Retired Draft

Warning Notice

The attached draft document has been RETIRED. NIST has discontinued additional development of this document, which is provided here in its entirety for historical purposes.

Retired Date March 10, 2022

Original Release Date September 30, 2016

Retired Document

Status Initial Public Draft (IPD)

Series/Number NISTIR 8138

Title Vulnerability Description Ontology (VDO): a Framework for Characterizing Vulnerabilities

Publication Date September 2016

Additional Information For current information on NIST's Vulntology Project, see https://github.com/usnistgov/vulntology.



Draft NISTIR 813	8
Vulnerability Description Ontology (VDO)	
A Framework for Characterizing Vulnerabilitie	S
Harold Boot Christopher Turne	



15	Draft NISTIR 8138
16	
17	Vulnerability Description Ontology
17	
18	(VDO)
19	A Framework for Characterizing Vulnerabilities
20	
21	Harold Booth
22	Computer Security Division
23	Information Technology Laboratory
24	
25	Christopher Turner
26	Booz Allen Hamilton
27	McLean, VA
28	
29	
30	
31	
32	
33	
34	
35	
36	September 2016
37	
38	
39 40	SUPPTINENT OF COMMITTINE SUPPTINENT OF COMMITTINE SUPPTINENT OF COMMITTINE SUPPTINENT OF COMMITTINE SUPPTINENT OF COMMITTINE SUPP
41	
42	U.S. Department of Commerce
43 44	Penny Pritzker, Secretary
45	National Institute of Standards and Technology
46	Willie May, Under Secretary of Commerce for Standards and Technology and Director

47	National Institute of Standards and Technology Internal Report 8138
48	45 pages (September 2016)
49	

50 Certain commercial entities, equipment, or materials may be identified in this document in order to describe an 51 experimental procedure or concept adequately. Such identification is not intended to imply recommendation or 52 endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best 53 available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to
 NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at
 http://csrc.nist.gov/publications.

63	Public comment period: September 30, 2016 through October 31, 2016
64	All comments are subject to release under the Freedom of Information Act (FOIA).
65	
66 67 68 69	National Institute of Standards and Technology Attn: Computer Security Division, Information Technology Laboratory 100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930 Email: nistir8138@nist.gov
70	
71	

Reports on Computer Systems Technology

73 The Information Technology Laboratory (ITL) at the National Institute of Standards and 74 Technology (NIST) promotes the U.S. economy and public welfare by providing technical 75 leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test 76 methods, reference data, proof of concept implementations, and technical analyses to advance the 77 development and productive use of information technology. ITL's responsibilities include the 78 development of management, administrative, technical, and physical standards and guidelines for 79 the cost-effective security and privacy of other than national security-related information in federal 80 information systems.

81

82

Abstract

83 This document aims to describe a more effective and efficient methodology for characterizing

84 vulnerabilities found in various forms of software and hardware implementations including but

85 not limited to information technology systems, industrial control systems or medical devices to

86 assist in the vulnerability management process. The primary goal of the described methodology

is to enable automated analysis using metrics such as the Common Vulnerability Scoring System
 (CVSS). Additional goals include establishing a baseline of the minimum information needed to

88 (CVSS). Additional goals include establishing a baseline of the infinitum information needed 89 properly inform the vulnerability management process, and facilitating the sharing of

properly inform the vulnerability management process, and facilitating the sharif

90 vulnerability information across language barriers.

- 91
- 92

Keywords

93 software defects; ontology; patching; taxonomy; vulnerabilities; vulnerability management

95

Acknowledgements

96 The authors, Harold Booth of the National Institute of Standards and Technology (NIST) and

97 Christopher Turner of Booz Allen Hamilton, wish to thank their colleagues who reviewed drafts

- 98 of this document and contributed to its technical content. The authors would like to acknowledge
- 99 Matthew Hansbury, Matthew Scola, and Steve Christey of the MITRE Corporation for their
- 100 insightful assistance in the development of this document.
- 101

Audience

102 This document is intended for anyone who participates in the vulnerability management process.

103 Possible stakeholders include security response teams of manufacturers who need to respond to

104 vulnerabilities discovered in their products, security researchers who wish to share vulnerability

105 information with manufacturers or other vulnerability coordination entities, system

administrators and/or owners who need to identify vulnerabilities in their systems and prioritize

107 their remediation, vulnerability discovery tool vendors, and vulnerability databases.

108

Note to Reviewers

109 This is the first draft of several anticipated drafts of a document intended to describe a

110 methodology for characterizing vulnerabilities. It is not intended to be complete at this time and

the authors do not expect that this draft reflects the full breadth and depth of the information

112 needed to fully automate the descriptions for vulnerabilities. Reviewers are asked to provide 113 feedback on terminology that is unclear, in conflict with established practice and are encouraged

feedback on terminology that is unclear, in conflict with established practice and are encouraged to provide feedback and examples where the current draft falls short in enabling the description

of a vulnerability. To the extent that is reasonable and in keeping with the purpose of this

- document (Section 1.1), future drafts will be produced attempting to incorporate this feedback
- 117 with the goal of improving the final version.

118 Questions and items of particular note have been highlighted to encourage feedback.

119

Trademark Information

120 CVE is a registered trademark of The MITRE Corporation.

121 All other registered trademarks or trademarks belong to their respective organizations.

123	Document Conventions
124	The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
125	"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
126	report are to be interpreted as described in Request for Comment (RFC) 2119 [RFC2119]. When
127	these words appear in regular case, such as "should" or "may", they are not intended to be
128	interpreted as RFC 2119 key words.

129			Table of Contents	
130	1	Intro	duction	.1
131		1.1	Purpose	2
132		1.2	Methodology	2
133	2	Over	view	3
134		2.1	Noun Group Value Types	3
135	3	Nour	n Groups	5
136		3.1	Vulnerability	5
137		3.2	Sector of Interest	6
138		3.3	Known Chain	6
139		3.4	Provenance	7
140		3.5	Scenario	7
141		3.6	Туре	8
142		3.7	Product	8
143		3.8	Attack Theater	9
144			3.8.1 Remote Type	0
145			3.8.2 Limited Remote Type 1	0
146		3.9	Barrier 1	11
147			3.9.1 Engineering Method1	12
148			3.9.2 Victim Type1	12
149			3.9.3 Race Condition Type 1	13
150			3.9.4 Privilege Information	13
151			3.9.5 Privilege Level 1	14
152		3.10	Context 1	15
153		3.11	Application Type 1	16
154		3.12	Entity Role 1	16
155		3.13	Mitigation 1	17
156		3.14	Impact Method 1	18
157			3.14.1 Trust Failure Type1	9
158		3.15	Logical Impact 2	20
159			3.15.1 Service Interrupt Type	21
160			3.15.2 Location	22
161		3.16	Physical Impact	22

162		3.16.1 Physical Consumption Type	23
163		3.16.2 Human Injury Level	23
164		3.17 Scope	24
165		3.18 Criticality	24
166	4	Conclusion	25
167 168		List of Appendices	
169	Ар	pendix A— Example Usage	26
170	Ар	pendix B— Conversion to descriptive text (English)	
171	Ар	pendix C— Mapping VDO representations to CVSS Scores	33
172	Ар	pendix D— Acronyms	35
173	Ар	pendix E— References	
174			
175		List of Tables	
176	Ta	ble 1 Valid Value Types	3
177			

178 **1** Introduction

179 When two or more groups share information, a common vocabulary is critical for success. The cybersecurity landscape is relatively new and therefore is still in its infancy in developing these 180 181 shared vocabularies. The ontology described in this document is a fundamental building block in 182 developing that shared understanding for vulnerabilities among cybersecurity professionals. For 183 the purposes of this document a vulnerability is defined as any weakness in the computational 184 logic found in products or devices that could be exploited by a threat source [NISTIR 7298]. 185 186 Managing these vulnerabilities within an organization is described as the vulnerability 187 management process. The vulnerability management process consists of identifying whether an 188 organization has endpoints containing the vulnerability, determining the exposure of the 189 vulnerability within the organization and evaluating the impact of successful exploitation of a 190 vulnerability within the context of the organization. An organization must determine whether the 191 exposure and impact of a specific vulnerability warrants a response and prioritize that response 192 among other critical activities. Organizations then need to make a similar decision for each 193 vulnerability. The analysis needed to inform the prioritization is currently a time-consuming, 194 manual process and is often based on reading security bulletins and vendor advisories which 195 sometimes provide incomplete or conflicting information. 196 197 This document defines a framework that improves upon this manual process by enabling a 198 mechanism to describe vulnerabilities in a machine consumable format. While this document 199 does not describe a particular format to encode the vulnerability data, it is expected other efforts 200 will use this document as a foundation for the creation of a machine processible format. The 201 format will enable automated tools to assist in the analysis process. In addition, consumers of 202 vulnerability information will be able use the vocabulary described in this framework to identify 203 missing information and encourage more complete and accurate vulnerability descriptions from 204 their providers. More complete and accurate descriptions will better facilitate the vulnerability 205 management process for organizations. 206 207 In addition to those responsible for an organization's vulnerability management function, other 208 stakeholders include: 209 • Security Researchers – who need to share and disclose vulnerability information to 210 vendors 211 • Software Publishers – who need to share and disclose vulnerability information to their 212 customers 213 • Vulnerability Coordinators – who need to share and disclose vulnerability information to 214 software publishers and to users of the affected software 215 Vulnerability Information Services – that need to provide vulnerability information to the • 216 consumers of their data, often performing additional analysis which can assist in the 217 prioritization of vulnerabilities for organizations 218 219 All of these stakeholders need a common language to describe and characterize vulnerabilities as 220 well as a way to express what information is needed to perform their activities. The framework 221 in this document intends to provide this common language and to provide a way for stakeholders

to describe required information.

224 **1.1 Purpose**

225 The purpose of this document is to create a more effective and efficient methodology within the

226 vulnerability management sphere that describes vulnerabilities in a universal manner.

Additionally, it enables automated scoring, improves the amount of detail that can be provided

- 228 about a vulnerability while minimizing the risk of the information being used to exploit the
- vulnerability, and allows for better sharing of vulnerability information across language barriers.
- 230

231 **1.2 Methodology**

Any recommended concept or idea from stakeholders that align with the purpose stated in

233 Section 1.1 will be considered. Specifically the framework is focused on vulnerability

234 management and automating that process, and thus any additions or modifications will be made

to improve that use case.

236

237 This document is not intended to provide guidance on a particular implementation of syntax or

serialization, but to provide a framework that specifies available characteristics, valid values, and

- relationships. If multiple serialization mechanisms are developed that adhere to this framework
- they would hopefully be semantically interoperable.

241 2 Overview

247

248

- 242 The framework is composed of:
- **noun group** a conceptual entity containing related noun group values;
- **noun group definition** description of a noun group; what it is and how it is used;
- usage each noun group is identified as:
 0 mandatory (M) indicates a value
 - \circ mandatory (M) indicates a value for the noun group SHALL be provided,
 - \circ recommended (R) indicates a value for the noun group SHOUD be provided,
 - \circ optional (O) indicates a value for the noun group MAY be provided;
- noun group values valid values are either chosen from an enumerated list of values
 specific to each noun group or have an expected format. The format is composed of types
 which are described in the ABNF notation of Section 2.1 with the type name represented
 in italics as follows: <typename>;
- noun group value definition description of a noun group value; what it is and how it is used; and
- relationships noun groups are related to each other through the allowed relationships as described for the noun group. The cardinality of the relationship indicates whether multiple values are permitted for the noun group. Noun group values may also have a relationship to another noun group. Relationships will be represented in the following format [<cardinality> <target noun group> " value/s " <usage> " be associated with " (<origin noun group>/<origin noun group value>)].
- 261 When noun group names are referenced throughout this document they will appear in italics.

262 **2.1 Noun Group Value Types**

The following section describes the available types used to describe the expected format for noun groups that have valid values that are not an enumerated list. The following uses Augmented Backus–Naur Form (ABNF) as described in [RFC5234]. The formats for the valid values are intended to describe the expected contents of the value and are not representative of any particular syntax or serialization mechanism.

268	Table 1 Valid Value Types	
	source	= string
	vulnerability-identifier	= namespace identifier
	vulnerability-type	= namespace identifier
	product-configuration	= 1*product-identifier / (namespace string)
	product-identifier	= namespace identifier
	namespace	= string

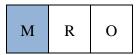
identifier	= string
string	= 1*VCHAR
Number	= 1*DIGIT

270 **3** Noun Groups

271 Noun groups are the core building block of the framework.

272 **3.1 Vulnerability**

A *Vulnerability* is any weakness in the computational logic found in products or devices that could be exploited by a threat source.



<vulnerability-< th=""><th>An identifier for a vulnerability supplied by a source.</th></vulnerability-<>	An identifier for a vulnerability supplied by a source.
•	• • • • •
identifier>	Examples include a knowledge base article number, patch number, a
	bug tracking datatbase identifier or a common identifier such as a
Example:	Common Vulnerabilities and Exposures (CVE) identifier. CVE is a
cve.mitre.org	widely adopted identifier used across many organizations.
CVE-2015-1234	

Relationships: Scenario, Sector of Interest, Known Chain, Provenance

- One or many Scenario values shall be associated with Vulnerability.
- Zero or many Sector of Interest values may be associated with Vulnerability.
- Zero or many Known Chain values may be associated with Vulnerability.
- Zero or many Provenance values may be associated with Vulnerability.

275 **3.2 Sector of Interest**

Supplemental information identifying potential sectors or use cases where the *Vulnerability* could have an impact.

М	R	0
---	---	---

Industrial Control Systems ¹	The <i>Vulnerability</i> affects software that interfaces with manufacturing or production control systems.
Health Care	The <i>Vulnerability</i> is found within information systems that are related to health care. This includes both software whose purpose is to provide services specifically for health care, as well as medical devices.
Financial	The <i>Vulnerability</i> is found within software that relates to financial operations or activities.
Relationships: Vuln	erability

Relationships: <u>Vulnerability</u>

• Zero or many Sector of Interest values may be associated with Vulnerability.

276

3.3 Known Chain

An identifier for another known *Vulnerability* that can be used in conjunction with the *Vulnerability* in question to achieve a different and likely greater impact.

М	R	0
---	---	---

<vulnerability-< th=""><th>A central identifier for each vulnerability supplied by a source.</th></vulnerability-<>	A central identifier for each vulnerability supplied by a source.
identifier>	Examples include a knowledge base article number, patch number, a
-	bug tracking database identifier or a common identifier such as a CVE
Example:	identifier.
cve.mitre.org	
CVE-2015-1234	
Relationships: Vul	nerability

Relationships: Vulnerability

• Zero or many Known Chain values may be associated with Vulnerability.

¹ The term 'industrial control system' is defined in NIST IR 7298 R2: http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

Μ

R

0

280 **3.4 Provenance**

Representation of the source of the information for the related item.



<source> The name of the source which provided the information related to the *Vulnerability*.

Relationships: <u>Vulnerability</u>

• Zero or many Provenance values may be associated with Vulnerability.

281

282 **3.5 Scenario**

A scenario is the placeholder to allow a description of the conditions surrounding the possible use of a vulnerability. *Vulnerability* must have a least one Scenario, with multiple possible *Scenarios* being common. A single *Vulnerability* can likely be exploited by many different approaches with possible varying impacts. For example a remote exploit could rely on user interaction to be downloaded, or a local attack could use the same vulnerability to obtain the same or similar impact.

<number> A simple numerical identifier identifying this *Scenario* within the *Vulnerability*.

Relationships: Vulnerability, Barriers, Context, Attack Theater, Product, Type

- One or many Scenario values shall be associated with Vulnerability.
- Zero or many Barrier values should be associated with Scenario.
- One or many Context values shall be associated with Scenario.
- One and only one Attack Theater shall be associated with Scenario.
- Zero or many Product values may be associated with Scenario.
- Zero or many Type values may be associated with Scenario.

286 **3.6 Туре**

The type, category, or weakness of the *Vulnerability*. When choosing a value, the most applicable types should be selected based on the type system used.



<vulnerability-type></vulnerability-type>	An identifier of the vulnerability category, type or weakness. Examples
	of type systems include the Open Web Application Security Project
Example:	(OWASP) Vulnerability Categories [OWASP-VULN] and the
cwe.mitre.org	Common Weakness Enumeration (CWE) [CWE] which provide
CWE-123	descriptions and names for various types of vulnerabilities.
Relationships: Scenario	

Relationships: <u>Scenario</u>

• Zero or many Type values may be associated with Scenario.

287 288

3.7 Product

The software and/or hardware configurations that are known to be vulnerable to exploitation of the *Vulnerability*. Different *Product* configurations can be associated with different *Scenarios* to allow for description of varying impacts and explotation mechanisms.



<product-< td=""><td>A list of identifiers or an applicability language which allows for the</td></product-<>	A list of identifiers or an applicability language which allows for the
configuration>	description of the product configuration. Example product identifiers
	are Software Identifiers (SWID) as described in [ISO/IEC 19770-
Example:	2:2015] and Common Platform Enumeration (CPE) names as
http://standards.iso.o	described in [CPEN]. An example of an applicability language would
rg/iso/19770/-2/2015	be the CPE Applicability Language described in [CPEAL].
2001-	
06.com.acme_ACM	
E_Application-1.01	

Relationships: Scenario

289

[•] Zero or many Product values may be associated with Scenario.

291 **3.8 Attack Theater**

Attack Theater is the area or place from which an attack must occur. Each separate theater represents varying levels of implied trust and attack surface.

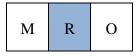
M R O

Remote	The exploit scenario requires that the attack occurs over the network stack; normally external to the target's internal network such as from the Internet. Common targets in the remote theater are public websites, Domain Name System (DNS) services, or web-browsers.
	Noun-specific relationship: <u>Remote Type</u>
	• One and only one Remote Type value should be associated with <i>Remote.</i>
Limited Remote	 The exploit scenario requires that the attack can occur over layer 2 or layer 3 technologies, but a limitation exists either by the nature of the network communication or by range constraints. Examples of range constraints are <i>Cellular</i>, <i>Wireless</i>, <i>Bluetooth</i>, <i>Infrared</i>, or <i>Line-Of-Sight</i>. <i>Noun-specific relationship: Limited Remote Type</i> One and only one Limited Remote Type value should be associated with Limited Remote.
Local	The exploit scenario requires that the attack can only occur after the adversary has logical local access to a device such as through a console, Remote Desktop Protocol (RDP), Secure Shell (SSH), or Telnet login.
Physical	The exploit scenario requires the attacker's physical presence at the target.
Relationships: Sce	
-	ly one Attack Theater value shall be associated with Scenario

• One and only one Attack Theater value shall be associated with Scenario.

294 3.8.1 Remote Type

Remote Type futher refines the *Remote* selection of the *Attack Theater* noun group to provide additional detail on where an adversary must be located. Selection of a *Remote Type* value will assist in determing the types of threats that can take advantage of the vulnerability.



The attack must be launched from within an organizations internal
network that is shielded from direct access of the Internet. (Ex: A router is configured by default to only allow connections from the Intranet ports and not the WAN ports.) This also represents broadcast domains.
An attacker must have access to a physical interface to the network, or collision domain.
dor An

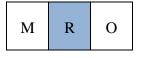
Relationships: <u>Remote</u>

• One and only one Remote Type value should be associated with Remote.

295 296

3.8.2 Limited Remote Type

Limited Remote Type futher refines the *Limited Remote* selection of the *Attack Theater* noun group to provide additional detail on where an adversary must be located. Selection of a *Limited Remote Type* value will assist in determing the types of threats that can take advantage of the vulnerability.



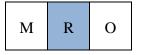
The attack must be launched from a cellular network.
The attack must be launched from a wireless (802.11x) network.
The attack must be launched relying on a Bluetooth communication
channel.
The attack must be launched relying on an Infrared communication
channel.
The attack must be launched using a Line-of-Sight system such as
ocular.
-

Relationships: <u>Limited Remote</u>

• One and only one Limited Remote Type value should be associated with Limited Remote.

299 **3.9 Barrier**

Any characteristic inherent in the vulnerability that could impede the adversary from achieving successful exploitation. A barrier increases the difficulty an attacker faces when attempting to execute an exploit for the vulnerability.



Social Engineering	 The exploit scenario requires that an attacker perform some type of social engineering to achieve a successful exploit attempt. Typically, an attacker convinces a victim into interacting with a malicious resource. <i>Noun-specific relationship: Engineering Method, Victim Type</i> One or many Engineering Method values should be associated with Social Engineering. Zero or one Victim Type values should be associated with Social Engineering.
Race Condition	 The exploit scenario includes requiring an attacker to take advantage of a race condition. <i>Noun-specific relationship:</i> <u>Race Condition Type</u> One and only one Race Condition Type value should be associated with Race Condition.
Specialized Condition	The exploit scenario requires specific, non-default configuration settings within the vulnerable software. For example the use of a non- standard port for a networked service like ssh.
Environmental Condition	The exploit scenario requires an environmental condition external to the vulnerable software that is not necessarily related to the vulnerable software itself. A congested network would be an example of an environmental condition.
Precondition Required	Information about the target is necessary in order to exploit the vulnerability on a specific target. For example the hostname of the device may necessary in order to exploit the vulnerability on that device.
Privilege Required	 The exploit scenario requires an attacker to have certain privileges prior to successful exploitation attempts. Noun-specific relationship: <u>Privilege Information</u> Zero or one Privilege Information values should be associated with Privilege Required. Noun-specific relationship: <u>Privilege Level</u> One and only one Privilege Level value should be associated with Privilege Required. Noun-specific relationship: <u>Context</u> One and only one Context value should be associated with Privilege Required.

Relationships: Scenario

• Zero or many Barrier values should be associated with Scenario.

300 301

3.9.1 Engineering Method

The method or mechanism used to manipulate a user into interacting with a malicious resource.



Malicious Link	A URL or hyperlink that has been crafted in a way that causes a target program or website to operate in an unintended fashion
Malicious File	A file that has been crafted in a way that causes a target program to operate in an unintended fashion
Malicious Website Content	A website that has been crafted in a way that causes a target program to operate in an unintended fashion or is used to simulate a site that the target user trusts.
Malicious Application	An application that has been modified or crafted to perform operations that are unintended
Relationships: Social	Engineering

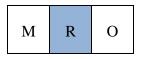
• One or many Engineering Method values should be associated with Social Engineering.

302

303 304

3.9.2 Victim Type

When a user is targeted through the use of *Social Engineering* the *Victim Type* is used to describe the possible *Privilege Level* values along with the *Context* of those privileges. The level of privilege the target has should be reflected in the *Logical Impact* and *Physical Impact* values selected.



<number></number>	A simple numerical identifier to identify this instance of a victim for
	the Scenario.

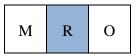
Relationships: <u>Social Engineering</u>, <u>Context</u>, <u>Privilege Level</u>

- Zero or one Victim Type instances should be associated with Social Engineering.
- One and only one Context value should be associated with Victim Type.
- One and only one Privilege Level value should be associated with Victim Type.

305

306 **3.9.3 Race Condition Type**

Race Condition Type further refines the *Race Condition* selection of the *Barrier* noun group to provide additional detail on the level of likely control an adversary has to trigger the vulnerable race condition. Note that this is only a description of how much control an attacker has over the inputs involved in the race condition and not an indication of the reproducibility of triggering the race condition itself.



No Control	An attacker has no control over how the race condition will be
	triggered. The attacker must be fortunate to encounter the race
	condition.
Partial Control	An attacker is able to start one or more of the inputs which take part in
	the race condition but does not have control over all inputs. For
	example a vulnerability exists in the processing of a particular type of
	input on the intial start-up of a device and an attacker must supply that
	input during the period when the device is starting up and the attacker
	has no control over when the device starts up.
Full Control	An attacker is able to routinely start all inputs which will trigger the
	race condition.

Relationships: <u>Race Condition</u>

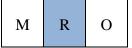
• One and only one Race Condition Type value should be associated with Race Condition.

307

308 309

3.9.4 Privilege Information

Extra information regarding the *Privilege Required* barrier. This includes factors about privileges required before an attack is launched that can alter the attack's complexity.



Multiple	Exploiting the vulnerability requires that the attacker authenticate two
Authentication	or more times, even if the same credentials are used each time. An
	example is an attacker authenticating to an operating system in
	addition to providing credentials to access an application hosted on
	that system.

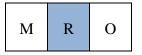
Relationships: Privilege Required

• Zero or one Privilege Information values should be associated with Privilege Required.

310 311

312 3.9.5 Privilege Level

Abstraction to assist in capturing relative privilege levels. The abstraction is only for the sake of discussing the vulnerability and is not intended to communicate the actual granular privileges that exist in most information system environments.



Anonymous	No privileges required. NOTE: Is this a needed value? Should the lack
-	of an associated Privilege Level infer this? Or does the absence of a
	Privilege Level indicate a lack of knowledge?
User	Representative of a generic or basic user.
Privileged	Representative of something more than a base user, but not the full
	control of an Administrator
Administrator	Representative of when the privilege allows complete or nearly
	complete access to the context. Common terms include Admin,
	Administrator, Root, System or Kernel.
Generic Trust	This level is for applications or software packages that allow public
	account creation. Meaning that anyone who has access to the software
	has the abilility to create an account and access basic functionality.
Relationships: Pr	ivilege Required, Privilege Escalation

• One and only one Privilege Level value should be associated with Privilege Required.

• One and only one Privilege level value should be associated with Privilege Escalation.

315 3.10 Context

The entity where the impacts are realized from successful exploitation of a security vulnerability. Different impacts can be realized by multiple contexts from multiple scenarios.

М	R	0
---	---	---

Hypervisor	A program or operating system that coordinates the sharing of
J 1	hardware resources for multiple operating systems. Each guest
	operating system appears to have its own processor, memory, and
	other resources to itself. However, the hypervisor is controlling the
	shared hardware resources, allocating what is needed to each operating
	system as necessary, and isolating the guest operating systems from
	each other.
Firmware	Stored software that is considered to be built-in to a device. This is
	most commonly seen within embedded devices, routers, firewalls,
	BIOS and UEFI.
Host OS	An operating system running as the foundation layer for other software
	applications. This is intended to be used when the Hypervisor context
	is not applicable, otherwise Guest OS should be used.
Guest OS	An operating system running as the foundation layer for other software
	applications. This is intended to be used when the Hypervisor context
	is applicable, otherwise Host OS should be used.
Application	A program designed and implemented to accomplish a specific task.
	Applications can run on or within operating systems, firmware or other
	applications.
	Noun-specific relationship: Application Type
	• Zero or more Application Type values should be associated
	with Application.
Channel	The logical communication medium that is being used between other
	contexts. Channel is intended to be used when a protocol or cipher
	suite has a flaw inherently as opposed to an implementation issue.
	Examples would be failures of sufficient entropy in the cipher text or
	cryptographic key strength.
Physical Hardware	The actual physical hardware such as the logic gates within processors,
	the sectors of a disk or cells within memory.
Relationships: <u>Entity</u>	y Role, Impact Method, Mitigation, Privilege Required, Victim Type
•	Entity Role values should be associated with Context.
•	Impact Method values shall be associated with Context.
•	Mitigation values may be associated with Context.
One and only	one Context value should be associated with Privilege Required.
 One and only 	one Context value should be associated with Victim Type

• One and only one Context value should be associated with Victim Type.

318 **3.11 Application Type**

Application Type further refines the *Application* noun group value to provide additional detail on the category or type of application.

M R O

NOTE: The noun group values are not exhaustive and are intended to be
illustrative of the types of values. Feedback on whether this is needed or
desired is requested.

Web Server	An application which provides general web server functions.
Database	An application which provide database functions.

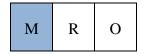
Relationships: Application

• Zero or many Application Type values should be associated with Application.

319

320 3.12 Entity Role

Describes the role an associated *Context* performs in the vulnerability scenario being described.



Vulnerable	Associated Context contains the Vulnerability
Primary Authorization	Associated <i>Context</i> is the main or initial authorization scope of the vulnerability scenario. See section 2.2 in [CVSSV3] for a full
Authorization	description of authorization scope.
Secondary	Associated Context is the secondary authorization scope of the
Authorization	vulnerability scenario. See section 2.2 in [CVSSV3] for a full
	description of authorization scope.

Relationships: Context

• One or many Entity Role values shall be associated with Context.

322 **3.13 Mitigation**

Describes protection mechanisms that may limit the impact or actions that can be taken even if the vulnerability is able to be exploited. These mechanisms are often part of the system in which the product is deployed or are inherent in how the product is used.



NOTE: This noun group is intended to capture situations where a vulnerability exists but the manner in which the product is used mitigates the vulnerability. Is this useful? Are the noun group values the right type of		
thing to capture?		
ASLR	Some form of Address space layout randomization (ASLR) is in use.	
Multi-Factor	Some form of Multi-Factor Authentication is required to access the	
Authentication	product.	
Sandboxed	The product is deployed within a sandbox.	
HPKP/HSTS	HTTP Public Key Pinning (HPKP) or HTTP Strict Transport Security	
	(HSTS) is in use.	
Physical Security	Some form of physical security is in place that would mitigate this	
	vulnerability.	

Relationships: Context

• Zero or many Mitigation values may be associated with Context.

325 3.14 Impact Method

A description of the method used to exploit a vulnerability providing some additional information on the impact of exploitation.

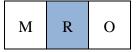
М	R	Ο
---	---	---

NOTE: Are there additional Noun Group values?

Context Escape	The Vulnerability allows an adversary to exploit a trust mechanism by
	breaking out of a sandbox and into another workspace. This Impact
	Method noun group value is intended to be associated with the Context
	that has been escaped.
	Noun-specific relationship: <u>Context</u>
	• One and only one Context value shall be associated with
	Context Escape. The association denotes where a sandbox
	breakout originated.
Trust Failure	Exploitation of the <i>Vulnerability</i> takes advantage of an assumed trust
	relationship leading to unexpected impacts. Examples include failures
	of inherent trust, failure to verify a communicator, or the content being
	transmitted.
	Noun-specific Relationship: <u>Trust Failure Type</u>
	• One or many Trust Failure Type values should be associated
	with Trust Failure.
Authentication	Exploitation of the Vulnerability takes advantage of a failure to
Bypass	identify the adversary properly, directly leading to additional access or
J 1	permissions.
Man-in-the-Middle	The exploit scenario requires that an adversary perform a Man-in-the-
	Middle (MitM) attack. MitM attacks involve an adversary positioning
	themselves inside a communication channel between two or more
	parties. This is usually accomplished by exploiting a trust mechanism
	and tricking both ends of the communication channel into believing
	that they are communicating with the intended party. Once
	successfully injected into a communication channel, the MitM is
	capable of sensitive data disclosure, modification of data being
	transmitted, transmission of false data to either party (impersonation)
	or denial of communication to either party.
Code Execution	Exploitation of the Vulnerability allows an adversary to execute
	unauthorized code, causing an impact to a <i>Context</i> .
Relationships: <u>Conte</u>	ext, <u>Logical Impact, Physical Impact</u>
• One or many	Impact Methods shall be associated with Context
• One or many	Logical Impacts shall be associated with Impact Method
•	y Physical Impacts should be associated with Impact Method

328 3.14.1 Trust Failure Type

A refinement to describe the type of failure in the associated <i>Context</i>
which exposed the vulnerability.



Failure to verify	The <i>Context</i> failed to ensure the entity on the receiving end of the	
receiver	communication is the intended entity.	
Failure to verify	The Context failed to ensure the entity on the transmitting end of the	
transmitter	communication is the intended entity.	
Failure to verify	The Context failed to ensure the content supplied is properly formatted	
content	and sanitized.	
Failure to establish	The <i>Context</i> failed to verify the input originated from a trusted source,	
trust	in other words a check is missing or non-existent.	
Relationships: Trust	Egilura	

Relationships: <u>Trust Failure</u>

• One or many Trust Failure Type of Trust values should be associated with Trust Failure.

330 **3.15 Logical Impact**

A description of the possible impacts to the *Context* that a successful exploitation of the *Vulnerability* can have. The same *Vulnerability* can have multiple and different *Logical Impact* noun group values across different *Context* or *Scenario* instances.



Write (Direct)	The Vulnerability allows an adversary to cause a breach in the integrity	
	of the Context through unauthorized modification or addition of data.	
Read (Direct)	The Vulnerability allows an adversary to cause a breach of	
	confidentiality by gaining unauthorized access to data in the Context.	
Resource Removal	The Vulnerability allows an adversary to perform an unauthorized	
(Data)	removal (deletion) of data from a resource in the Context.	
Service Interrupt	The Vulnerability allows an adversary to cause an unauthorized loss of	
	availability by temporarily or permanently disabling all or a portion of	
	the Context.	
	Noun-specific relationship: <u>Service Interrupt Type</u>	
	• One or many Service Interrupt Type values should be	
	associated with Service Interrupt.	
Indirect Disclosure	The Vulnerability allows an adversary to learn information about the	
	<i>Context</i> , but the knowledge gained is not from a direct read operation.	
	Examples include but are not limited to discovering memory locations	
	protected by ASLR, information from side-channel attacks, or	
	information gained from traffic analysis.	
Privilege Escalation	The Vulnerability allows an adversary to gain a level of privilege that	
	was not intended. Unlike the other Logical Impact noun group values,	
	Privilege Escalation is intended to represent that anything the	
	Privilege Level acquired can do, can be done by the adversary. If an	
	adversary is able to only accomplish a subset of the other Logical	
	Impact noun group values, that subset MUST be associated to the	
	Context as well. Otherwise, all other Logical Impact noun group values	
	are assumed.	
	Noun-specific relationship: Privilege Level	
	• One and only one Privilege level value should be associated	
	with Privilege Escalation.	
Relationships: Impac	<u>et Method</u> , <u>Location</u> , <u>Scope</u>	
0		

- One or many Logical Impact values shall be associated with Impact Method
- Zero or many Location values may be associated with Logical Impact
- One and only one Scope value shall be associated with Logical Impact

332 **3.15.1 Service Interrupt Type**

Additional information to describe the nature and type of service interruption possible through the exploitation of a *Vulnerability*. Both *Service Interrupt Type* and *Scope* noun group values should be applied where applicable.



Churcherry	The convice intermetion results in the Constant shutting down	
Shutdown	The service interruption results in the <i>Context</i> shutting down	
Reboot	The service interruption results in the <i>Context</i> powering off, but	
	starting back up immediately.	
Hang	The service interruption results in the <i>Context</i> being stuck at a certain	
	point and unable to continue function	
Panic	The service interruption results in the <i>Context</i> crashing	
Unrecoverable	The service interruption results in a complete and unrecoverable loss	
	of the <i>Context</i> but is non-physical in nature. For example the	
	corruption of the firmware on a hardware device with no possibility of	
	reload.	

Relationships: <u>Service Interrupt</u>

• One or many Service Interrupt Type values should be associated with Service Interrupt.

334 3.15.2 Location

A refinement to the Logical Impact noun group values designating the specific area or location impacted. Serves as supplemental information for the overall *Vulnerability* description.

Memory	The Logical Impact is able to occur within memory
File System	The Logical Impact is able to occur within the file system
Network Traffic	The Logical Impact is able to occur within network traffic
Relationships: Logic	al Impact

• Zero or many Location values may be associated with Logical Impact

335

336 **3.16 Physical Impact**

Used when exploitation of the *Vulnerability* could result in a tangible impact to the physical device or machinery controlled by or through the *Context*, or the surrounding environment, which could be other nearby devices, machinery or people.

Physical Resource	An exploit of the <i>Vulnerability</i> could cause excessive physical resource	
Consumption	consumption resulting in a tangible cost.	
	Noun-specific relationship: <u>Physical Consumption Type</u>	
	• One or many Physical Consumption Type values must be	
	associated with Physical Resource Consumption.	
Property Damage	An exploit of the Vulnerability could result in physical damage to the	
	device or surrounding environment.	
Human Injury	An exploit of the Vulnerability could result in injury to users or nearby	
	individuals.	
Noun-specific relationship: <u>Human Injury Level</u>		
	• One and only one Human Injury Level value should be	
	associated with Human Injury.	
Relationships: Impo	act Method Scope	

Relationships: <u>Impact Method</u>, <u>Scope</u>

- One and only one Scope value shall be associated with Physical Impact
- Zero or many Physical Impact values should be associated with Impact Method

337





R

0

Μ

338 **3.16.1 Physical Consumption Type**

NISTIR 8138 (DRAFT)

The *Vulnerability* allows for consumption of resources outside the digital realm. This consumption could lead to wear and tear on the hardware or financial implications from usage.

Electricity	Exploitation of the Vulnerability enables excessive electricity usage
Water	Exploitation of the Vulnerability enables excessive water usage
Assets	Exploitation of the Vulnerability enables excessive use of an asset.
	The excessive use could decrease the usable lifetime of the asset or
	unnecessarily consume fuel.
Delationshing	Physical Persures Consumption

Relationships: <u>Physical Resource Consumption</u>

• One or many Physical Consumption Type values should be associated with Physical Resource Consumption.

339

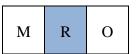
340 **3.16.2 Human Injury Level**

A description of the possible impacts to any human as a result of exploitation of the *Vulnerability*. Descriptions below are based on Table D.3 in [ISO/IEC 14971:2007].

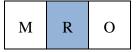
Negligible	Inconvenience or temporary discomfort	
Minor	Temporary injury or impairment not requiring professional medical	
	intervention	
Serious	Injury or impairment requiring professional medical intervention	
Critical	Permanent impairment or life-threatening injury	
Catastrophic	Death	
Delationalization I		

Relationships: <u>Human Injury</u>

• One and only one Human Injury Level value should be associated with Human Injury.



```
nables excessiv
```



342 3.17 Scope

A coarse measure of the level of impact an exploit could have on a target. In some cases, an impact has no constraints at all. An example of this is a vulnerability with a 'Read (Direct)' *Logical Impact* association in which the adversary has access to the entire system, and thus has no constraints. In other cases, an *Impact* might have some constraints in place. An example of this is 'Write (Direct) *Impact* where the attacker is able to modify resources only accessible by the user.

Limited	There are restrictions to the associated impact	
	Noun-specific relationship: Criticality	
	• One and only one Criticality value shall be associated with	
	Limited	
Unlimited	There are no restrictions to the associated impact	
Relationships: 1	logical Impact, <u>Physical Impact</u>	
• One and	only one Scope value shall be associated with I opical Impact	

• One and only one Scope value shall be associated with Logical Impact

• One and only one Scope value shall be associated with Physical Impact

343

344 **3.18 Criticality**

A measure of the relative importance of the associated *Scope*. This noun group is only relevant when the *Scope* has a value of 'Limited'. When *Scope* is 'Limited', *Criticality* must be used in order to provide additional information about its importance.

Criticality must be considered in concert with the *Context* to which it is associated. That is, for a given *Context* (such as *Guest OS* or *Application*), the *Criticality* should reflect how significant an associated impact could be for the specific *Context*. An impact in a '*Guest OS*' *Context* may be of lower significance than the same impact in a '*Host OS*' *Context* and should be reflected accordingly by its associated *Criticality*.

Low The impact is relatively insignificant.	
---	--

High The impact is relatively significant.

Relationships: <u>Scope</u>

• One and only one Criticality value shall be associated with Limited



М	R	0
---	---	---

346 **4** Conclusion

347 This first draft of this document provides one possible framework for describing vulnerabilities.

348 It is expected that comments on this draft will significantly influence the framework and as the

349 document evolves it will reflect a broad consensus. Future drafts will continue to refine all

350 aspects of the framework including alternative noun groups, noun group values, or even a

351 wholesale change in approach if necessary.

353 Appendix A—Example Usage

354 This appendix is intended to be an informative section describing one way on how to apply the framework to task of

describing a vulnerability. This section will continue to be updated as the framework evolves.

CVE-2012-1516

VMware host memory overwrite vulnerability (data pointers)

Due to a flaw in the handler function for RPC commands, it is possible to manipulate data pointers within the VMX process. This vulnerability may allow a guest user to crash the VMX process or potentially execute code on the host.

Vulnerability:	cve.mitre.org	CVE-2012-1516
----------------	---------------	---------------

Provenance: http://www.vmware.com/security/advis	sories/VMSA-2012-0009.html
Scenario: 1	The first scenario
Product:	Scenario 1 is in relation to the bare metal hypervisor
cpe.nist.gov	products
cpe:2.3:a:vmware:esx:4.0:*:*:*:*:*:*:*	
cpe:2.3:a:vmware:esx:4.1:*:*:*:*:*:*:*	
cpe:2.3:a:vmware:esx:3.5:*:*:*:*:*:*:*	
cpe:2.3:a:vmware:esxi:4.0:*:*:*:*:*:*:*	
cpe:2.3:a:vmware:esxi:4.1:*:*:*:*:*:*:*	
cpe:2.3:a:vmware:esxi:3.5:*:*:*:*:*:*:*	
Attack Theater: Remote	Malformed RPC commands are sent from the Guest
Remote Type: Intranet	OS to the Hypervisor
Barrier: Privilege Required	The attacker must first have user access to a GuestOS
Privilege Level: User	to launch the attack
Relating to Context: GuestOS	
Context: GuestOS	One of the <i>Contexts</i> with recognized impacts due to the
	vulnerability
Entity Role: Primary Authorization	The GuestOS is where the attack is launched and
	represents the first authorization scope
Impact Method: Code Execution	Direct result of failed code execution would be a crash
Logical Impact: Service Interrupt	of the Hypervisor and inherent crash of the GuestOS.
Location: Memory	Since the GuestOS would be completely taken offline,
Service Interrupt Type: Panic	the criticality is listed as High
Scope: Limited	
Criticality: High	
Context: Hypervisor	Another Context with recognized impacts due to the
J. T. T. J. T.	vulnerability
Entity Role: Vulnerable	Based on the description the Hypervisor is what is
	considered vulnerable.
Entity Roles Secondary Authorization	The hypervisor represents an authorization boundary
5	that is different from the GuestOS
Impact Method: Trust Failure	The Hypervisor fails to ensure that the data is in a form
Trust Failure Type: Failure to Verify Content	that prevents unintended Code Execution
Impact Method: Code Execution	L
Logical Impact: Read(Direct)	The information supplied does not explicitly explain
Scope: Limited	the extent of the code execution which results in each
Criticality: High	plausible logical impact being enumerated. If
Logical Impact: Write(Direct)	limitations to this code execution existed, it should be
Scope: Limited	reflected in these <i>Logical Impacts</i> .
Criticality: High	
Logical Impact: Service Interrupt	
Logical impact. Service interrupt	

Scope: Limited Criticality: High	
Logical Impact: Resource Removal(Data)	
Scope: Limited	
Criticality: High	
Scenario: 2	The second scenario
Product:	Scenario 2 is in relation to application based
cpe.nist.gov	Hypervisors
cpe:2.3:a:vmware:workstation:7.1.5:*:*:*:*:*:*:*	
cpe:2.3:a:vmware:player:3.1.6:*:*:*:*:*:*:*	
Attack Theater: Local	Malformed RPC commands are sent from the Guest
	OS to the Hypervisor, but by nature of the product
	everything is local to the HostOS where everything has been installed.
Barrier: Privilege Required	The attacker must first have user access to a GuestOS
Privilege Level: User	to launch the attack
Relating to Context: GuestOS	to numeri the utuer
Context: GuestOS	One of the <i>Contexts</i> with recognized impacts due to the
	vulnerability
Entity Role: Primary Authorization	The GuestOS is where the attack is launched and
, , , , , , , , , , , , , , , , , , ,	represents the first authorization scope
Impact Method: Code Execution	Direct result of failed code execution would be a crash
	of the Hypervisor and inherent crash of the GuestOS.
	— Since the GuestOS would be completely taken offline,
Logical Impact: Service Interrupt	the criticality is listed as High
Location: Memory	
Service Interrupt Type: Panic	
Scope: Limited	
Criticality: High Context: Hypervisor	Another <i>Context</i> with recognized impacts due to the
Context. Hypervisor	vulnerability
Entity Role: Vulnerable	Based on the description the Hypervisor is what is
	considered vulnerable.
Entity Role: Secondary Authorization	The hypervisor represents an authorization boundary
	that is different from the GuestOS
Impact Method: Trust Failure	The Hypervisor fails to ensure that the data is in a form
Trust Failure Type: Failure to Verify Content	that prevents unintended Code Execution
Impact Method: Code Execution	
Logical Impact: Read(Direct)	The information supplied does not explicitly explain the extent of the code execution which results in each
Scope: Limited	the extent of the code execution which results in each plausible logical impact being enumerated. If
Criticality: High Logical Impact: Write(Direct)	limitations to this code execution existed, it should be
Scope: Limited	reflected in these Logical Impacts.
Criticality: High	
Logical Impact: Service Interrupt	
Scope: Limited	
Criticality: High	
Logical Impact: Resource Removal(Data)	
Logical Impact: Resource Removal(Data) Scope: Limited	

CVE-2015-1863

A vulnerability was found in how wpa_supplicant uses SSID information parsed from management frames that create or update P2P peer entries (e.g., Probe Response frame or number of P2P Public Action frames). SSID field has valid length range of 0-32 octets. However, it is transmitted in an element that has a 8-bit length field and potential maximum payload length of 255 octets. wpa_supplicant was not sufficiently verifying the payload length on one of the code paths using the SSID received from a peer device.

This can result in copying arbitrary data from an attacker to a fixed length buffer of 32 bytes (i.e., a possible overflow of up to 223 bytes). The SSID buffer is within struct p2p_device that is allocated from heap. The overflow can override couple of variables in the struct, including a pointer that gets freed. In addition about 150 bytes (the exact length depending on architecture) can be written beyond the end of the heap allocation.

This could result in corrupted state in heap, unexpected program behavior due to corrupted P2P peer device information, denial of service due to wpa_supplicant process crash, exposure of memory contents during GO Negotiation, and potentially arbitrary code execution.

Provenance: http://w1.fi/security/2015-1/wpa_supplican	t-p2p-ssid-overflow.txt
Scenario: 1	The first scenario
Type: cve.mitre.org CWE-119	
Product:	
cpe.nist.gov	
cpe:2.3:a:w1.fi:wpa_supplicant:1.0	
cpe:2.3:a:w1.fi:wpa_supplicant:1.1	
cpe:2.3:a:w1.fi:wpa_supplicant:2.0	
cpe:2.3:a:w1.fi:wpa_supplicant:2.1	
cpe:2.3:a:w1.fi:wpa_supplicant:2.2	
cpe:2.3:a:w1.fi:wpa_supplicant:2.3	
cpe:2.3:a:w1.fi:wpa_supplicant:2.4	
Attack Theater: Limited Remote	The attacker must be within radio range
Remote Type: Wireless	
Barrier: Specialized Condition	CONFIG_P2P build option must be enabled
Context: Application	
Entity Role: Primary Authorization	The Application is the only authorization scope
Entity Role: Vulnerable	
Impact Method: Trust Failure	The Code Execution can lead to limited read of
Trust Failure Type: Failure to Verify Content	memory, crash of the process or unexplored other
Impact Method: Code Execution	outcomes.
Logical Impact: Service Interrupt	
Service Interrupt Type: Panic	
Scope: Limited	
Criticality: High	
Logical Impact: Read(Direct)	
Location: Memory	
Scope: Limited	
Criticality: Low	
Logical Impact: Write(Direct)	
Scope: Limited	
Criticality: High	

Vulnerability: cve.mitre.org CVE-2015-1863

CVE-2015-5611

Unspecified vulnerability in Uconnect before 15.26.1, as used in certain Fiat Chrysler Automobiles (FCA) from 2013 to 2015 models, allows remote attackers in the same cellular network to control vehicle movement, cause human harm or physical damage, or modify dashboard settings via vectors related to modification of entertainment-system firmware and access of the CAN bus due to insufficient "Radio security protection," as demonstrated on a 2014 Jeep Cherokee Limited FWD.

Vulnerability	. cve mitre org	CVE-2015-1863
vunciaonity	. eve.innue.org	$CVL^{-2013-1003}$

Provenance: http://illmatics.com/Remote%20Car%2	20Hacking.pdf
Scenario: 1	The first scenario
Product: cpe.nist.gov cpe:2.3:a:fca:uconnect:15.26.1:*:*:*:*:*:*:*	
epe.2.5.a.rea.uconneet.15.20.1.	
Attack Theater: Limited Remote	The attacker must be on the same cellular network as
Limited Remote Type: Cellular	the target
Context: Application	
Entity Role: Primary Authorization	The Application is the only authorization scope
Entity Role: Vulnerable	
Impact Method: Trust Failure Trust Failure Type: Failure of Inherent Trust Impact Method: Code Execution Logical Impact: Read(Direct) Scope: Limited Criticality: High Logical Impact: Write(Direct) Scope: Limited Criticality: High Logical Impact: Service Interrupt	Anonymous access to the D-bus service allows execution of arbitrary code. This code execution allows modification of lateral internal devices, bricking of chipset or issuing of basic commands. Once these actions are taken, an attacker can control most aspects of the vehicle such as AC, radio and even physical functions such as steering and braking.
Scope: Unlimited	
Logical Impact: Resource Removal (Data) Scope: Limited Criticality: High	
Physical Impact: Human Injury	
Human Injury Type: Critical	
Physical Impact: Property Damage	
Scope: Unlimited	

CVE-2014-8606

Directory traversal vulnerability in the XCloner plugin 3.1.1 for WordPress and 3.5.1 for Joomla! allows remote administrators to read arbitrary files via a .. (dot dot) in the file parameter in a json_return action in the xcloner_show page to wp-admin/admin-ajax.php.

Vulnerability	· cve mitre org	CVE-2014-8606
v uniciaonity	. eve.mue.org	C 1 L 2014 0000

Provenance: http://www.vapid.dhs.org/advisories/wordg	press/plugins/Xcloner-v3.1.1/
Scenario: 1	
Type: cve.mitre.org CWE-22	
Products:	
cpe.nist.gov	
cpe:2.3:a:xcloner:xcloner:3.1.1:*:*:*:wordpress:*:*	
cpe:2.3:a:xcloner:xcloner:3.5.1:*:*:*:joomla\!:*:*	
Attack Theater: Remote	The attack can be launched from the Internet
Remote Type: Internet	
Barriers: Privilege Required	The attacker is required to have administrator rights
Privilege Level: Administrator	within the application prior to exploit
Relating to Context: Application	
Context: Application	
Entity Roles: Primary Authorization	The Application is the initial authorization scope
Entity Roles: Vulnerable	
Impact Method: Trust Failure	The attack can read files on the HostOS, which implies
Trust Failure Type: Failure to Verify Content	some file read realative to the Application as well.
Logical Impact: Read(Direct)	Since the user is already an administrator of the
Scope: Limited	application, the criticality is Low
Criticality: Low	
Context: HostOS	
Entity Roles: Secondary Authorization	
Impact Method: Code Execution	
Logical Impact: Read(Direct)	The attack can read files on the HostOS. Since the file
Scope: Limited	in the example supplied is etc/passwd the criticality can
Criticality: High	be High.

CVE-2015-3459

The communication module on the Hospira LifeCare PCA Infusion System before 7.0 does not require authentication for root TELNET sessions, which allows remote attackers to modify the pump configuration via unspecified commands.

Vulnerability: cve.mitre.org CVE-2015-3459

Provenance: http://www.fda.gov/MedicalDevices/Safety/Alertsance Sector of Interest: Health Care	inolices/ucili440809.intili
Scenario: 1	The first scenario
Type: cve.mitre.org CWE-306	The attack takes advantage of a lack of authentication on the telnet service
Product: cpe.nist.gov cpe:2.3:o:hospira:lifecare_pcainfusion_firmware:5.0:*:*:*:*:*:* Attack Theater: Remote	The attack can be launched from the interne
Remote Type: Internet Context: Host OS	The vulnerability is in the underlying host OS that provides the remote programming capability for the pump
Entity Role: Primary Authorization Entity Role: Vulnerable	The Host OS is the initial authorization scope and is also the vulnerable Context
Impact Method: Trust Failure Trust Failure Type: Failure of Inherent Trust Impact Method: Authorization Bypass Logical Impact: Service Interrupt Scope: Unlimited Logical Impact: Read(Direct) Location: File System Scope: Unlimited Logical Impact: Write(Direct) Location: FileSystem Scope: Unlimited	The attack involves remotely taking advantage of the lack of authentifcaiton during use of telnet on the host OS. Since there is no authorization, this is a exploitation of a trust relationship. This car lead to unspecified types of service interruption and the ability to view and modify the pump's configuration.
Physical Impact: Human Injury	The attack can result in the delivery of an incorrect, and possible deadly level of medicine

363

366 Appendix B—Conversion to descriptive text (English)

367 This appendix will include an informative demonstration of how to convert the framework

368 selections and values into English text. Future drafts will include this information.

369 Appendix C—Mapping VDO representations to CVSS Scores

370 One of the motivations for the VDO is to assist in the automation of CVSS scores. Currently the NVD is responsible 371 for manually consolidating public records and performing analysis on the information available. One of the 372 challenges of performing the analysis is that information supplied is usually lacking in sufficient detail, conflicts 373 with other reports or contains misinformation due to different perspectives. The most notable reason for this 374 challenge is that vulnerability reporting has existed in a mostly free text format. With a defined vocabulary and 375 format for reporting the characterization of a vulnerability, the NVD would be able to automate the scoring process. 376 Below are a few examples of how this would be accomplished at a high level. The following description is only 377 intended to serve as a proof of concept until the VDO itself is in a more static and community agreed upon state. 378 379 NVD intends to create a system that will establish this style of mapping through an expression language. In their 380 simplest form, this would be represented as a series of qualifying statements. Some of which would be as simple as a 381 1:1 mapping and others being a far more complex expression. Using one of the simpler examples from Appendix A

382 (CVE-2014-8606) we can walk through the process similar to how the expressions would operate.

383 384

Using the metrics established in Appendix A, we can break this down into the components currently relevant to a
 CVSS v2.0 score.

VDO Metrics	CVSS v2.0	Reasoning
Attack Theater: Remote	AV:N	The remote attack theater is in line with the definition for the Attack Vector: Network CVSS metric.
Barriers: Privileges Required	Au:S	Only one layer of privilege is required, so it meets the definition for the Authentication: Single CVSS metric.
Context: Application Logical Impact: Read(Direct) Scope: Limited Criticality: Low	C:P	In regards to the application, there is a read available of Low Criticality and a Scope of Limited. This does not grant any reason to go past the Confidentiality: Partial CVSS metric.
Context: HostOS Logical Impact: Read(Direct) Scope: Limited Criticality: High	C:P	CVSS v2.0 scores are relative to the host device the vulnerability has been exploited on. In regards to the HostOS, there is a read available of High Criticality. While the information gained may be considered of great importance, the Scope is Limited and still constitutes the Confidentiality: Partial CVSS metric.

Now we have the metrics we know mapped, we simply fill in the blanks for the metric strings.

First we will establish the non-impact metrics:	Non-Impact metrics:	AV:N/Au:S/AC:L
Then the impact metrics for each context:	Application Context Score:	C:P/I:N/A:N
	HostOS Context Score:	C:P/I:N/A:N
Then join the two:	Application Context Score:	AV:N/Au:S/AC:L/C:P/I:N/A:N
	HostOS Context Score:	AV:N/Au:S/AC:L/C:P/I:N/A:N

The last step once each score has been enumerated is to establish which score to use. CVSS v2.0 is specifically designed to score in relation to the host device. In our example we happen to have a Context of HostOS enumerated, which makes our choice of vector string simple.

CVE-2014-8606 CVSS v2.0 Score: AV:N/Au:S/AC:L/C:P/I:N/A:N

389	Mapping to a CVSS v3.0 score would follow a similar path	h.
-----	--	----

VDO Metrics	CVSS v3.0 Mapping	Reasoning
Attack Theater: Remote	AV:N	The remote attack theater is in line with the definition for the Attack Vector: Network CVSS metric.
Barrier: Privilege Required Privilege Level: Administrator	PR:H	The privilege level of the user must be of administrator to the application, this qualifies for the Privileges Required: High CVSS metric
Context: Application Entity Role: Primary Authorization Logical Impact: Read(Direct) Scope: Limited Criticality: Low	C:L	The vulnerability allows for limited read to files within the applications authorization scope. Due to the low criticality, this qualifies for Confidentiality: Low
Context: HostOS Entity Role: Secondary Authorization Logical Impact: Read(Direct) Scope: Unlimited	C:H S:C	The vulnerability allows for seemingly unlimited read within the filesystem of the HostOS, this is inherently of high criticality and qualifies for Confidentiality: High When multiple contexts exist, it is imperative to check if there are multiple authorization scopes. In this scenario the Application represents the Primary Scope and the HostOS represents the Secondary scope. When impacts are recognized across multiple authorization scopes the vulnerability qualifies for the Scope: Changed CVSS v3.0 Metric.

In a similar fashion to how we created the v3.0 score we will first establish the non-Impact metrics:

First we will establish the non-impact metrics:	Non-Impact metrics:	AV:N/AC:N/PR:H/UI:N/S:C
Then the impact metrics for	Application Context Score:	C:L/I:N/A:N
each context:	HostOS Context Score:	C:H/I:N/A:N
Then join the two:	Application Context Score:	AV:N/AC:N/PR:H/UI:N/S:C/C:L/I:N/A:N
	HostOS Context Score:	AV:N/AC:N/PR:H/UI:N/S:C/C:H/I:N/A:N

Due to the nature of the CVSS v3.0 ruleset, the proper course of action when a scope change occurs is to take the highest rated impact as the score. Therefore we, again, use the HostOS vector string.

CVE-2014-8606 CVSS v2.0 Score: AV:N/Au:S/AC:L/C:P/I:N/A:N

392 Appendix D—Acronyms

394

393 Selected acronyms and abbreviations used in this paper are defined below.

ABNF	Augmented Backus-Naur Form
ASLR	Address space layout randomization
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DNS	Domain Name System
НРКР	HTTP Public Key Pinning
HSTS	HTTP Strict Transport Security
HTTP	Hypertext Transfer Protocol
OWASP	Open Web Application Security Project
RDP	Remote Desktop Protocol
RFC	Request for Comments
SSH	Secure Shell

Appendix E—References		
[CPE23AL]	Waltermire, D., Cichonski, P., and Scarfone, K. Common Platform Enumeration: Applicability Language Specification 2.3. National Institute of Standards and Technology Interagency Report 7698, August 2011. http://csrc.nist.gov/publications/nistir/ir7698/NISTIR-7698-CPE- Language.pdf [accessed 09/09/2016].	
[CPE23M]	Parmelee, M., Booth, H., Waltermire, D., and Scarfone, K. Common Platform Enumeration: Name Matching Specification 2.3. National Institute of Standards and Technology Interagency Report 7696, August 2011. http://csrc.nist.gov/publications/nistir/ir7696/NISTIR-7696-CPE- Matching.pdf [accessed 09/09/2016].	
[CPE23N]	Cheikes, B. A., Waltermire, D., and Scarfone, K. Common Platform Enumeration: Naming Specification 2.3. National Institute of Standards and Technology Interagency Report 7695, August 2011. http://csrc.nist.gov/publications/nistir/ir7695/NISTIR-7695-CPE- Naming.pdf [accessed 09/09/2016].	
[CVSSV3]	Common Vulnerability Scoring System (CVSS) v3.0: Specification Document (v1.7), Forum of Incident Response and Security Teams (FIRST) Common Vulnerability Scoring System (CVSS) Special Interest Group (SIG), https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf, [accessed 09/23/2016].	
[CWE]	Common Weakness Enumeration, MITRE [website], https://cwe.mitre.org, [accessed 09/09/2016].	
[ISO/IEC 14971:2007]	International Organization for Standardization/International Electrotechnical Commission, Medical Devices Application of risk management to medical devices, ISO/IEC 14971:2007, March 2007. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csn umber=38193 [accessed 09/23/2016].	
[ISO/IEC 19770-2:2015]	International Organization for Standardization/International Electrotechnical Commission, Information technology Software asset management Part 2: Software identification tag, ISO/IEC 19770- 2:2015, October 2015. http://www.iso.org/iso/catalogue_detail?csnumber=65666 [accessed 09/09/2016].	
[NISTIR 7298]	Richard Kissel, <i>Glossary of Key Information Security Terms</i> , NISTIR 7298 Revision 2, National Institute of Standards and Technology, Gaithersburg, Maryland, May 2013, 218pp. https://dx.doi.org/10.6028/NIST.IR.7298r2 [accessed 09/09/2016].	

[OWASP- VULN]	Category:Vulnerability, Open Web Application Security Project [website], https://www.owasp.org/index.php/Category:Vulnerability, [accessed 09/09/2016].
[RFC2119]	S. Bradner, <i>Key words for use in RFCs to Indicate Requirement Levels</i> , Internet Engineering Task Force (IETF) Network Working Group Request for Comments (RFC) 2119, March 1997. https://tools.ietf.org/html/rfc2119 [accessed 09/06/2016].
[RFC5234]	D. Crocker and P. Overell, <i>Augmented BNF for Syntax Specifications:</i> <i>ABNF</i> , Internet Engineering Task Force (IETF) Network Working Group Request for Comments (RFC) 5234, January 2008. https://tools.ietf.org/html/rfc5234 [accessed 09/06/2016].