?
Yes
VMS
No
Is the vulnerability present only on some (or all) MacOS variants?
Yes
Mac OS
No
Is the vulnerability present on more than one operating system?
Yes
Multiple OS
No
Other

vii. Application

This feature defines the application that has the vulnerability. This classifier is relevant for those vulnerabilities that are present in user-level programs, daemons, servers,. etc. that are not a part of the operating system itself. This feature can take on many values and here we give a small subset as examples.

Netscape-> Netscape WWW Browser

HotJava-> SUN's HotJava WWW Browser

JDK appviewer -> Java Developer Kit's appler viewer

Ora_pbrow -> Oracle PowerBrowser

XMCD -> CD digital .audio player utility for X1 1/Motif

NIS -> Network Information. System

Apache -> Apache WWW httpd .

FrontPage-> Microsoft FrontPage

InternetExploer Microsoft Explorer

NetscapeNewsServer-> ,Netscape's News Server

Minicom -> Linux free telecom

NTHTTPServer -> HTTP Server included in Windows NT

rpcbind -> Universal addresses to RPC program number mapper

rlogin-> Remote login

stat-> File status

ftpd-> Internet File Transfer Protocol server (ftpd)

talkd -> Serverfortalk program (talkd)

ps -> Report process status ps

rmail-> Read mail program in Unix

lpr-> Unix lpr Send a job to the printer

ircd -> IRC Server

NCSAhttpd NCSA WWW httpd

pkgtool-> PKGTOOL Linux Software Management Utility

syediag-> HP System Diagnostics tool e majordomo

Majordomo-> mailer.

passwd-> Unix password change utility

binmail -> /uar/bin/mail on Unix

rdist-> Remote file distribution client

ppp-> Implementation of the Point to point protocol for TCP/IP

sperl -> SetUID Perl

xterm-> Terminal emulator for X

cxterm-> Chinese Terminal emulator for X

admintool-> Sun administration tool

inperson -> InPerson desktop video con ferencing package

lynx -> Lynx text web browser

swinstall-> HPUX software installation utility

glance-> HPUX Glance software

workman-> Workman CD digital audio player

lpd -> Line printer daemon

sendmail-> Unix program for sending email over the Internet

lmgrd -> FLEXlm license manager daemon
expreserve -> vi and ex file preservation utility
crontable->System clock daemon manager for users
ld. so -> runtime linker used by dynamically linked executables (a.out)
telnetd ->DARPA TELNET protocol sever
norton-> Norton Utilities . "
fm_fls-> FrameMaker license server .
usrmgr -> NT user manager
voLd -> Solaris volume mounting daemon
mstimeserv ->NT Time Server
kcms -> Kodak Color Management System
msaccess-> Microsoft Access
wuftpd ->Washington University ftpd,
msoffice -> Microsoft Office
rpcmountd -> rpc mount daemon
iind -> Internet News daemon
df -> Disk space reporting command .
ordist ->IRIX version of rdist
pset -> IRIX processor set modification utility
chkey -> RPC change key utility
cdplayer -> SGI CD digital audio player
fpkg2swpkg-> HP product spec. conv. utility
newgrp ->Program to create a new group
bash ->GNU Project's Bourne Again SHell
BIND -> Berjkeley Internet Name Domain
Elm->Elm Mall System
testcgi -> A script that return status of the cgi systems on http daemons

viii. Ease of exploit

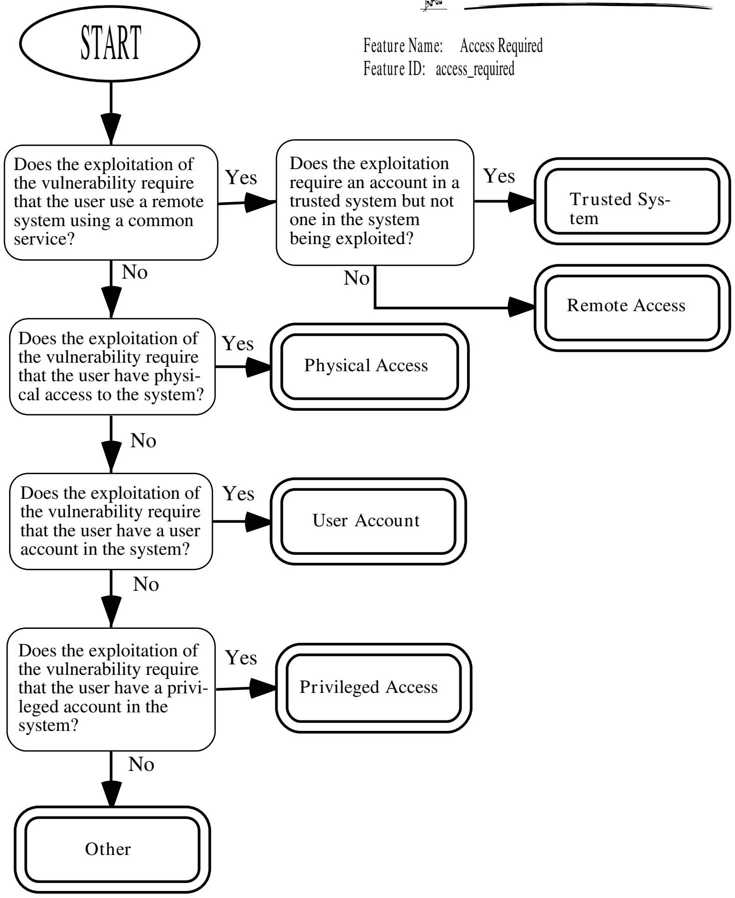
This classifier was originally defined from a talk given by Tom Longstaff [Lon97] and attempts to identify how easy (or how hard) it is to exploit the vulnerability.

simple -> Simple command
toolkit-> Toolkit available
expertise -> Expertise required
user -> Must convince a user to take an action
Administrator-> Must convince an administrator to take an action

ix. Access required

This classifier was originally defined from a talk given by Tom Longstaff [Lon97] and define the kind of access that is required to exploit the vulnerability.

Feature Name: Access Required
Feature ID: access_required


START
Does the exploitation of the vulnerability require that the user use a remote system using a common service?
Yes
Does the exploitation require an account in a trusted system but not one in the system being exploited?
Yes
Trusted System
No
Remote Access
No
Does the exploitation of the vulnerability require that the user have physical access to the system?
Yes
Physical Access
No
Does the exploitation of the vulnerability require that the user have a user account in the system?
Yes
User Account
No
Does the exploitation of the vulnerability require that the user have a privileged account in the system?
Yes
Privileged Access
No
Other

x. Complexity of exploit

This feature attempts to identify the complexity of the exploitation of a vulnerability, regardless of whether a script or toolkit exists for the exploitation of the vulnerability.

1. The notion of a simple sequence of commands will, of course, vary from person to person. We will consider a simple sequence of commands a linear sequence of commands (i.e. no loops, gotos, etc.) of no more than a dozen commands. Also, these dozen commands must be common commands supported by the operating system, common applications and utilities. Commands that involve scripts and applications that the exploiter must compile, install, etc., do not qualify.

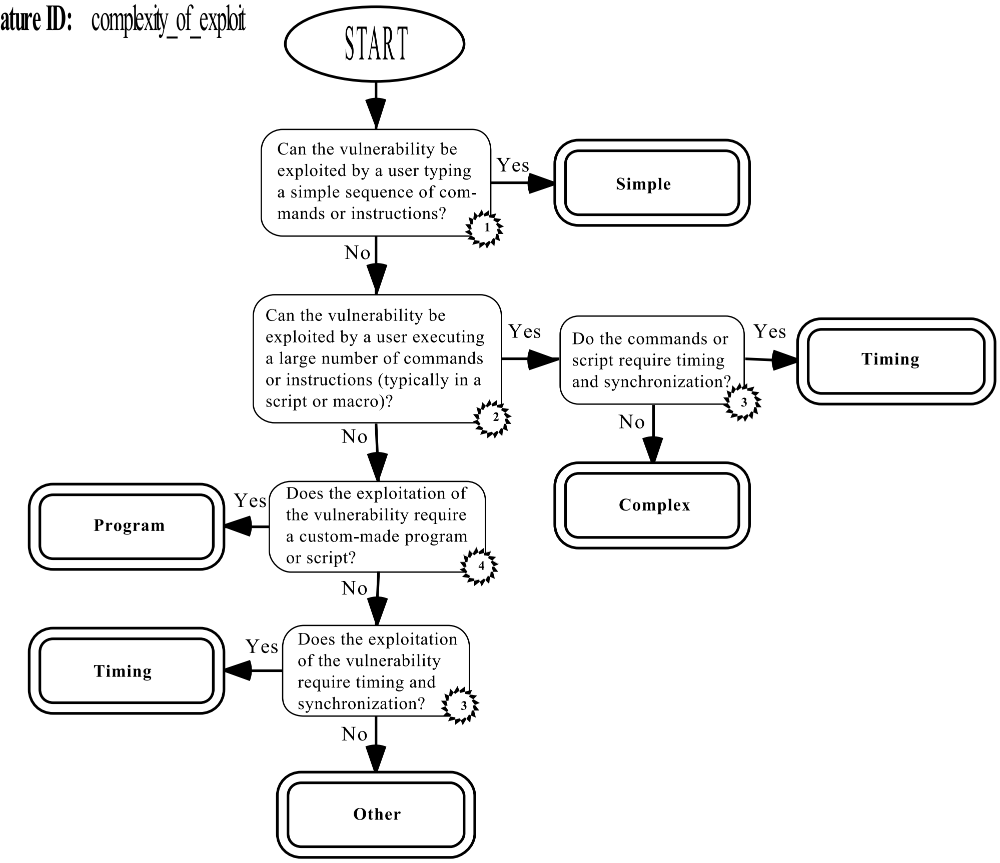
2. Shell scripts, command interpreter source files and macros all qualify. Programs that are implemented in a general purpose programming language (including such languages as Perl) do not qualify.

3. Typically requires a script or application that tries several times and may require slowing down the system.

4. Applications that the exploiter must compile, install, etc.

Feature Name: Complexity of Exploit

Feature ID: complexity_of_exploit

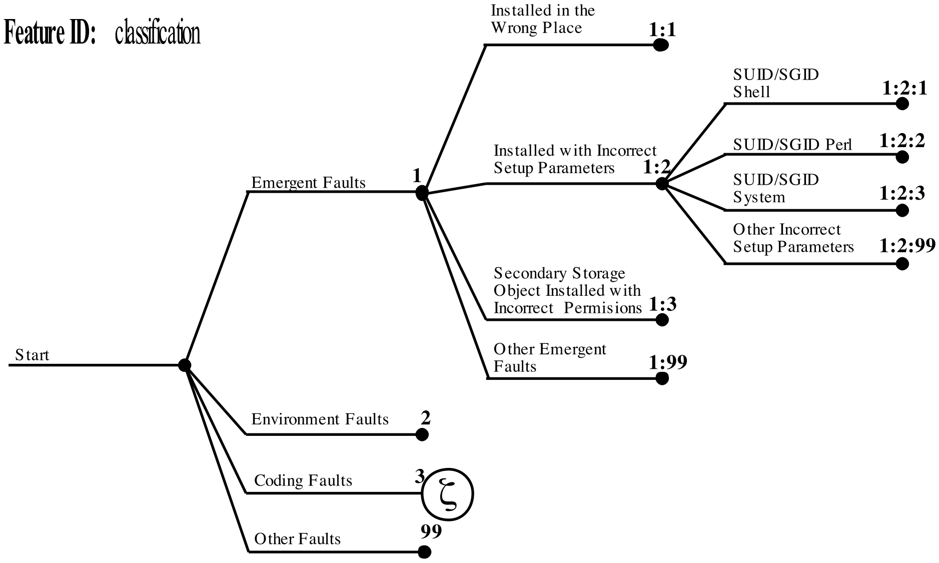

START
Can the vulnerability be exploited by a user typing a simple sequence of commands or instructions?
1
Yes
Simple
No
Can the vulnerability be exploited by a user executing a large number of commands or instructions (typically in a script or macro)?
2
Yes
Do the commands or script require timing and synchronization?
3
Yes
Timing
No
Complex
No
Does the exploitation of the vulnerability require a custom-made program or script?
4
Yes
Program
No
Does the exploitation of the vulnerability require timing and synchronization?
3
Yes
Timing
No
Other

xi a. Aslam classification decision tree, part 1 of 2. The Aslam classification has been expanded and a decision tree has been introduced to eliminate ambiguities end resolve some conflicts.

VULNERABILITY DATABASE
FEATURE SELECTION CRITERIA

Feature Name: Aslam Classification

Feature ID: classification

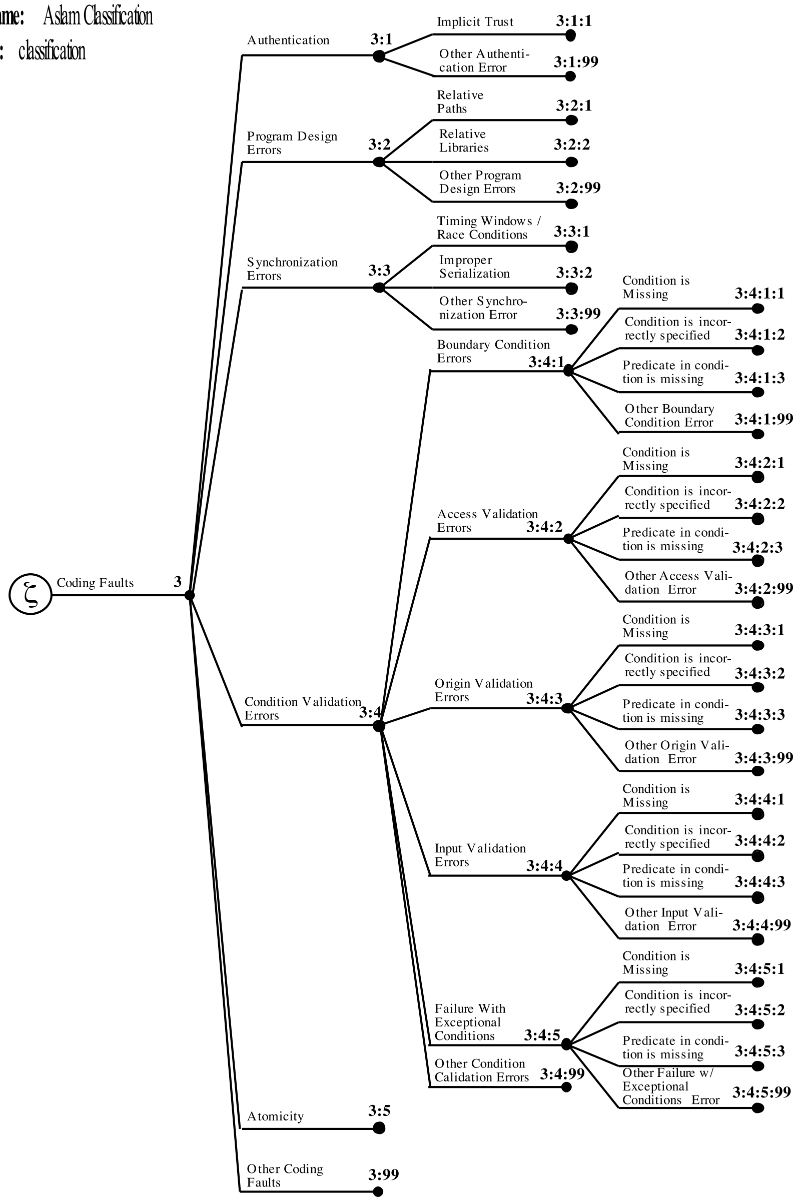


xi b. Aslam classification decision tree, part 2 of 2.

VULNERABILITY DATABASE FEATURE SELECTION CRITERIA

Feature Name: Aslam Classification

Feature ID: classification


ζ
Coding Faults
3
Authentication
3:1
Implicit Trust
3:1:1
Other Authentication Error
3:1:99
Program Design Errors
3:2
Relative Paths
3:2:1
Relative Libraries
3:2:2
Other Program Design Errors
3:2:99
Synchronization Errors
3:3
Timing Windows / Race Conditions
3:3:1
Improper Serialization
3:3:2
Other Synchronization Error
3:3:99
Condition Validation Errors
3:4
Boundary Condition Errors
3:4:1
Condition is Missing
3:4:1:1
Condition is incorrectly specified
3:4:1:2
Predicate in condition is missing
3:4:1:3
Other Boundary Condition Error
3:4:1:99
Access Validation Errors
3:4:2
Condition is Missing
3:4:2:1
Condition is incorrectly specified
3:4:2:2
Predicate in condition is missing
3:4:2:3
Other Access Validation Error
3:4:2:99
Origin Validation Errors
3:4:3
Condition is Missing
3:4:3:1
Condition is incorrectly specified
3:4:3:2
Predicate in condition is missing
3:4:3:3
Other Origin Validation Error
3:4:3:99
Input Validation Errors
3:4:4
Condition is Missing
3:4:4:1
Condition is incorrectly specified
3:4:4:2
Predicate in condition is missing
3:4:4:3
Other Input Validation Error
3:4:4:99
Failure With Exceptional Conditions
3:4:5
Condition is Missing
3:4:5:1
Condition is incorrectly specified
3:4:5:2
Predicate in condition is missing
3:4:5:3
Other Failure w/ Exceptional Conditions Error
3:4:5:99
Other Condition Calidation Errors
3:4:99
Atomicity
3:5
Other Coding Faults
3:99

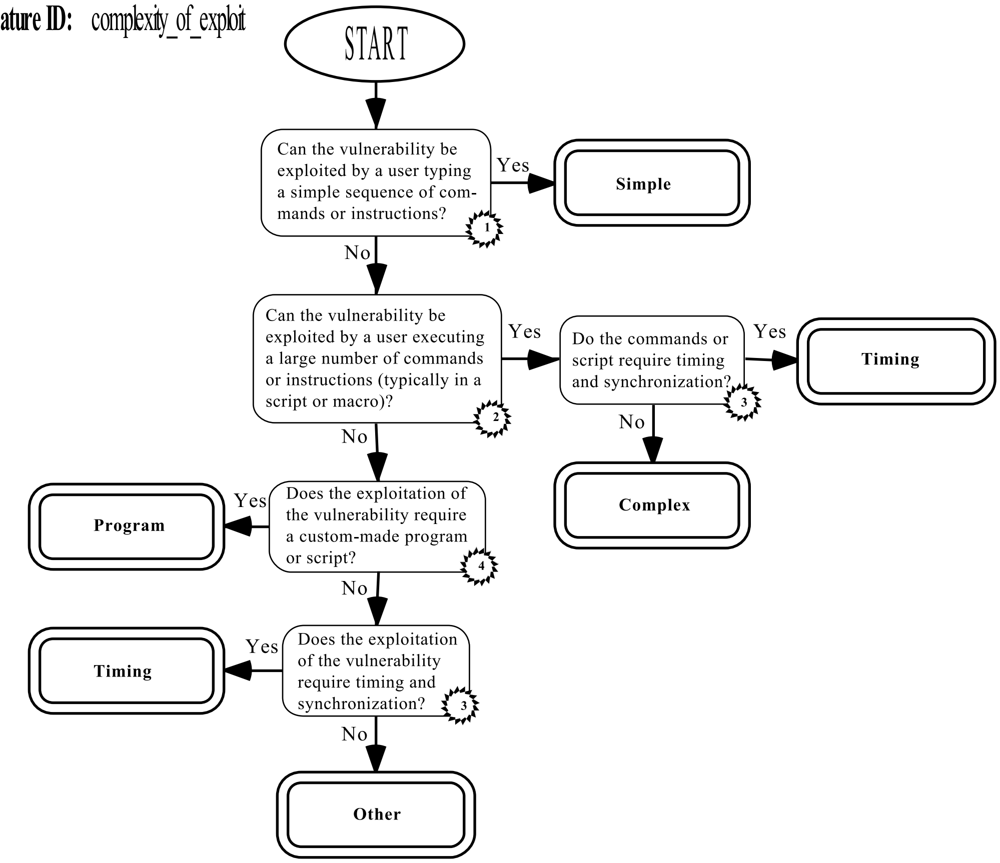
xii. Environmental Category (envass) This feature attempts to identify the environmental assumptions that were made by programmers.

Envass Possible Values	
Item Value	**Item Description**
nameinv	Assumes that a name (i.e. a path) is strongly bounded to a specific system object
objinv	Assumes the invariance of an object during the execution of program (i.e. the program assumes that no other subject can change the object while program is running)
objne	A program assumes that an object does not exist at the time of execution (i.e. a program assumes that name does not exist)
tempdel	A program assumes that a temporary item it created cannot be deleted by another subject while the program is running
memavail	A program that assume sufficient memory for its execution will always exist
netdata	A program assumes that data from a network service will always be reliable
envdata	A program assumes that the data in environment variables is valid and bounded
userdata	A program assumes that user provided input is valid and bounded
filedata	A program assumes that the input from a file is valid and bounded
reassembly	A program assumes that the reassembly of a data object form fragments will not affect the essential properties of the original object
execpath	A program assumes a specific execution path
objatt	A program assumes that certain attributes of certain objects have predefined values
perstore	A program assumes that persistent store is immutable (i.e. assumes that a file it writes cannot be modified by any other subject in between program runs)
dataexec	A program assumes that the modification of program data (by external subject) will not affect the semantics of the program
nameover	A program assumes that, while creating a file, any existing file that has the same name can be overwritten
falseconst	A program falsely assumes that a constraint or property holds in the system
insufverif	A program falsely assumes that a set of operations are sufficient for the verification of the property of an object
namepurpose	A program assumes that there is a strong binding between the name and purpose of an object
reservedobject	A program assumes that an object with a specific name will not be used by any other entity in the system by virtue of its name alone
Other	Other
NA	Does not apply
?	Unknown

xiii. System component category feature.
This feature attempts to identify the system component that contains the vulnerability.

Feature Name: Complexity of Exploit

Feature ID: complexity_of_exploit


START
Can the vulnerability be exploited by a user typing a simple sequence of commands or instructions?
1
Yes
Simple
No
Can the vulnerability be exploited by a user executing a large number of commands or instructions (typically in a script or macro)?
2
Yes
Do the commands or script require timing and synchronization?
3
Yes
Timing
No
Complex
No
Does the exploitation of the vulnerability require a custom-made program or script?
4
Yes
Program
No
Does the exploitation of the vulnerability require timing and synchronization?
3
Yes
Timing
No
Other

xiv. Nature_ object Possible Values

Nature_object Possible Values	
Item Value	**Item Description**
user_files	User files in the system
system_ file	System related or administrative files in the system
public_files	Publicly available files in the system
directory	Directories in the system
partition	A file system partition
heap_data	Data in the heap of running program
heap_code	Executable code in the heap of running program
stack_data	Data in the stack of a running program
static_data	Data that is statically allocated in a running program
stack_return	Return address of a function in the stack of a running program
stack_code	Executable code in the stack of a running program
password	Password or access token, can also be a pass-phrase
shell_command	Shell command
system_program	System program
user_program	User installed or owned program
system_info	Information regarding the system
outfiles	Files outside a restricted space Describes files that should not be environment, virtual environment, sandbox, etc
classloader	A ClassLoader object in Java or any object responsible for loading dynamic classes in any object oriented programming language
1ibrary	System function or service library
a_net_connection	Network connections to arbitrary hosts
web_pages	WWW page
names	User names, domain names, workgroup names, etc
pass_known	Well-known nonce encrypted with user password
o_attributes	System managed object attributes. Attributes the object itself (or entities other than the system) does not manage
cpu	CPU time
os	Operating System
email	Electronic Mail
netport	Network Port
packets	Network Packets
system_names	Internal system names in control of the system
device	A device in the system
addr_mapping	Address mapping maintained by the system i.e. an ARP cache
command_prompt	A command prompt presented to the user
other	Other
NA	Does not apply
?	Unknown

xv. Nature_effect Possible Values

Nature_effect Possible Values	
Item values	**Description**
replaced	Contents are completely replaced
changed	Can be written or can be changed
read	Can be read
append	Information can be appended
created	Can be created
displayed	Displayed or revealed
change_owner	Ownership can be changed
change_permission	Permissions can be changed
predictable	Is predictable or can be guessed
executed	Can be executed in isolation of expected policy
loaded	Can be dynamically loaded and linked
clear_text	Is transmitted or stored in clear text
exhausted	Is exhausted
crash	Crashes
bound	Can be bound to in violation of expected policy
exported	Can be exported for mounting
mounted	Is mounted or attached
locked	Can be locked
debugged	Can be debugged or attached to with a debugger
presented	Presented to the user in a console or Terminal
other	Other
NA	Does not apply
?	Unknown

xvi. Nature_method Possible Values

Nature_method Possible Values	
Item Value	**Item Description**
symlink	Program follows symbolic link or late binding link without verifying that the object being pointed to is correct
memcpy	Program uses strcpy, sprintf, or bcopy to copy data of arbitrary length to a stack buffer
config	Configuration error
back_ticks	Back ticks in parameter or input string
sepcial_chars	Special characters in input string, including file completion characters, special shell characters
dotdot	Uses ".." to climb up a directory tree past allowable bounds
verify_fail	Byte code or code verifier allows code that catches a security exception when creating an object loader
mod_name	Modifying compiled code to alter the name of objects
mod_env	Modifying environment variables
NTML_auth	NTML authentication process requires action
inherit_privs	Program inherits unnecessary privileges
capability	System provides inappropriate capability
hidden_mount	System provides hidden system mount point
syscall_disclose	System call discloses sensitive information
incorr_imp	Incorrect environment (mistaken environmental assumption)
rel_paths	Program refers to relative paths
incprot	Program fails to implement the protection mechanisms correctly
proxy	Program uses a trusted intermediary or proxy to bypass protection mechanisms
coresymlink	A program dumps a core file that follows symbolic links or late
infloop	Program uses an infinite and tight loop that consumes resources
criticalsect	Program fails to protect isolate a critical section
other	Other
NA	Does not apply
	Unknown

xvii. Nature_method_input Possible Values

Nature_method_input Possible Values	
Item Value	**Item Description**
env	Environment variable
command	User command line option
netdata	Network data
store	Persistent store
tempfile	Temporary file
conffile	Configuration file
datafile	Data file
gecos	System User information (Name, phone, number etc.)
parameter	Parameter to a system call
libparameter	Parameter to a library call
floppy	Removeable media
other	Other
NA	Does not apply
?	Unknown