

NIST SPECIAL PUBLICATION 1800-17C

Multifactor Authentication for E-Commerce

Risk-Based, FIDO Universal Second Factor Implementations for Purchasers

Volume C:
How-To Guides

William Newhouse

Information Technology Laboratory
National Institute of Standards and Technology

Brian Johnson

Sarah Kinling

Blaine Mulugeta

Kenneth Sandlin

The MITRE Corporation
McLean, VA

August 2018

DRAFT

This publication is available free of charge from:

<https://nccoe.nist.gov/projects/use-cases/multifactor-authentication-ecommerce>



DISCLAIMER

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

National Institute of Standards and Technology Special Publication 1800-[17C], Natl. Inst. Stand. Technol. Spec. Publ. 1800-[17C], 180 pages, (August 2018), CODEN: NSPUE2

FEEDBACK

You can improve this guide by contributing feedback. As you review and adopt this solution for your own organization, we ask you and your colleagues to share your experience and advice with us.

Comments on this publication may be submitted to: consumer-nccoe@nist.gov.

Public comment period: August 22, 2018 through October 22, 2018.

All comments are subject to release under the Freedom of Information Act (FOIA).

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
100 Bureau Drive
Mailstop 2002
Gaithersburg, MD 20899
Email: nccoe@nist.gov

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE), a part of the National Institute of Standards and Technology (NIST), is a collaborative hub where industry organizations, government agencies, and academic institutions work together to address businesses' most pressing cybersecurity issues. This public-private partnership enables the creation of practical cybersecurity solutions for specific industries, as well as for broad, cross-sector technology challenges. Through consortia under Cooperative Research and Development Agreements (CRADAs), including technology partners—from Fortune 50 market leaders to smaller companies specializing in IT security—the NCCoE applies standards and best practices to develop modular, easily adaptable example cybersecurity solutions using commercially available technology. The NCCoE documents these example solutions in the NIST Special Publication 1800 series, which maps capabilities to the NIST Cybersecurity Framework and details the steps needed for another entity to recreate the example solution. The NCCoE was established in 2012 by NIST in partnership with the State of Maryland and Montgomery County, Md.

To learn more about the NCCoE, visit <https://www.nccoe.nist.gov/>. To learn more about NIST, visit <https://www.nist.gov>.

NIST CYBERSECURITY PRACTICE GUIDES

NIST Cybersecurity Practice Guides (Special Publication Series 1800) target specific cybersecurity challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the adoption of standards-based approaches to cybersecurity. They show members of the information security community how to implement example solutions that help them align more easily with relevant standards and best practices, and provide users with the materials lists, configuration files, and other information they need to implement a similar approach.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. These documents do not describe regulations or mandatory practices, nor do they carry statutory authority.

ABSTRACT

As retailers in the United States have adopted chip-and-signature and chip-and-PIN (personal identification number) point-of-sale (POS) security measures, there have been increases in fraudulent online card-not-present (CNP) electronic commerce (e-commerce) transactions. The risk of increased fraudulent online shopping became more widely known following the adoption of chip-and-PIN technology that increased security at the POS in Europe.

The NCCoE at NIST built a laboratory environment to explore methods to implement multifactor authentication (MFA) for online retail environments for the consumer and the e-commerce platform

administrator. The NCCoE also implemented logging and reporting to display authentication-related system activity.

This NIST Cybersecurity Practice Guide demonstrates to online retailers that it is possible to implement open standards-based technologies to enable Universal Second Factor (U2F) authentication at the time of purchase when risk thresholds are exceeded.

The example implementations outlined in this guide encourage online retailers to adopt effective MFA implementations by using standard components and custom applications that are composed of open-source and commercially available components.

KEYWORDS

electronic commerce (e-commerce) security; internet shopping security; multifactor authentication (MFA)

ACKNOWLEDGMENTS

We are grateful to the following individuals for their generous contributions of expertise and time.

Name	Organization
Greg Dicovitsky	RSA
Leonardo Andrade	RSA
Adam Cohn	Splunk
Arshad Noor	StrongKey
Kamil Kreiser	TokenOne
Derek Hanson	Yubico
Brian Abe	The MITRE Corporation
Lorrayne Auld	The MITRE Corporation
Lura Danley	The MITRE Corporation

Name	Organization
Sallie Edwards	The MITRE Corporation
Charles Jones, Jr.	The MITRE Corporation
Joshua Klosterman	The MITRE Corporation
Jay Vora	The MITRE Corporation
Mary Yang	The MITRE Corporation

The Technology Partners/Collaborators who participated in this build submitted their capabilities in response to a notice in the Federal Register. Respondents with relevant capabilities or product components were invited to sign a Cooperative Research and Development Agreement (CRADA) with NIST, allowing them to participate in a consortium to build these example implementations. We worked with:

Technology Partner/Collaborator	Build Involvement
RSA	RSA Adaptive Authentication (Cloud) Version 13.1
Splunk	<ul style="list-style-type: none"> • Splunk Enterprise Version 6.6.1 • Splunk DB Connect Version 3.1.2 • Splunk Universal Forwarder Version 7.0.1
StrongKey	<ul style="list-style-type: none"> • StrongKey CryptoEngine (SKCE) Version 2.0 Open Source Fast IDentity Online (FIDO) U2F Server • MagentoFIDO (magfido) 1st Edition Module
TokenOne	TokenOne cloud-based Authentication Version 2.8.5
Yubico	Yubico YubiKey NEO Security Key

Contents

1	1	Introduction	1
2	1.1	Practice Guide Structure	1
3	1.2	Example Builds Overview	2
4	1.2.1	Usage Scenarios	2
5	1.2.2	Architectural Overview	3
6	1.2.3	General Infrastructure Details and Requirements.....	7
7	1.2.3.1	Domain Name System	9
8	1.3	Typographic Conventions.....	10
9	2	How to Install and Configure	11
10	2.1	StrongKey CryptoEngine FIDO U2F Server	11
11	2.1.1	StrongKey CryptoEngine Overview	11
12	2.1.2	SKCE Requirements.....	13
13	2.1.2.1	SKCE Software Requirements.....	13
14	2.1.2.2	Hardware Requirements	14
15	2.1.2.3	Network Requirements	14
16	2.1.3	Install SKCE, the FIDO U2F Authentication Server.....	14
17	2.2	Magento Open Source Electronic Commerce Platform.....	17
18	2.2.1	Magento Overview	19
19	2.2.2	Magento Requirements.....	19
20	2.2.2.1	Software Requirements.....	19
21	2.2.2.2	Hardware Requirements	20
22	2.2.3	Magento Preinstallation	20
23	2.2.4	Magento Installation.....	34
24	2.2.5	Configuring the Magento Account Lockout Feature	44
25	2.2.6	Disabling Magento Guest Checkout.....	49
26	2.3	StrongKey magfido Module.....	51
27	2.3.1	StrongKey magfido Overview	51

28	2.3.2 StrongKey magfido Installation and Configuration.....	53
29	2.4 RSA Adaptive Authentication	62
30	2.4.1 RSA Overview.....	64
31	2.4.2 RSA Preinstallation Steps	64
32	2.4.3 Adaptive Authentication Installation and Configuration.....	72
33	2.4.4 RSA Adaptive Authentication Policy Creation.....	94
34	2.5 TokenOne	98
35	2.5.1 TokenOne Overview	100
36	2.5.2 Preinstallation Steps	100
37	2.5.3 TokenOne Installation and Configuration.....	100
38	2.5.4 TokenOne Provisioning	109
39	2.5.5 Administrator Login with TokenOne Authentication.....	116
40	2.6 Splunk Enterprise	119
41	2.6.1 Splunk Technologies Overview	121
42	2.6.2 Splunk Enterprise	121
43	2.6.2.1 Overview.....	121
44	2.6.2.2 Splunk Enterprise Requirements	121
45	2.6.2.3 Splunk Enterprise: Prepare for Installation	121
46	2.6.2.4 Splunk Enterprise Installation.....	121
47	2.6.3 Splunk Universal Forwarder.....	122
48	2.6.3.1 Splunk Universal Forwarder Overview	122
49	2.6.3.2 Splunk Universal Forwarder Requirements.....	122
50	2.6.3.3 Splunk Universal Forwarder: Prepare for Installation	122
51	2.6.3.4 Splunk Universal Forwarder: Installation	122
52	2.6.4 Splunk DB Connect.....	124

53	2.6.4.1	Overview.....	124
54	2.6.4.2	Splunk DB Connect Requirements.....	124
55	2.6.4.3	Splunk DB Connect Installation	124
56	2.6.4.4	Setup.....	127
57	2.6.4.5	Creating Identities	130
58	2.6.4.6	Creating Connections.....	131
59	2.6.4.7	Creating Inputs	133
60	2.6.4.8	Creating Database Lookups	138
61	2.6.5	Splunk Enterprise Queries and Dashboards	142
62	2.6.5.1	Query: Total Attempted Single-Factor Authentications.....	142
63	2.6.5.2	Query: Failed Single-Factor Authentications Within Past Five Minutes...143	
64	2.6.5.3	Query: Attempted Single-Factor Authentications in Past Five Minutes...143	
65	2.7	Testing FIDO Key Registration and Checkout.....	143
66	2.7.1	Creating an Example Magento Customer Account.....	143
67	2.7.2	FIDO Key Registration	146
68	2.7.3	Testing Customer Checkout.....	149
69	Appendix A	FIDO U2F Security Key Registration	153
70	A.1	Display Function	153
71	A.2	Preregister Function.....	156
72	A.3	Register Function.....	158
73	A.3.1	The Checkout Process	159
74	A.3.2	The FIDO Authentication Flow for the Example Implementations.....	160
75	A.3.3	Information About the magfido Files and Directories	161
76	A.3.4	Solutions to Common Challenges When Configuring Magento and magfido	163
77	A.3.4.1	Code Was Modified but Change Did Not Take Effect.....	163
78	A.3.4.2	Magento Is Unable to Read the WSDL of the FIDO Server.....	164
79	A.3.4.3	Error 500 When Attempting to Access the Home Page	164
80	Appendix B	List of Acronyms	165

81	Appendix C Glossary	167
82	Appendix D References.....	169

83 **List of Figures**

84	Figure 1-1 MFA for E-Commerce High-Level Cost Threshold Reference Architecture.....	4
85	Figure 1-2 MFA for E-Commerce High-Level Risk Engine Reference Architecture	6
86	Figure 1-3 MFA for E-Commerce Lab Network Architecture	8
87	Figure 2-1 StrongKey CryptoEngine Components.....	12
88	Figure 2-2 Magento Open Source E-Commerce Platform Components	18
89	Figure 2-3 StrongKey magfido Module Components.....	52
90	Figure 2-4 RSA Adaptive Authentication Components	63
91	Figure 2-5 TokenOne Authentication Components	99
92	Figure 2-6 Splunk Enterprise Components.....	120
93	Figure A-1 Browser Display Without Any Security Keys Registered	154
94	Figure A-2 Browser Display with Two Security Keys Registered.....	155
95	Figure A-3 Display Function Part of the FIDO Registration Process	156
96	Figure A-4 Preregistration Part of the FIDO Registration Process	157
97	Figure A-5 Third and Final Step of the FIDO Registration Process	158
98	Figure A-6 Magento Checkout Workflow	159
99	Figure A-7 Overview of the FIDO Authentication Process	161

100 **List of Tables**

101	Table 1-1 Cost Threshold Architecture List of Components	5
102	Table 1-2 Risk Engine Architecture List of Components	7
103	Table 1-3 MFA Example Lab Build Network Details.....	9

104	Table 1-4 Lab Network Host Record Information.....	9
105	Table 2-1 Network Ports to Be Enabled.....	14
106	Table 2-2 Local Ports	14

107 1 Introduction

108 The following volume of this guide shows information technology (IT) professionals and security
109 engineers how we implemented the two example implementations. We cover all of the products
110 employed in these reference designs. We do not recreate the product manufacturers' documentation,
111 which is presumed to be widely available and is referenced when needed. Rather, this volume shows
112 how we incorporated the products together in our environment.

113 *Note: These are not comprehensive tutorials. There are many possible service and security configurations
114 for these products that are out of scope for these reference designs.*

115 1.1 Practice Guide Structure

116 This National Institute of Standards and Technology (NIST) Cybersecurity Practice Guide demonstrates
117 standards-based reference designs and provides retailers with the information they need to replicate
118 the multifactor authentication (MFA) for electronic commerce (e-commerce) example implementations.
119 These reference designs are modular and can be deployed in whole or in parts.

120 This guide contains three volumes:

- 121 ■ NIST Special Publication (SP) 1800-17A: *Executive Summary*
- 122 ■ NIST SP 1800-17B: *Approach, Architecture, and Security Characteristics* – what we built and why
- 123 ■ NIST SP 1800-17C: *How-To Guides* – instructions for building the example implementations (**you
124 are here**)

125 Depending on your role in your organization, you might use this guide in different ways:

126 **Business decision makers, including chief security and technology officers**, will be interested in the
127 *Executive Summary*, *NIST SP 1800-17A*, which describes the following topics:

- 128 ■ challenges enterprises face in implementing MFA to reduce online fraud
- 129 ■ example implementations built at the National Cybersecurity Center of Excellence (NCCoE)
- 130 ■ benefits of adopting one or more of these example implementations

131 **Technology or security program managers** who are concerned with how to identify, understand, assess,
132 and mitigate risk will be interested in *NIST SP 1800-17B*, which describes what we did and why. The
133 following sections of Volume B will be of particular interest:

- 134 ■ Section 3.4, Risk Assessment, provides a description of the risk analysis we performed
- 135 ■ Appendix A, Mapping to Cybersecurity Framework, maps NIST and consensus security
136 references to the Cybersecurity Framework subcategories that are addressed in this practice
137 guide. Additionally, work roles in NIST SP 800-181, *National Initiative for Cybersecurity Education*

138 (NICE) Cybersecurity Workforce Framework (National Institute of Standards and Technology
139 (NIST), 2017), that perform the tasks necessary to implement those cybersecurity functions and
140 subcategories were identified.

141 You might share the *Executive Summary*, *NIST SP 1800-17A*, with your leadership team members to help
142 them understand the importance of adopting standards-based solutions when implementing MFA that
143 can increase assurance of who is using the purchaser's credit card and account information.

144 **IT security professionals** who want to implement approaches like these will find the whole practice
145 guide useful. You can use the How-To portion of the guide, *NIST SP 1800-17C*, to replicate all or parts of
146 the build created in our lab. The How-To portion of the guide provides specific product installation,
147 configuration, and integration instructions for deploying the example implementations. We do not
148 recreate the product manufacturers' documentation, which is generally widely available. Rather, we
149 show how we incorporated the products together in our environment to create example
150 implementations.

151 This guide assumes that IT professionals have experience implementing security products within the
152 enterprise. While we have used a suite of commercial products to address this challenge, this guide does
153 not endorse these particular products. Your organization can adopt these example implementations or
154 one that adheres to these guidelines in whole, or you can use this guide as a starting point for tailoring
155 and implementing parts of these e-commerce fraud-reducing capabilities. Your organization's security
156 experts should identify the products that will best integrate with the existing tools and IT system
157 infrastructure. We hope that you will seek products that are congruent with applicable standards and
158 best practices. Volume B, Section 3.5, Technologies, lists the products that we used and maps them to
159 the cybersecurity controls provided by the reference implementations.

160 A NIST Cybersecurity Practice Guide does not describe "the" solution but a possible solution. This is a
161 draft guide. We seek feedback on its contents and welcome your input. Comments, suggestions, and
162 success stories will improve subsequent versions of this guide. Please contribute your thoughts to
163 consumer-nccoe@nist.gov.

164 1.2 Example Builds Overview

165 The NCCoE at NIST built two example laboratory environments to explore MFA options available to
166 online retailers, which are described in this section.

167 1.2.1 Usage Scenarios

168 The example implementations fulfill the use cases of a returning purchaser with established login
169 account credentials with the retailer, and who possesses a Fast IDentity Online (FIDO) Universal Second
170 Factor (U2F) authenticator [1], [2]. The purchaser's U2F authenticator is used when the retailer system
171 requests additional authentication. This gives the retailer additional assurance that the purchaser is a
172 returning customer, when the checkout process occurs in circumstances that exceed the retailer's risk

173 thresholds. In these NCCoE reference architectures, the risk thresholds that initiate MFA requests are
174 based on the total cost of the shopping cart transaction, or upon input received from the risk engine.
175 The NCCoE worked with members of the NCCoE Retail Community of Interest to develop a set of use
176 case scenarios to help design and test the reference implementations. For a detailed description of the
177 example builds' architectures and the use cases that they are based upon, reference Sections 4 and 5 in
178 Volume B.

179 1.2.2 Architectural Overview

180 The MFA for e-commerce high-level reference architectures illustrated in [Figure 1-1](#) and [Figure 1-2](#) show
181 the *cost threshold* and *risk engine* example implementations, respectively. The high-level reference
182 architectures display the data communication among the returning purchaser, retailer e-commerce
183 platform, risk assessment / MFA module and risk engine, MFA mechanisms, and logging and reporting
184 dashboard.

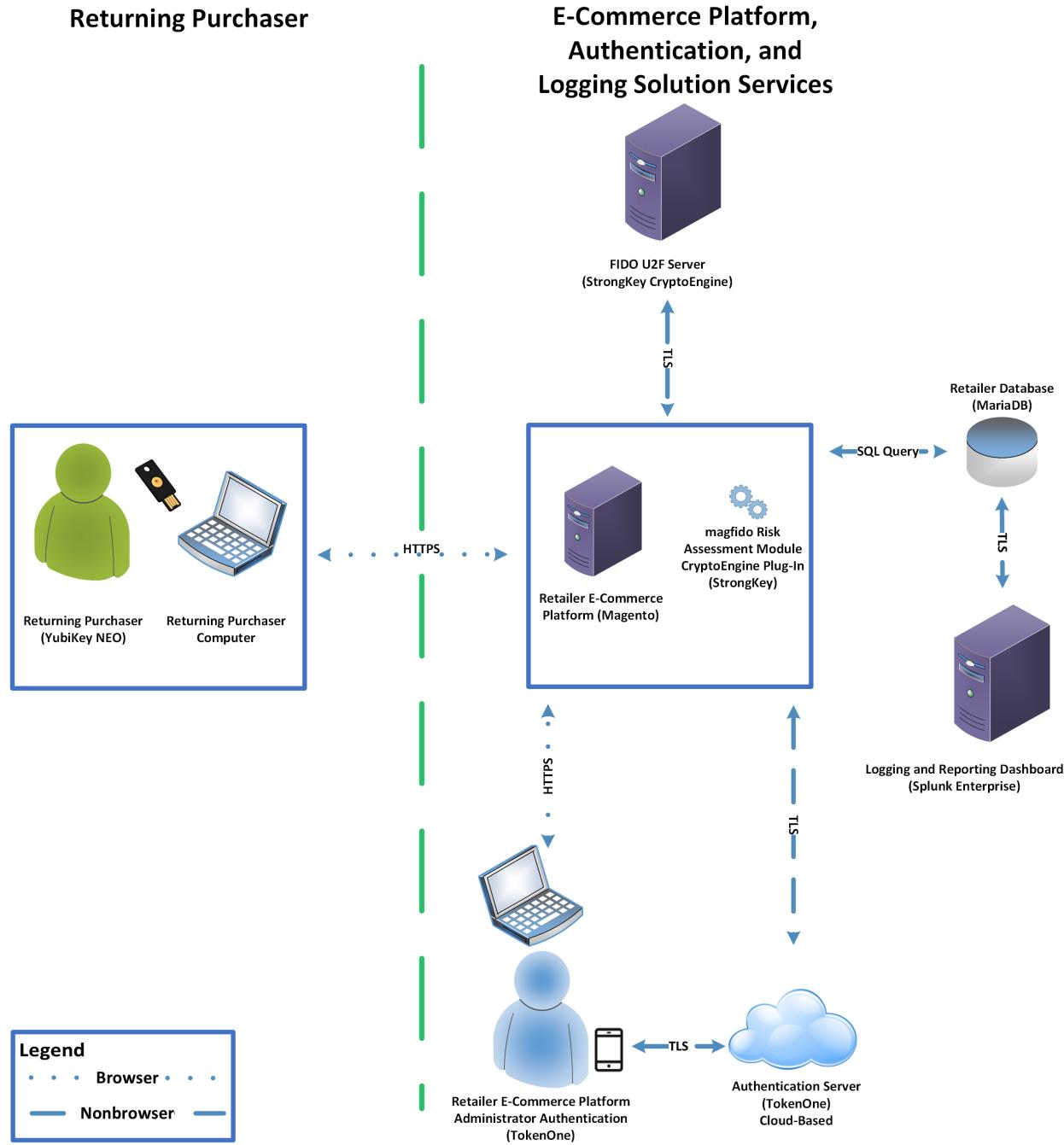
185 The *cost threshold* example implementation uses a predetermined shopping cart price threshold to
186 require the use of MFA by the returning purchaser. The *risk engine* example implementation uses
187 analytics to determine if and when MFA is required by the returning purchaser. The two example
188 implementations include e-commerce platform capabilities, risk assessment and MFA, and logging and
189 display capabilities.

190 The example implementations were constructed on the NCCoE's VMware vSphere virtualization
191 operating environment. Internet access was used to connect to remote cloud-based components, while
192 software components were installed as virtual servers within the vSphere environment.

193 TokenOne's authentication capability authenticates the Magento e-commerce platform administrator
194 before any administration modifications are made to the e-commerce platform. It is based upon
195 TokenOne's cloud-based authentication infrastructure and a smartphone application on either an
196 Android or iPhone device. This helps secure the overall e-commerce organization's infrastructure.

197 The lab network that was used to build and configure the example implementations is not connected to
198 the NIST enterprise network.

199 Figure 1-1 MFA for E-Commerce High-Level Cost Threshold Reference Architecture



200

201 The *cost threshold* example build illustrated in [Figure 1-1](#) uses the components listed in [Table 1-1](#).

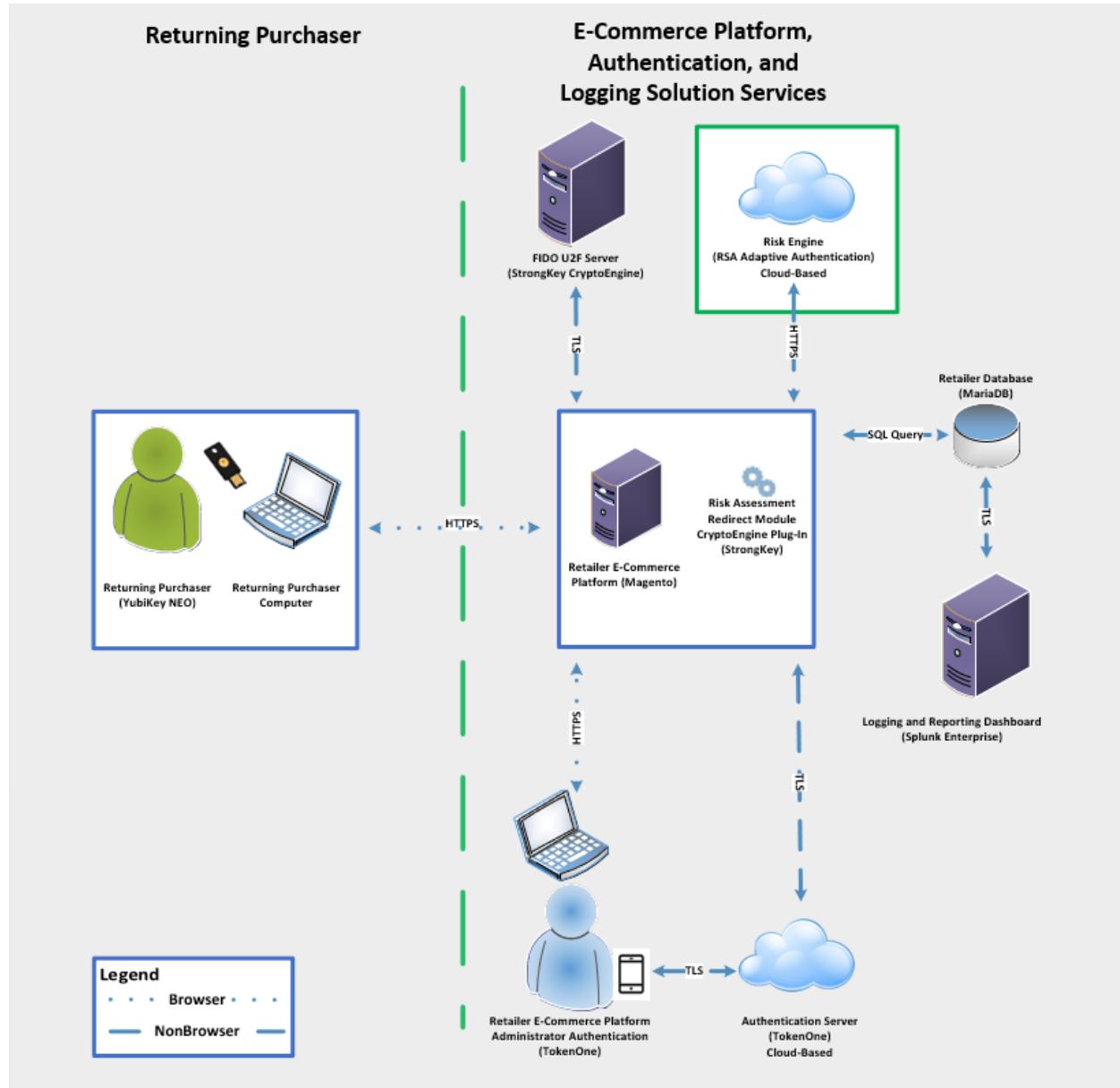
202 **Table 1-1 Cost Threshold Architecture List of Components**

Components	Installation Guidance
StrongKey CryptoEngine (SKCE) FIDO U2F Server and CryptoEngine plug-in	Section 2.1
Magento Open Source e-commerce platform	Section 2.2
StrongKey Magento magfido risk assessment module	Section 2.3
TokenOne Authentication	Section 2.5
Splunk Enterprise logging/reporting dashboard	Section 2.6
Yubico YubiKey NEO Security Key	Section 2.7

203

204

Figure 1-2 MFA for E-Commerce High-Level Risk Engine Reference Architecture



205

206 The *risk engine* example build illustrated in [Figure 1-2](#) uses the components listed in [Table 1-2](#).

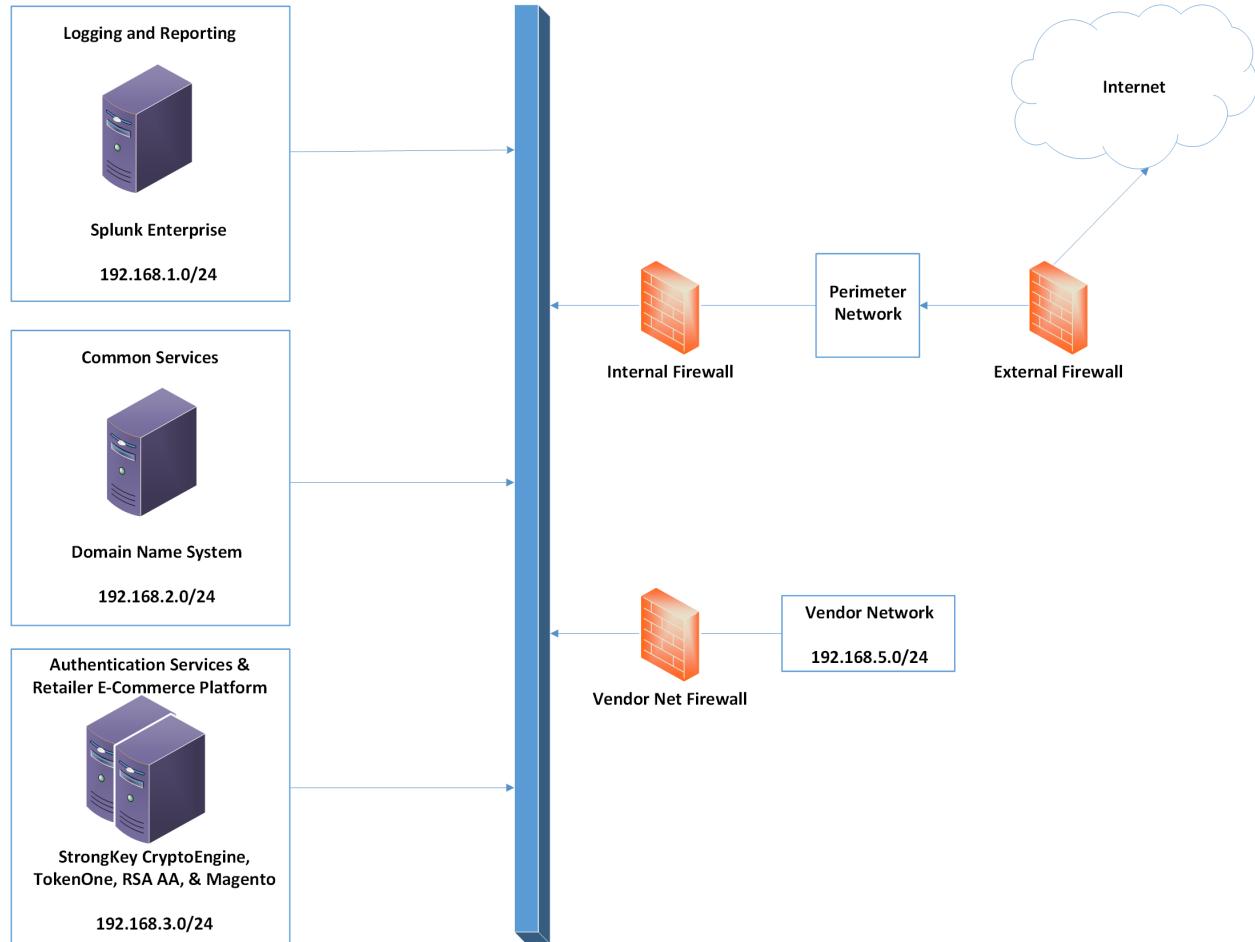
207 **Table 1-2 Risk Engine Architecture List of Components**

Components	Installation Guidance
SKCE FIDO U2F Server and CryptoEngine plug-in	Section 2.1
Magento Open Source e-commerce platform	Section 2.2
RSA Adaptive Authentication	Section 2.4
TokenOne Authentication	Section 2.5
Splunk Enterprise logging/reporting dashboard	Section 2.6
Yubico YubiKey NEO Security Key	Section 2.7

208 **1.2.3 General Infrastructure Details and Requirements**

209 The lab network architecture is shown in [Figure 1-3](#), where the relationship among the MFA example
210 implementation components, firewalls, and network design are illustrated. The installation and
211 configuration for many of the components shown in [Figure 1-3](#) will be referenced in this volume of the
212 guide.

213 **Figure 1-3 MFA for E-Commerce Lab Network Architecture**



214

215 [Table 1-3](#) lists the MFA example lab build's network Internet Protocol (IP) address range, system, and
 216 associated IP addresses. These network addresses were used in the example implementation builds and
 217 will be modified to reflect actual network architectures when deployed into a retailer's information
 218 system network.

219 **Table 1-3 MFA Example Lab Build Network Details**

Network	System	IP Address
192.168.1.0/24	Splunk Enterprise server logging and reporting	192.168.1.10
192.168.2.0/24	Domain Name System (DNS) common services	192.168.2.10
192.168.3.0/24	SKCE FIDO U2F server authentication services	192.168.3.30
192.168.3.0/24	RSA Adaptive Authentication connectivity, TokenOne, Magento Open Source authentication services and retailer e-commerce platform	192.168.3.155
192.168.5.0/24	Optional future services for vendor network	As assigned

220

221 There are both prerequisite infrastructure and example implementation components, whose installation
 222 and configuration are described below.

223 *1.2.3.1 Domain Name System*

224 DNS was configured within the lab to facilitate data communication among the example implementation
 225 components. The domain names and IP address ranges will be modified to reflect actual network
 226 architectures when deployed into an online retailer's information system network.

227 The name of the domain used for this example build is mfa.local. Create the following host records in
 228 the mfa.local forward lookup zone by using the hostnames, fully qualified domain names (FQDNs), and
 229 IP addresses listed in [Table 1-4](#).

230 **Table 1-4 Lab Network Host Record Information**

Hostname	FQDN	IP Address
Splunk	Splunk.mfa.local	192.168.1.10
DNS	DNS.mfa.local	192.168.2.10
Magento	Magento.mfa.local	192.168.3.30
Magento2	Magento2.mfa.local	192.168.3.155

231

232 The network adapter configuration for the DNS server is as follows:

- 233 ▪ Network Configuration (Interface 1)
 - 234 • IPv4 Manual
 - 235 • IPv6 Disabled

- 236 • IP Address: 192.168.2.10
 237 • Netmask: 255.255.255.0
 238 • Gateway: 192.168.2.1
 239 • DNS Name Servers: 192.168.2.10
 240 ▪ DNS-Search Domains: mfa.local

241 1.3 Typographic Conventions

242 The following table presents typographic conventions used in this volume.

Typeface/Symbol	Meaning	Example
<i>Italics</i>	Filenames and pathnames, references to documents that are not hyperlinks, new terms, and placeholders	For detailed definitions of terms, see the <i>NCCoE Glossary</i> .
Bold	names of menus, options, command buttons, and fields	Choose File > Edit .
Monospace	command-line input, on-screen computer output, sample code examples, and status codes	<code>mkdir</code>
Monospace Bold	command-line user input contrasted with computer output	service sshd start
<u>blue text</u>	link to other parts of the document, a web URL, or an email address	All publications from NIST's National Cybersecurity Center of Excellence are available at https://www.nccoe.nist.gov

243 2 How to Install and Configure

244 This section of the practice guide contains detailed instructions for installing and configuring the
245 products used to build the example implementations.

246 2.1 StrongKey CryptoEngine FIDO U2F Server

247 This section of the guide provides installation and configuration guidance for the SKCE, which provides
248 FIDO authentication services.

249 2.1.1 StrongKey CryptoEngine Overview

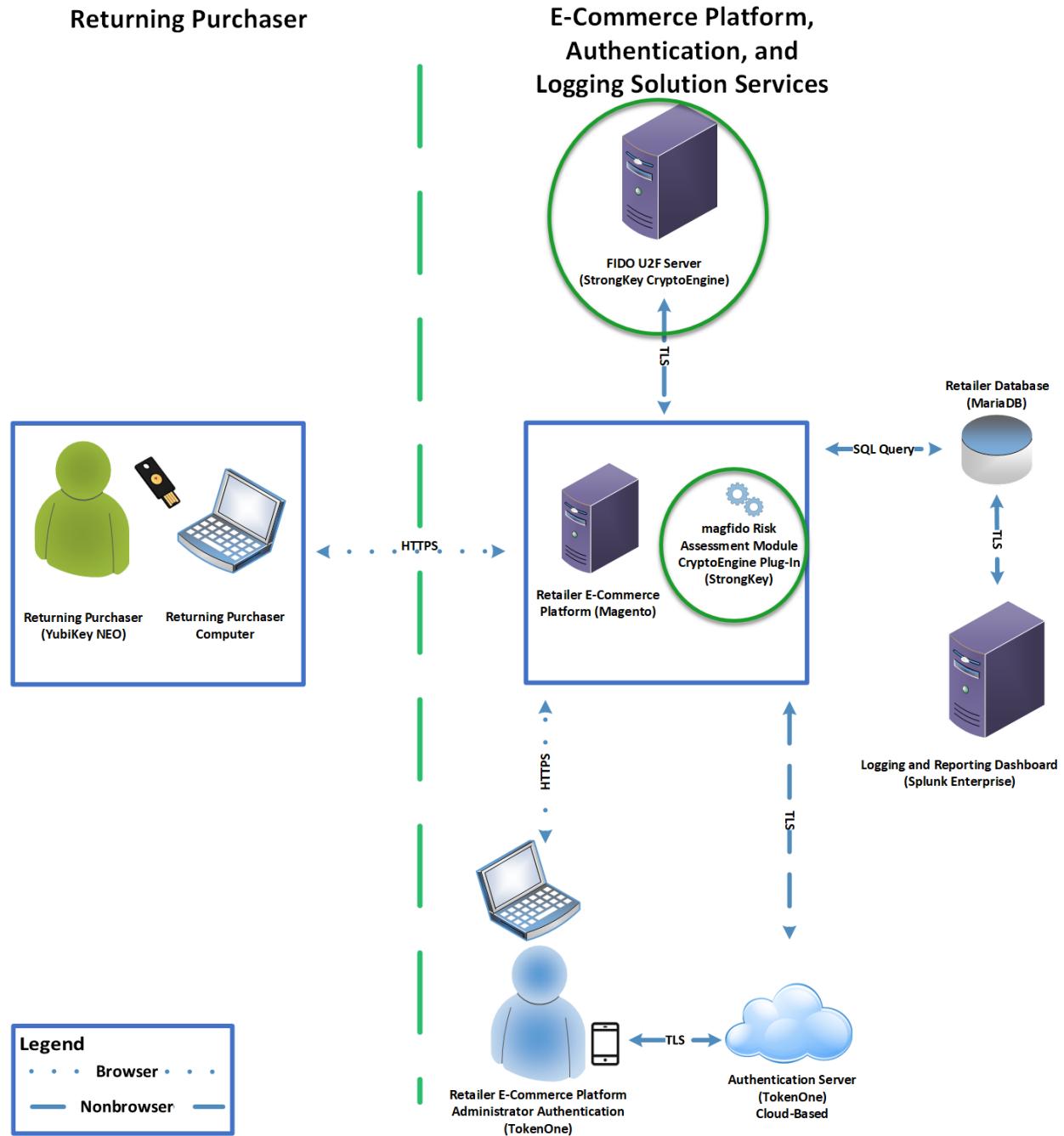
250 The SKCE 2.0 Build 163 from StrongKey [3] performs the FIDO U2F [1], [2] server functionality in the
251 build architecture.

252 StrongKey's main product is the StrongKey Key Appliance, but the company also distributes much of its
253 software under the *Lesser General Public License*, published by the Free Software Foundation. SKCE was
254 downloaded from the StrongKey repository on SourceForge and was used in this build.

255 The CryptoEngine plug-in enables Magento to communicate with the SKCE when the returning
256 purchasers require MFA.

257 Both the *cost threshold* and *risk engine* example implementations use the SKCE's capabilities. The
258 components that are installed by using the instructions in this section are illustrated in [Figure 2-1](#)
259 (circled in green).

260 Figure 2-1 StrongKey CryptoEngine Components



261

262 Installation instructions and the product download site for StrongKey's FIDO U2F server, SKCE, can be
263 found at <https://sourceforge.net/projects/skce/>. For this example implementation, we installed and
264 configured a local copy of SKCE by using the [SKCE installation instructions](#) documented below in
265 [Section 2.1.2](#).

266 **2.1.2 SKCE Requirements**

267 The following subsections document the software, hardware, and network requirements for SKCE
268 Version 2.0.

269 ***2.1.2.1 SKCE Software Requirements***

270 For this build, SKCE was installed on a Community Enterprise Operating System (CentOS) 7.4 64-bit
271 server.

272 Because SKCE is a Java application, it is compatible with operating systems that support a compatible
273 version of Java and the other required software. The application was built with the Oracle Java
274 Development Kit (JDK) Version 8, Update 72. Instructions for obtaining Oracle JDK and the other
275 necessary components are provided in this section.

276 SKCE can be installed manually or with an installation script included in the download. SKCE depends on
277 other software components, including a Structured Query Language (SQL) database, a Lightweight
278 Directory Access Protocol (LDAP) directory server, and the Glassfish Java application server. By default,
279 the script will install MariaDB, OpenDJ, and Glassfish all on a single server.

280 For this build, the scripted installation was used with the default software components. The required
281 software components listed below must be downloaded prior to running the installation script:

- 282 ■ Glassfish 4.1 2010
283 ■ Java Cryptography Extension Unlimited Strength Jurisdiction Policy Files 8 2011
284 ■ JDK 8, Update 121 2012
285 ■ OpenDJ 3.0.0 2013
286 ■ MariaDB 10.1.22 2014
287 ■ MariaDB Java Client 2015

See StrongKey's scripted installation instructions for details and preinstallation software download links:

<https://sourceforge.net/p/skce/wiki/Install%20StrongKey%20CryptoEngine%202.0%20%28Build%20163%29/>.

Note: To download OpenDJ, you must register for a free account for ForgeRock BackStage.

288

2.1.2.2 *Hardware Requirements*

StrongKey recommends installing SKCE on a server with at least 10 gigabytes (GB) of available disk space and 4 GB of random access memory (RAM).

2.1.2.3 *Network Requirements*

The SKCE Application Programming Interface (API) uses Transmission Control Protocol (TCP) Port 8181 ([Table 2-1](#)). Any applications that request U2F registration, authentication, or deregistration actions from the SKCE need to be able to connect on this port. Glassfish runs a Hypertext Transfer Protocol Secure (HTTPS) service on this port. Use firewall-cmd, iptables, or any other system utility for manipulating the firewall to open this port.

298 **Table 2-1 Network Ports to Be Enabled**

Port	Use
TCP 8181	U2F Application Access

299

300 Other network services listen on the ports listed in [Table 2-2](#). For the scripted installation, where all of 301 these services are installed on a single server, there is no need to adjust firewall rules for these services 302 when they are only accessed from localhost.

303 **Table 2-2 Local Ports**

Port	Use
TCP 3306	MariaDB listener
TCP 4848	Glassfish administrative console
TCP 1389	OpenDJ LDAP service

304 [2.1.3 Install SKCE, the FIDO U2F Authentication Server](#)

305 The installation procedure consists of the following steps:

- 306 ■ Download the software dependencies to the server where SKCE will be installed.
307 ■ Make any required changes to the installation script.
308 ■ Run the script as root/administrator.
309 ■ Perform post-installation configuration.

See StrongKey's scripted installation instructions for details and preinstallation software download links:

<https://sourceforge.net/p/skce/wiki/Install%20StrongKey%20CryptoEngine%202.0%20%28Build%20163%29/>.

- 310
- 311 The installation script creates a “strongauth” Linux user and installs all software under
312 `/usr/local/strongauth`. Rather than reproduce the installation steps here, this section provides some
313 notes on the installation procedure:
- 314 1. Download the software. Download and unzip the SKCE build to a directory on the server where
315 SKCE is being installed. Download all installers as directed in the SKCE instructions to the same
316 directory.
- 317 2. Change software versions as required in the install script. If different versions of any of the soft-
318 ware dependencies were downloaded, update the file names in the install script (*install-*
319 *skce.sh*). Using different versions of the dependencies, apart from minor point-release versions,
320 is not recommended. For the lab build, JDK Version 8u151 was used instead of the version refer-
321 enced in the instructions. This required updating the JDK and JDKVER settings in the file.
- 322 3. Change passwords in the install script. Changing the default passwords in the delivered script is
323 strongly recommended. The defaults are readily discoverable, as they are distributed with the
324 software. Passwords should be stored in a password vault or other agency-approved secure
325 storage. Once the installation script has been run successfully, the script should be deleted or
326 sanitized to remove passwords. The following lines in the install script contain passwords:

```
327     LINUX_PASSWORD=ShaZam123                     # For 'strongauth' account
328     GLASSFISH_PASSWORD=adminadmin                 # Glassfish Admin password
329     MYSQL_ROOT_PASSWORD=BigKahuna                 # MySQL 'root' password
330     MYSQL_PASSWORD=AbracaDabra                     # MySQL 'skles' password
331     SKCE_SERVICE_PASS=Abcd1234!                  # Webservice user 'service-cc-ce' password
332     SAKA_PASS=Abcd1234!
333     SERVICE_LDAP_BIND_PASS=Abcd1234!
334     SEARCH_LDAP_BIND_PASS=Abcd1234!
```

335 4. Set the App ID (identifier) Uniform Resource Locator (URL): The App ID setting in *install-skce.sh*
336 should point to a URL that will be accessible to clients where the *app.json* file can be down-
337 loaded. The default location is a URL on the SKCE server, but the SKCE would not be exposed to
338 mobile clients in a typical production deployment. In the lab, *app.json* was hosted on the follow-
339 ing SKCE server:

340 */usr/local/strongauth/payara41/glassfish/domains/domain1/docroot/app.json*

341 This enables the file to be accessed by clients at the following URL: *https://magento.mfa.lo-*
342 *cal:8181/app.json*.

343 5. Run the script. *install-skce.sh* must be run as the root user. If the install script terminates with an
344 error, then troubleshoot and correct any problems before continuing.

345 6. (For CentOS 7) create the firewall rule. The install script attempts to open the required port by
346 using iptables, which does not work on CentOS 7. In that case, the following commands will
347 open the port:

```
348 # firewall-cmd --permanent --add-port 8181/tcp  
349 success  
350 # firewall-cmd --reload  
351 success
```

352 7. Restart Glassfish. On CentOS 7, run the following command:

```
353 $ sudo systemctl restart glassfishd
```

354 8. Complete Step 3b in the SKCE installation instructions to activate the cryptographic module.

355 9. Complete Step 3c in the SKCE installation instructions to create the domain signing key. When
356 prompted for the App ID, use the URL referenced above in the App ID setting of the *install-*
357 *skce.sh* script.

358 10. Complete Step 4 in the SKCE installation instructions if secondary SKCE instances are being in-
359 stalled; this was not done for this build, but is recommended for a production installation.

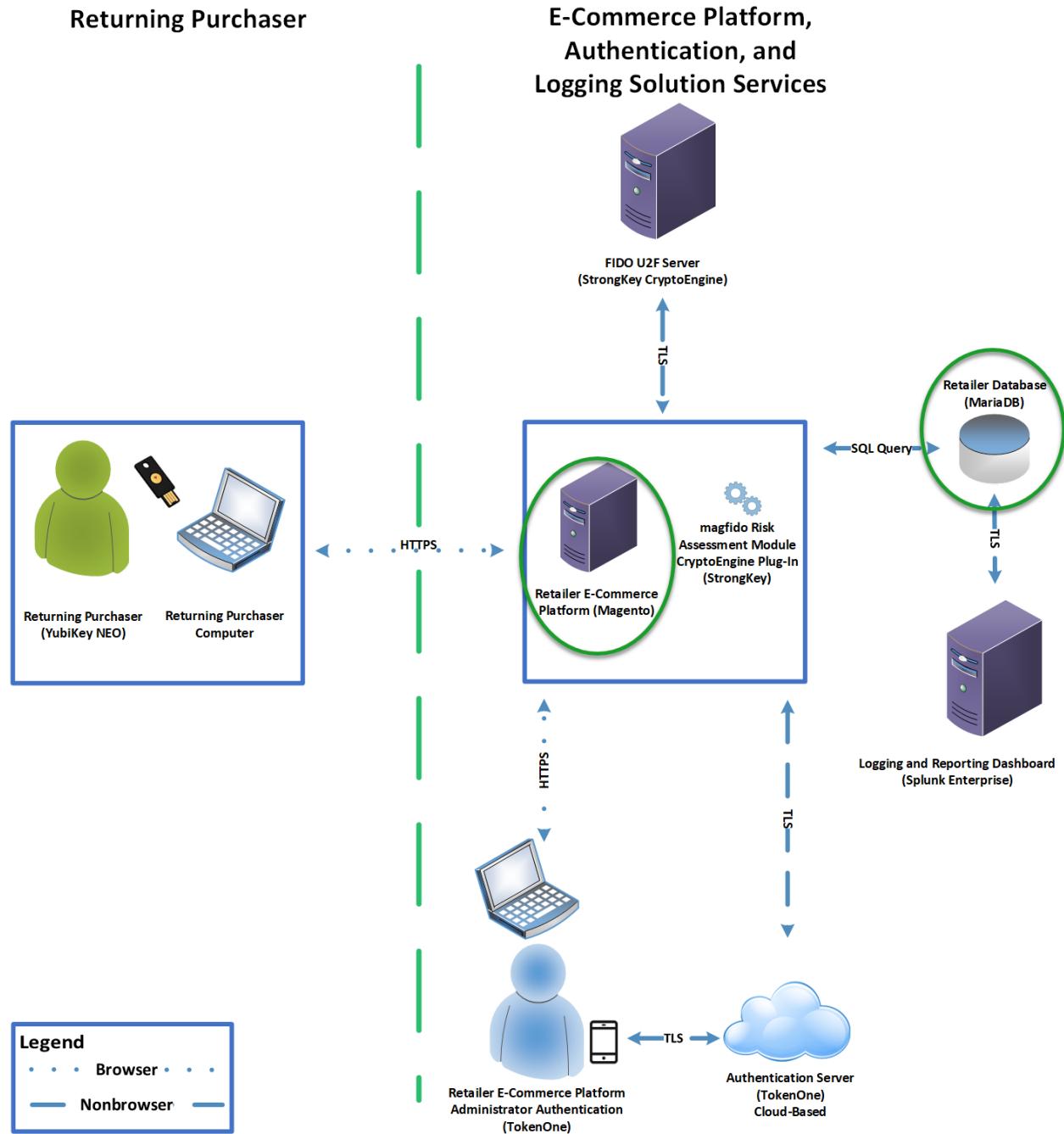
360 11. Test the FIDO Engine. Follow the testing instructions under Step D at the following URL:
361 <https://sourceforge.net/p/skce/wiki/Test%20SKCE%202.0%20Using%20a%20Client%20Program%20%28Build%20163%29/>.

363 There are additional tests on that web page to test the other cryptographic functions of the
364 SKCE; however, only the FIDO Engine tests are critical for this build.

365 **2.2 Magento Open Source Electronic Commerce Platform**

366 This section provides installation and configuration guidance for the Magento Open Source e-commerce
367 platform. The Magento platform provides connectivity to most of the example implementations'
368 components. Both example implementation builds use Magento. The location of the Magento
369 components that are installed using the instructions in this section are illustrated in [Figure 2-2](#) (circled in
370 green).

371 Figure 2-2 Magento Open Source E-Commerce Platform Components



372

373 **2.2.1 Magento Overview**

374 Magento is an e-commerce platform that offers on-premises and cloud solutions to retailers. For this lab
375 implementation, we leveraged the Magento Open Source version of this platform, which was hosted on-
376 premises. This section describes how to install and configure Magento Open Source [4], [5] and how to
377 configure it with StrongKey's SKCE FIDO U2F server capabilities. For the e-commerce platform, Magento
378 Open Source Version 2.1.8 was used in the example implementation.

379 The installation procedure consists of the following steps:

- 380 □ Download the Magento software to the server where it will be installed.
381 □ Download the software dependencies to the server where Magento will be installed.
382 □ Execute commands as root/administrator.
383 □ Perform post-installation configuration.

384 **2.2.2 Magento Requirements**

385 The following subsections document the software, hardware, and network requirements for Magento
386 Open Source 2.1.X.

387 ***2.2.2.1 Software Requirements***

388 For this implementation, Magento was installed on a CentOS 7.0 server.

389 Magento Open Source developer's documentation states that Magento can operate on Linux operating
390 systems, such as these:

- 391 □ RedHat Enterprise Linux
392 □ CentOS
393 □ Ubuntu
394 □ Debian

395 Magento Open Source 2.1.X requires the following installations:

- 396 □ Web Server: Apache 2.2 or 2.4, or nginx 1.X
397 □ Database: MySQL 5.6, MariaDB, Percona, or other binary-compatible MySQL technologies
398 □ Hypertext Preprocessor (PHP): 7.0.2, 7.0.4, 7.0.6-7.0.X, or 7.1.X
399 □ Secure Socket Layer (SSL)
400 □ Mail Server: Redis 3.0, Varnish 3.5, memcached

See Magento's developer's documentation for additional details and download links:
<https://devdocs.magento.com/guides/v2.1/install-gde/system-requirements-tech.html>.

401

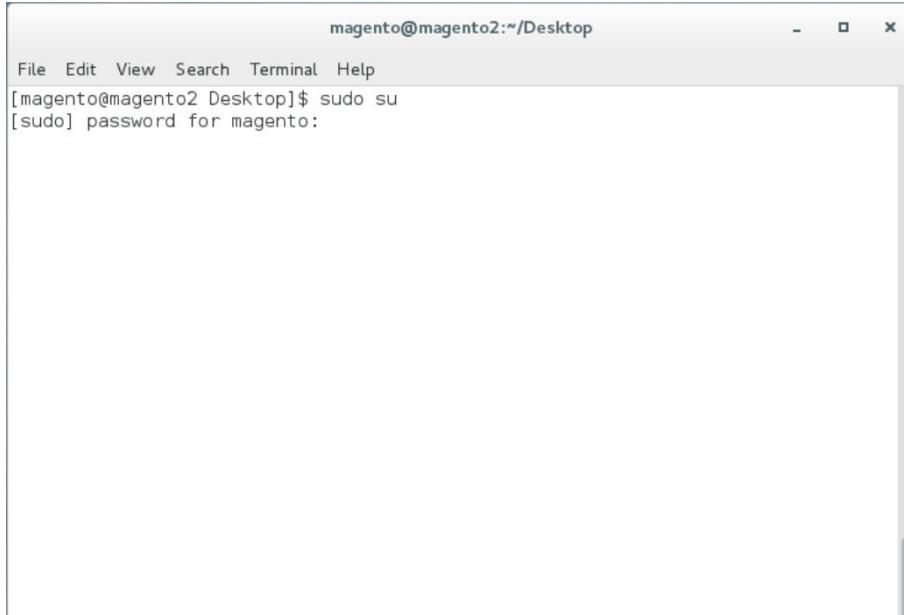
402 *2.2.2.2 Hardware Requirements*

403 Magento requires installing Magento Open Source on a server with at least 2 GB of RAM.

404 **2.2.3 Magento Preinstallation**405 Magento requires the Linux, Apache, MySQL, PHP (LAMP) software stack. This section describes the
406 process of installing and configuring the software stack that uses versions compatible with Magento.

407 1. Open a terminal window, and enter the following command to log in as root:

408 sudo su

409 a. After entering the command, you will be prompted to enter the password for the cur-
410 rent user.A screenshot of a terminal window titled "magento@magento2:~/Desktop". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the command "[magento@magento2 Desktop]\$ sudo su" being typed. A password prompt "[sudo] password for magento:" is visible, indicating the user is about to enter their root password to switch to root user mode.

411

412 2. To install wget from the terminal, enter the following command:

413 yum install wget

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# yum install wget
```

414

415 3. Download the Extra Packages for Enterprise Linux repository by entering the following command:

417 wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

418

419 4. Download the Remi repository by entering the following command:

420 wget http://rpms.remirepo.net/enterprise/remi-release-7.rpm

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# wget https://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

421

422 5. Add the two repositories—so that YUM can locate them when needed—by entering the following command:

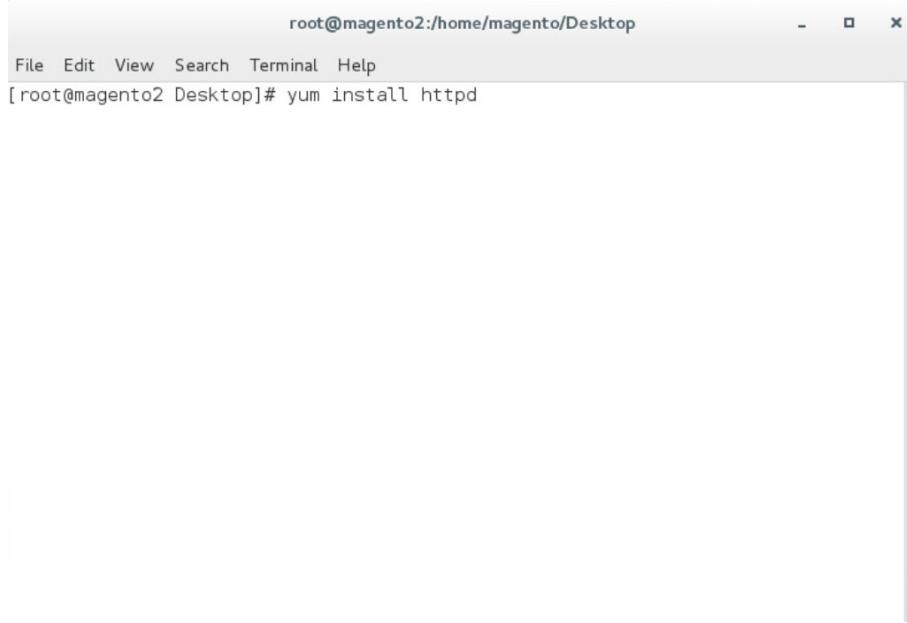
424 rpm -Uvh remi-release-7.rpm epel-release-latest-7.noarch.rpm

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# rpm -Uvh remi-release-7.rpm epel-release-latest-7.noarch.rpm
```

425

426 6. Install the Apache server by entering the following command:

427 yum install httpd

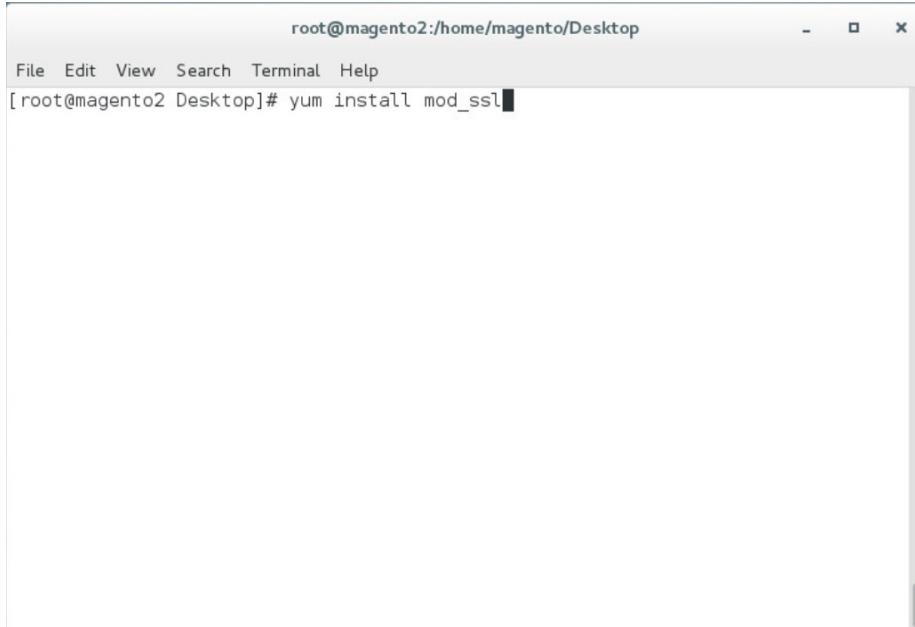


A screenshot of a terminal window titled "root@magento2:/home/magento/Desktop". The window has a standard Linux-style interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar. The main area of the terminal shows the command "[root@magento2 Desktop]# yum install httpd" entered and ready to be executed.

428

429 7. Install Transport Layer Security (TLS)/SSL support for Hypertext Transfer Protocol Daemon
430 (HTTPD) by entering the following command:

431 yum install mod_ssl

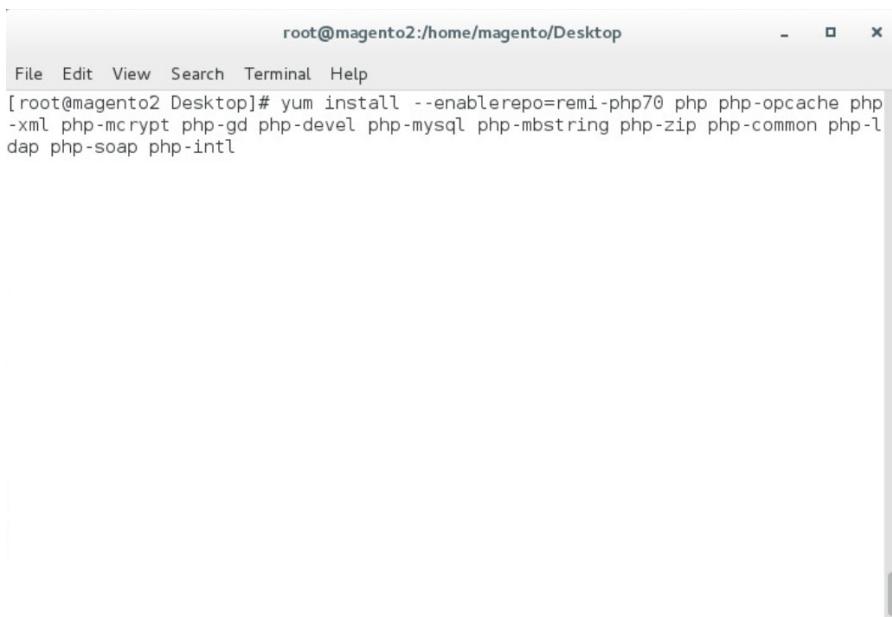


A screenshot of a terminal window titled 'root@magento2:/home/magento/Desktop'. The window has a standard title bar with icons for minimize, maximize, and close. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main area shows the command '[root@magento2 Desktop]# yum install mod_ssl' being typed into the terminal.

432

433 8. Install PHP by entering the following command:

434 yum install --enablerepo=remi-php70 php php-opcache php-xml php-mcrypt php-gd
435 php-devel php-mysql php-mbstring php-zip php-common php-ldap php-soap php-intl



A screenshot of a terminal window titled 'root@magento2:/home/magento/Desktop'. The window has a standard title bar with icons for minimize, maximize, and close. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main area shows the command '[root@magento2 Desktop]# yum install --enablerepo=remi-php70 php php-opcache php-xml php-mcrypt php-gd php-devel php-mysql php-mbstring php-zip php-common php-ldap php-soap php-intl' being typed into the terminal.

436

437 9. Create a file named *Maria.repo* in the */etc/yum.repos.d* by entering the following command:

438 vim /etc/yum.repos.d/Maria.repo

```
root@magento2:/home/magento/Desktop - □ ×
File Edit View Search Terminal Help
[root@magento2 Desktop]# vim /etc/yum.repos.d/Maria.repo
```

439

440 10. In the text editor, enter the following contents:

```
441 [mariadb]
442 name = MariaDB
443 baseurl = http://yum.mariadb.org/10.2/centos7-amd64
444 gpgkey = https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
445 gpgcheck = 1
```

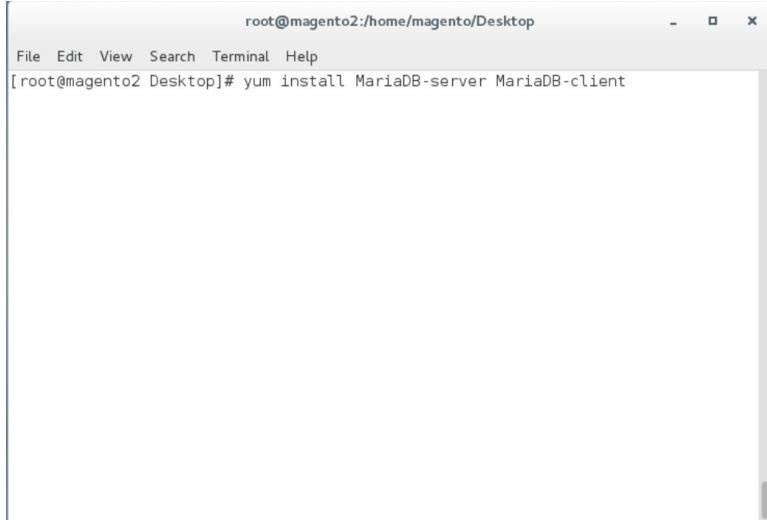
446

447 11. Save the file, and exit by entering the following command:

448 :wq!

449 12. Install MariaDB by entering the following command:

450 yum install MariaDB-server MariaDB-client

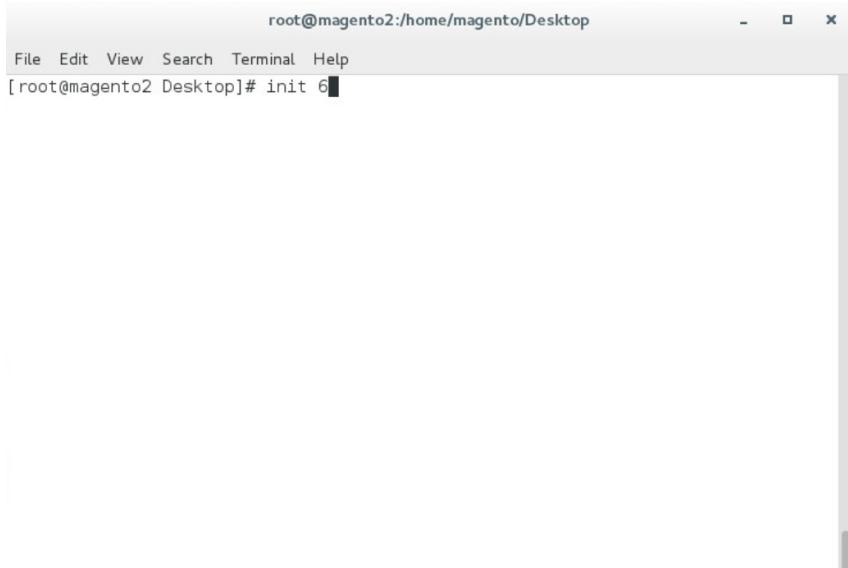


A screenshot of a terminal window titled "root@magento2:/home/magento/Desktop". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command line shows the root user at the prompt: [root@magento2 Desktop]# yum install MariaDB-server MariaDB-client. The terminal is mostly empty below the command line.

451

452 13. Restart the computer system by entering the following command:

453 init 6

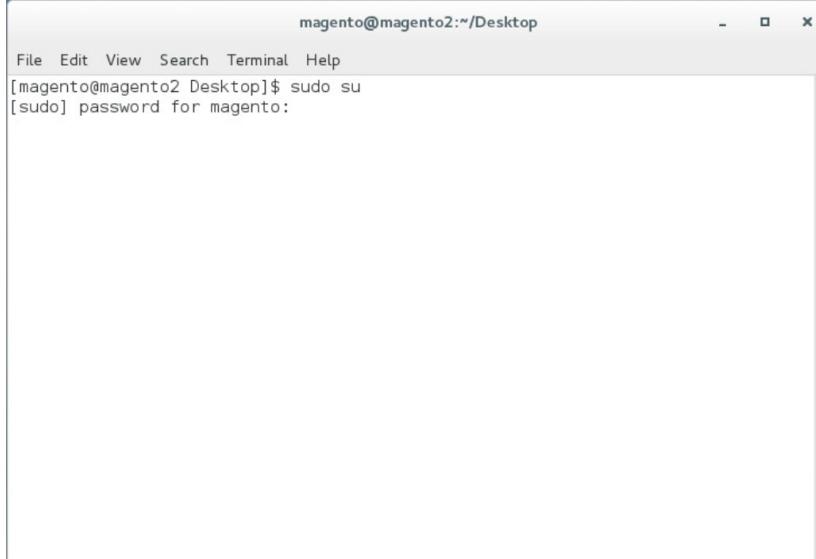


A screenshot of a terminal window titled "root@magento2:/home/magento/Desktop". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command line shows the root user at the prompt: [root@magento2 Desktop]# init 6. The terminal is mostly empty below the command line.

454

455 14. Open a terminal window, and enter the following command to log in as root:

456 sudo su

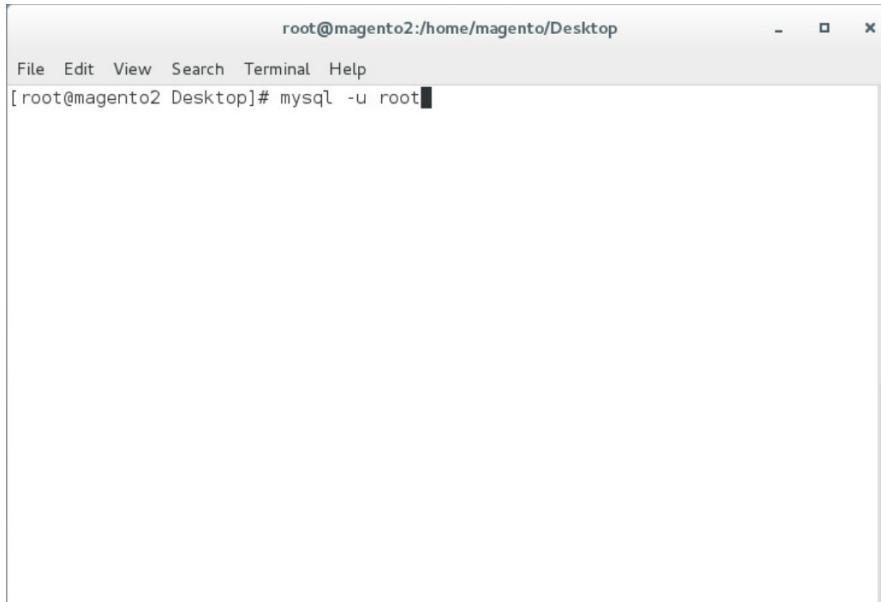


A screenshot of a terminal window titled "magento@magento2:~/Desktop". The window has a standard title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the command "[magento@magento2 Desktop]\$ sudo su" followed by the message "[sudo] password for magento:". The terminal is currently empty, awaiting a password entry.

457

458 15. Log into MariaDB as root by entering the following command (Note: Even though the MariaDB
459 relational database is being used, it uses the same tools as the MySQL database.):

460 mysql -u root

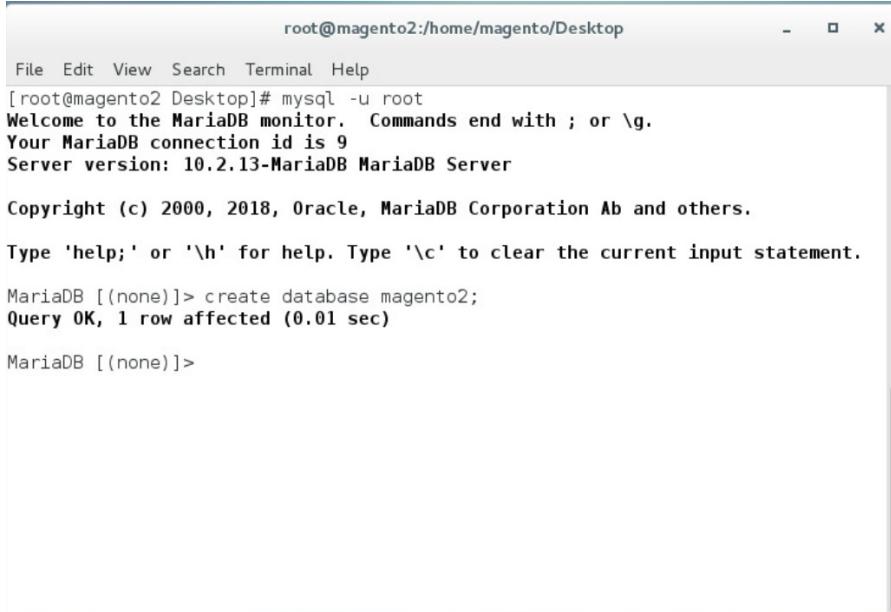


A screenshot of a terminal window titled "root@magento2:/home/magento/Desktop". The window has a standard title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the command "[root@magento2 Desktop]# mysql -u root" with the cursor at the end of the command line.

461

462 16. Create the Magento database by entering the following SQL command:

463 create database magento2;



The screenshot shows a terminal window titled "root@magento2:/home/magento/Desktop". The window contains the following text:

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.2.13-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database magento2;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]>
```

464

465 17. Create the Magento user by entering the following command, replacing parameters in <> with
466 values appropriate for your installation:

467 GRANT ALL PRIVILEGES ON magento2.* TO magento@localhost IDENTIFIED BY '<db
468 password>';

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.2.13-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> GRANT ALL PRIVILEGES ON magento2.* TO magento@localhost IDENTI
FIED BY '*****';
```

469

470 18. Flush the database privileges by entering the following SQL command:

471 flush privileges;

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.2.13-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

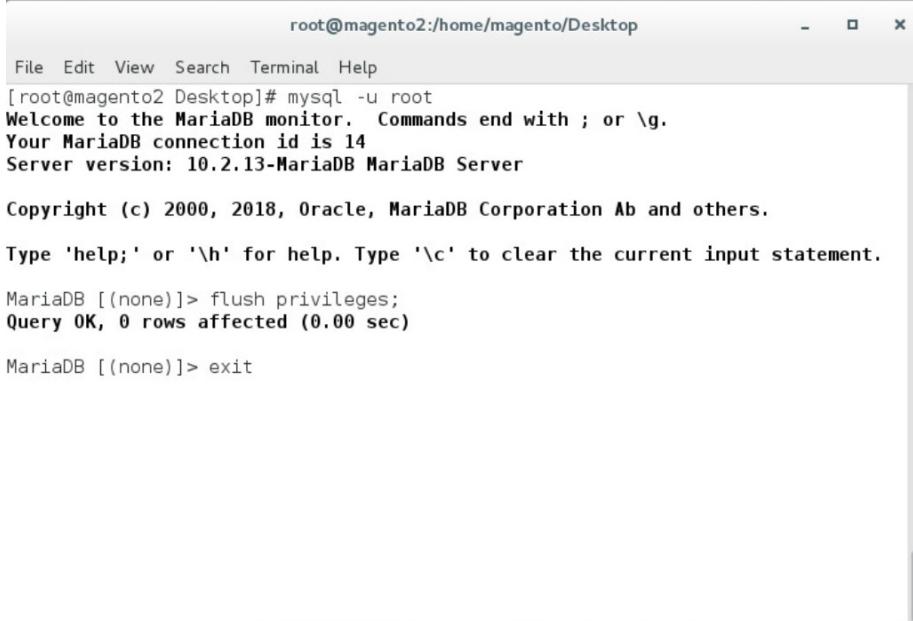
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> flush privileges;
```

472

473 19. Exit the MariaDB shell by entering the following command:

474 exit



A screenshot of a terminal window titled "root@magento2:/home/magento/Desktop". The window shows a MySQL session:

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.2.13-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

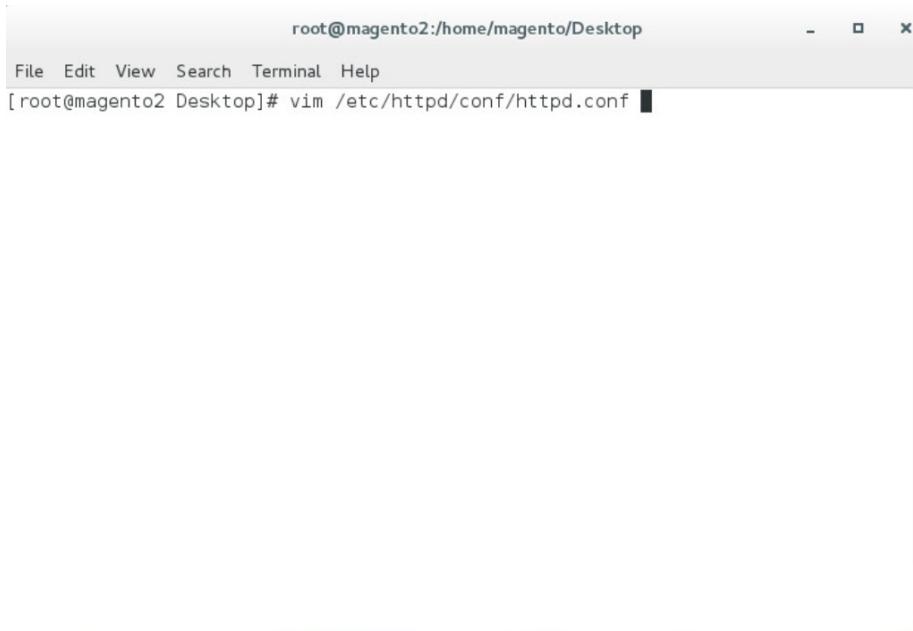
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
```

475

476 20. Open *httpd.conf* to modify Apache settings by entering the following command:

477 vim /etc/httpd/conf/httpd.conf



A screenshot of a terminal window titled "root@magento2:/home/magento/Desktop". The window shows the command "vim /etc/httpd/conf/httpd.conf" being entered:

```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]# vim /etc/httpd/conf/httpd.conf
```

478

479 21. Locate the <Directory "/var/www/html"> section, and change "AllowOverride None" to
480 "AllowOverride All".

```

root@magento2:/home/mag
File Edit View Search Terminal Help
<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>
# Further relax access to the default document root:
<Directory "/var/www/html">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksIfOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important. Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks
    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   Options FileInfo AuthConfig Limit
    #
    AllowOverride All
    #
    # Controls who can get stuff from this server.
    #
    Require all granted
</Directory>

```

481

482 22. Save, and exit by entering the following command:

483 :wq!

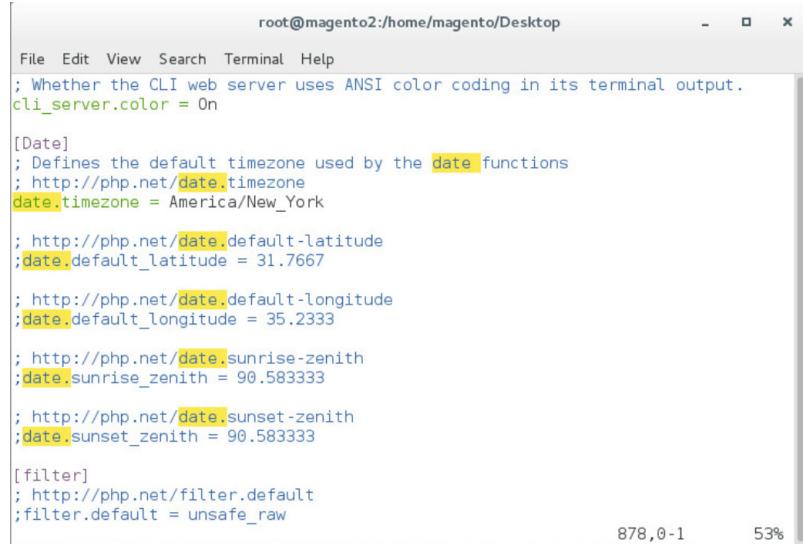
484 23. Open *php.ini* to modify PHP settings by entering the following command:

485 vim /etc/php.ini

The screenshot shows a terminal window titled 'root@magento2:/home/magento/Desktop'. The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The terminal itself displays the command '[root@magento2 Desktop]# vim /etc/php.ini' in black text on a white background.

486

487 24. Uncomment the line containing `date.timezone` by removing the ";" character preceding the
 488 text, and enter your time zone as shown below (this example is for the eastern United States).
 489 `date.timezone = America/New_York`



```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
; Whether the CLI web server uses ANSI color coding in its terminal output.
cli_server.color = On

[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = America/New_York

; http://php.net/date.default-latitude
;date.default_latitude = 31.7667

; http://php.net/date.default-longitude
;date.default_longitude = 35.2333

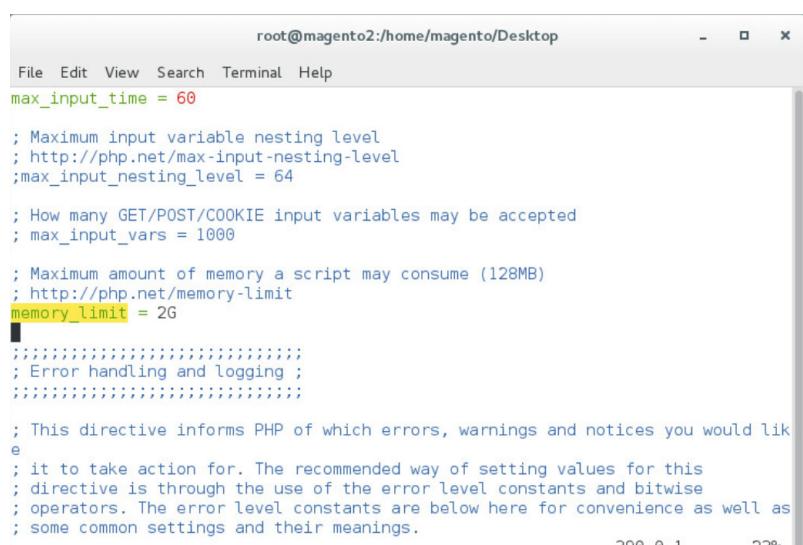
; http://php.net/date.sunrise-zenith
;date.sunrise_zenith = 90.583333

; http://php.net/date.sunset-zenith
;date.sunset_zenith = 90.583333

[filter]
; http://php.net/filter.default
;filter.default = unsafe_raw
```

490
 491 25. Uncomment the line containing `memory_limit` by removing the ";" character preceding the text,
 492 and enter 2G as the value, as shown below.

493 `memory_limit = 2G`



```
root@magento2:/home/magento/Desktop
File Edit View Search Terminal Help
max_input_time = 60

; Maximum input variable nesting level
; http://php.net/max-input-nesting-level
;max_input_nesting_level = 64

; How many GET/POST/COOKIE input variables may be accepted
; max_input_vars = 1000

; Maximum amount of memory a script may consume (128MB)
; http://php.net/memory-limit
memory_limit = 2G

;;;;;;
; Error handling and logging ;
;;;;;;

; This directive informs PHP of which errors, warnings and notices you would like
; it to take action for. The recommended way of setting values for this
; directive is through the use of the error level constants and bitwise
; operators. The error level constants are below here for convenience as well as
; some common settings and their meanings.
```

494
 495 26. Open `10-opcache.ini` to modify PHP settings by entering the following command:
 496 `vim /etc/php.d/10-opcache.ini`

```
root@magento2:~  
File Edit View Search Terminal Help  
[root@magento2 ~]# vim /etc/php.d/10-opcache.ini
```

497

498 27. Uncomment the line containing `opcache.save_comments` by removing the ";" character preceding the text. The line should then read as shown below.

499
500 `opcache.save_comments=1`

```
root@magento2:/home/magento/Desktop  
File Edit View Search Terminal Help  
  
; How often (in seconds) to check file timestamps for changes to the shared  
; memory storage allocation. ("1" means validate once per second, but only  
; once per request. "0" means always validate)  
;opcache.revalidate_freq=2  
  
; Enables or disables file search in include_path optimization  
;opcache.revalidate_path=0  
  
; If disabled, all PHPDoc comments are dropped from the code to reduce the  
; size of the optimized code.  
;opcache.save_comments=1  
  
; If enabled, a fast shutdown sequence is used for the accelerated code  
;opcache.fast_shutdown=0  
  
; Allow file existence override (file_exists, etc.) performance feature.  
;opcache.enable_file_override=0  
  
; A bitmask, where each bit enables or disables the appropriate OPcache  
; passes  
;opcache.optimization_level=0xffffffff
```

501

44,0-1 31%

502 **2.2.4 Magento Installation**

503 For the e-commerce platform, Magento Open Source Version 2.1.8 [5] was used in the example
504 implementation.

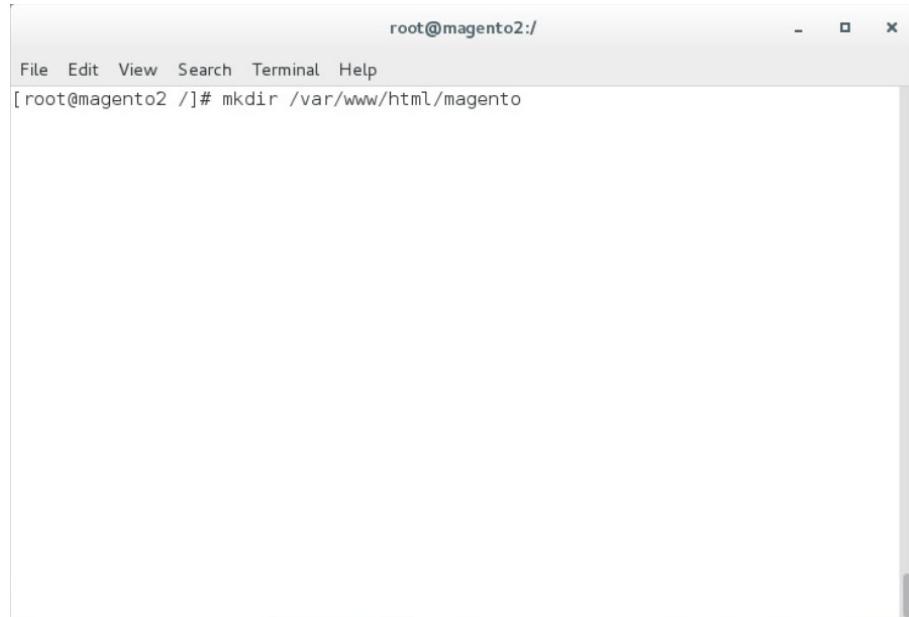
To download the open-source copy of Magento, navigate to the site:
<https://magento.com/products/open-source>.

505
506 When redirected to the resource page, specify the download format. In the example implementation,
507 we installed Magento on CentOS by selecting a file that ends in .tar.gz, as shown in the example below.

508 Magento-Community-Edition-2.1.8.tar.gz

509 1. Create a Magento directory inside HTTPD's DocumentRoot folder by entering the following com-
510 mand:

511 mkdir /var/www/html/magento



The screenshot shows a terminal window with a light gray background and a dark gray title bar. The title bar displays 'root@magento2:' followed by standard window control buttons. The main window area shows the command 'mkdir /var/www/html/magento' being typed at the root prompt. The terminal window has a vertical scroll bar on the right side.

512
513 2. Move the *Magento-CE-2.1.8.tar.gz* into the Magento directory with the following command:

514 mv <download location>/Magento-CE-2.1.8-2017-08-09-96-91-21.tar.gz
515 /var/www/html/magento

```
root@magento2:/  
File Edit View Search Terminal Help  
[root@magento2 /]# mv /home/magento/Downloads/Magento-CE-2.1.8-2017-08-09-06-01-  
21.tar.gz var/www/html/magento/
```

516

517 3. Change the directory to the Magento directory by entering the following command (all com-
518 mands following this step should be run from this directory):

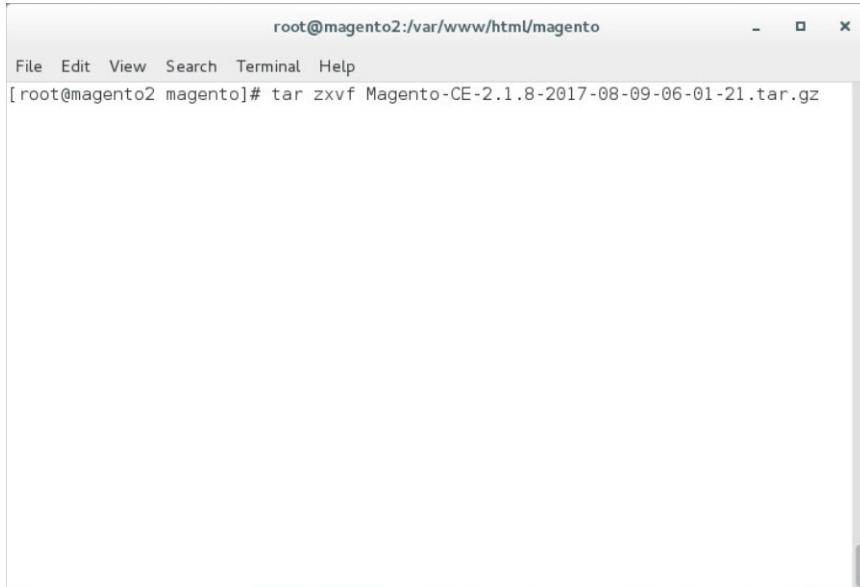
519 cd /var/www/html/magento

```
root@magento2:/  
File Edit View Search Terminal Help  
[root@magento2 /]# cd var/www/html/magento/
```

520

521 4. Extract the Magento distribution from *Magento-CE-2.1.8.tar.gz* by entering the following com-
522 mand:

523 tar zxvf Magento-CE-2.1.8-2017-08-09-96-91-21.tar.gz

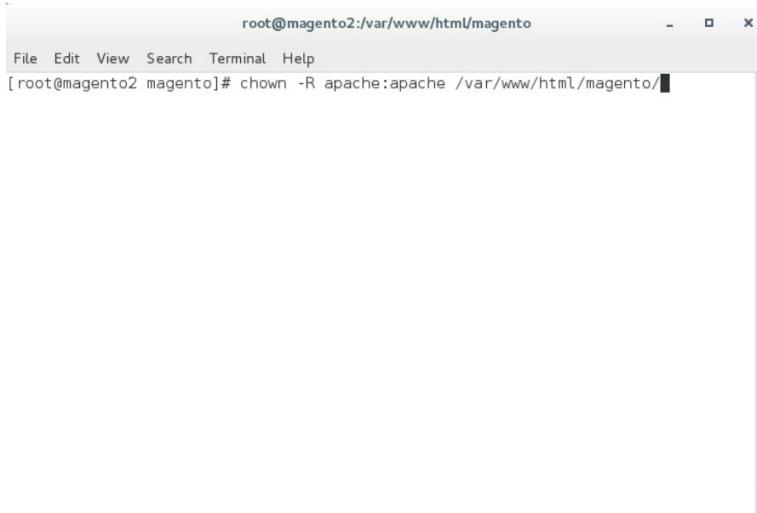


A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The terminal menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command entered is "[root@magento2 magento]# tar zxvf Magento-CE-2.1.8-2017-08-09-06-01-21.tar.gz". The terminal is currently empty of output.

524

525 5. Change ownership of the extracted files to the Apache user by entering the following command:

526 chown -R apache:apache /var/www/html/magento



A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The terminal menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command entered is "[root@magento2 magento]# chown -R apache:apache /var/www/html/magento/". The terminal is currently empty of output.

527

528 6. Change file permissions by entering the following command (Note: This is a single command
529 that must be executed on a single line.):

530 find var vendor pub/static pub/media app/etc -type f -exec chmod u+w {} \; &&
531 find var vendor pub/static pub/media app/etc -type d -exec chmod u+w {} \; &&
532 chmod u+x bin/magento

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# find var vendor pub/static pub/media app/etc -type f -e
xec chmod u+w {} \; && find var vendor pub/static pub/media app/etc -type d -exe
c chmod u+w {} \; && chmod u+x bin/magento
```

533

534 7. Change the Security-Enhanced Linux (SELinux) context permissions to allow the Apache user to
535 have read/write access to specific directories within the Magento directory, by entering the fol-
536 lowing command:

537 chcon -R --type httpd_sys_rw_content_t app/etc var pub/media pub/static

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# chcon -R --type httpd_sys_rw_content_t app/etc var pub/
media pub/static
```

538

539 8. Open the web browser to log into <https://marketplace.magento.com> and access your account.
540 Click **Access Keys**.

The screenshot shows the Magento Marketplace dashboard. At the top, there are links for Extensions, Themes, and Partners. On the right, there is a shopping cart icon and the user's name, Blaine Mulugeta. Below the header is a search bar. The main navigation bar has tabs for Marketplace, Magento, and Developer Portal. Under the Marketplace tab, there are sections for My Products, Payment, and My Information. The My Products section includes links for Access Keys, Purchase History, My Purchases, and Refunded Orders.

541

- 542 9. In the Magento tab, click **Create A New Access Key**.

The screenshot shows a modal dialog titled "Create new access keys". Inside the dialog, there is an input field containing "MFANccoe". Below the input field, there is an error message: "No white space please". At the bottom of the dialog are two buttons: "Cancel" and "OK".

543

- 544 10. Enter a name for your new access key, and click **OK**.

The screenshot shows the same modal dialog as before, but now it displays a success message: "Changes were saved." above the input field. The input field still contains "MFANccoe" and the error message "No white space please" is still present. The "OK" button is highlighted.

545

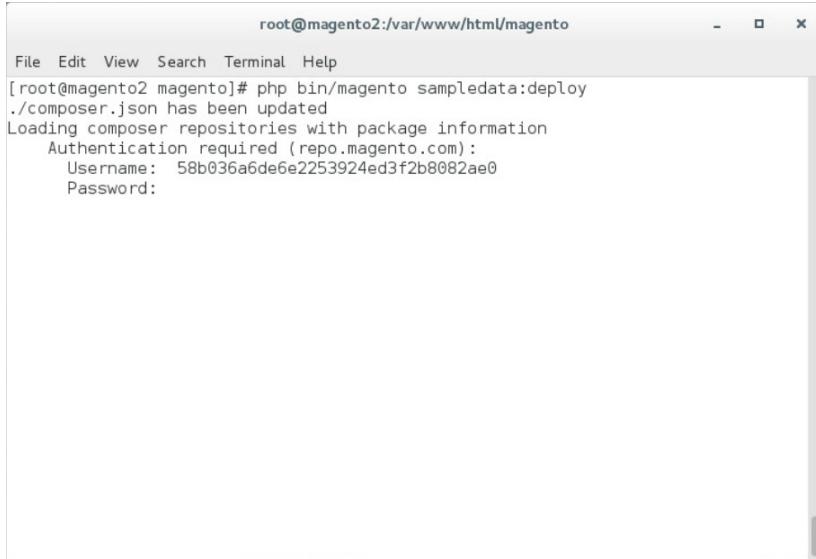
- 546 11. The new access keys will be displayed in the menu with the **Status of Enabled**.

The screenshot shows the "Access Keys" table in the Magento 2 dashboard. The table has columns for Name, Access Keys, Status, and Actions. There is one row in the table with the following data:

Name	Access Keys	Status	Actions
MFANccoe	Public Key: 58b036a6de6e2253924ed3f2b8082ae0 Private Key: bbf557e31e3049c19a0f696049f3ab55	Enabled	Delete

547

548 12. Install Magento's sample data by entering the following command and then providing <public
549 key> when a **Username** is requested and <private key> as the **Password** when prompted:
550 php bin/magento sampledata:deploy



```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[ root@magento2 magento]# php bin/magento sampledata:deploy
./composer.json has been updated
Loading composer repositories with package information
Authentication required (repo.magento.com):
Username: 58b036a6de6e2253924ed3f2b8082ae0
Password:
```

551
552 13. Install the Magento software distribution by issuing the following command, replacing parameters in <> with values appropriate for your installation (Note: This is a single command that must
553 be executed on a single line.):
554

555 php bin/magento setup:install --admin-firstname=<First Name> --admin-
556 lastname=<Last Name> --admin-email=<email> --admin-user=strongauth --admin-
557 password=<password> --baseurl=https://<fully-qualified-domainname>/magento/ --
558 db-host=127.0.01 --db-name=magento2 --db-user=magento --db-password=<db-
559 password> --use-secure-admin=1

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# php bin/magento setup:install --admin-firstname=admin --admin-lastname=admin --admin-email=admin@example.com --admin-user=admin --admin-password=Password1! --base-url=https://magento2.mfa.local/magento/ --db-host=127.0.0.1 --db-name=magento2 --db-user=magento --db-password=Password1! --use-secure-admin=1
```

560

561 14. Modify compiled file permissions by issuing the following command:

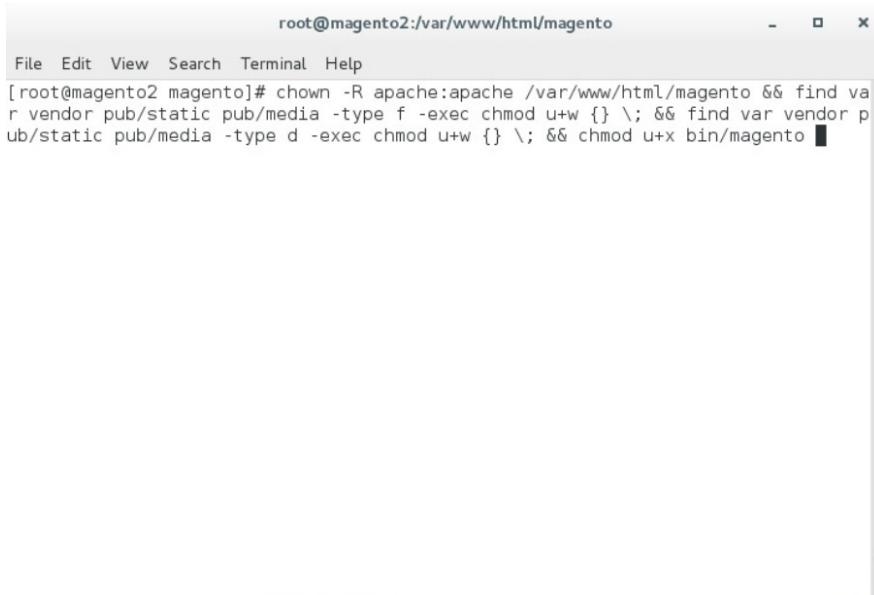
562 chmod -R u-w app/etc

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# chmod -R u-w app/etc
```

563

564 15. Modify compiled file permissions by issuing the following command:

565 chown -R apache:apache /var/www/html/magento && find var vendor pub/static
566 pub/media -type f -exec chmod u+w {} \; && find var vendor pub/static pub/media
567 -type d -exec chmod u+w {} \; && chmod u+x bin/magento



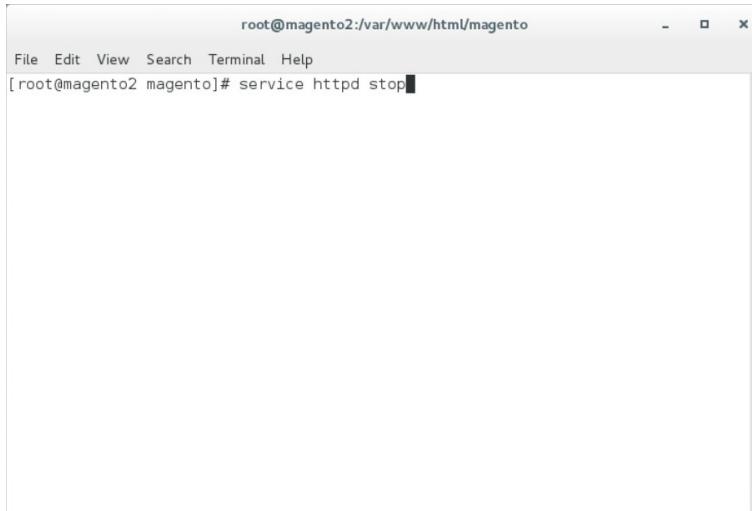
A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The terminal content shows a command being run in root shell:

```
root@magento2:/var/www/html/magento
[root@magento2 magento]# chown -R apache:apache /var/www/html/magento && find va
r vendor pub/static pub/media -type f -exec chmod u+w {} \; && find var vendor p
ub/static pub/media -type d -exec chmod u+w {} \; && chmod u+x bin/magento
```

568

569 16. Modify SELinux permissions to enable HTTPD to access the database, by executing the following
570 commands:

571 a. service httpd stop



A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The terminal content shows a command being run in root shell:

```
root@magento2:/var/www/html/magento
[root@magento2 magento]# service httpd stop
```

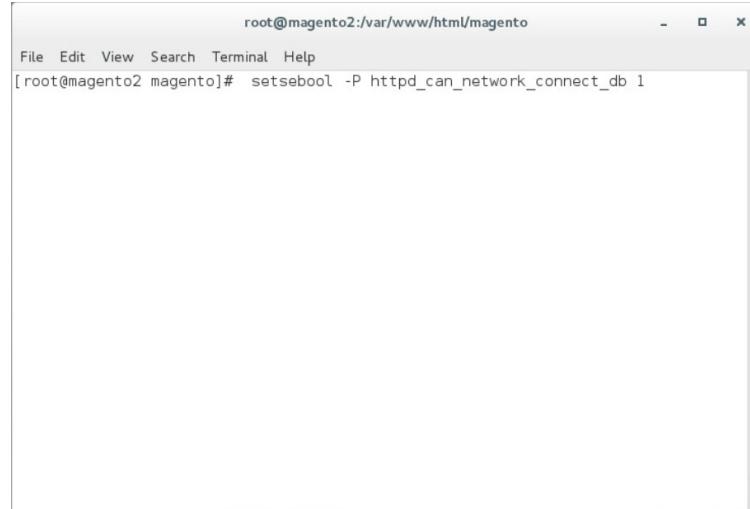
572

573 b. setsebool -P httpd_can_network_connect 1

574

575

c. setsebool -P httpd_can_network_connect_db 1

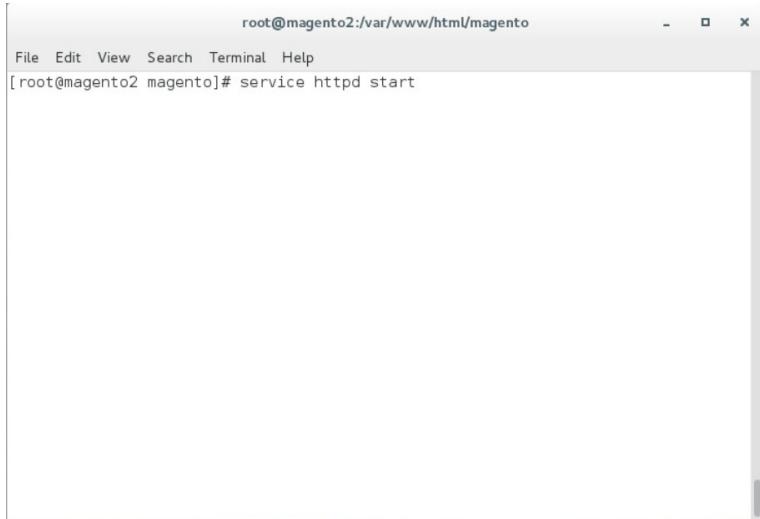


```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# setsebool -P httpd_can_network_connect_db 1
```

576

577

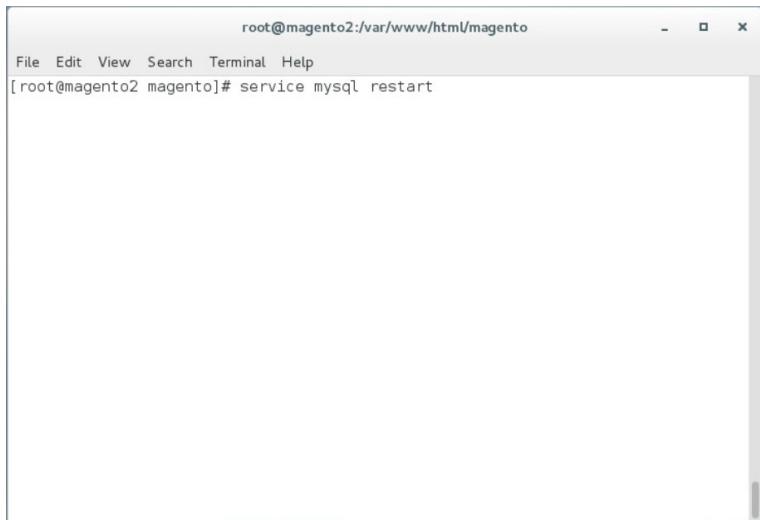
d. service httpd start



A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The main area of the terminal shows the command "[root@magento2 magento]# service httpd start" entered by the user.

578

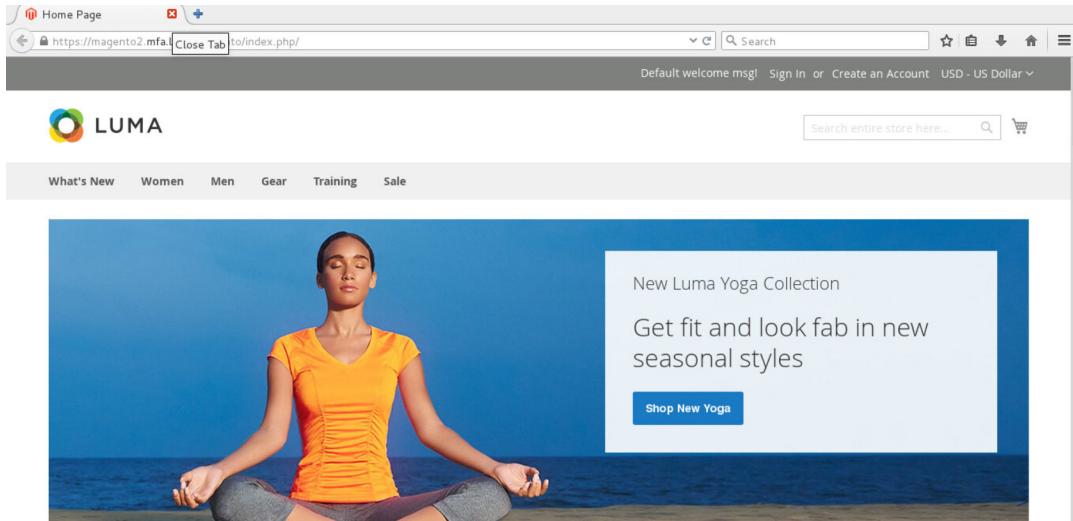
579 e. service mysql restart



A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The main area of the terminal shows the command "[root@magento2 magento]# service mysql restart" entered by the user.

580

581 17. Verify the installation by navigating in the browser to the store URL, which was set up in
582 Section 2.2.4, Step 13 (<https://magento2.mfa.local/magento>).



583

584 2.2.5 Configuring the Magento Account Lockout Feature

585 This section describes the steps required to configure account lockouts after a specified number of failed
586 login attempts. For our example implementation, we specified five as the maximum number of
587 login-attempt failures before temporarily disabling the account, and 20 minutes as the lockout time.
588 These parameters can be adjusted, and the administrator of the Magento site has the information
589 system privileges to set these values based on the implementer's preference.

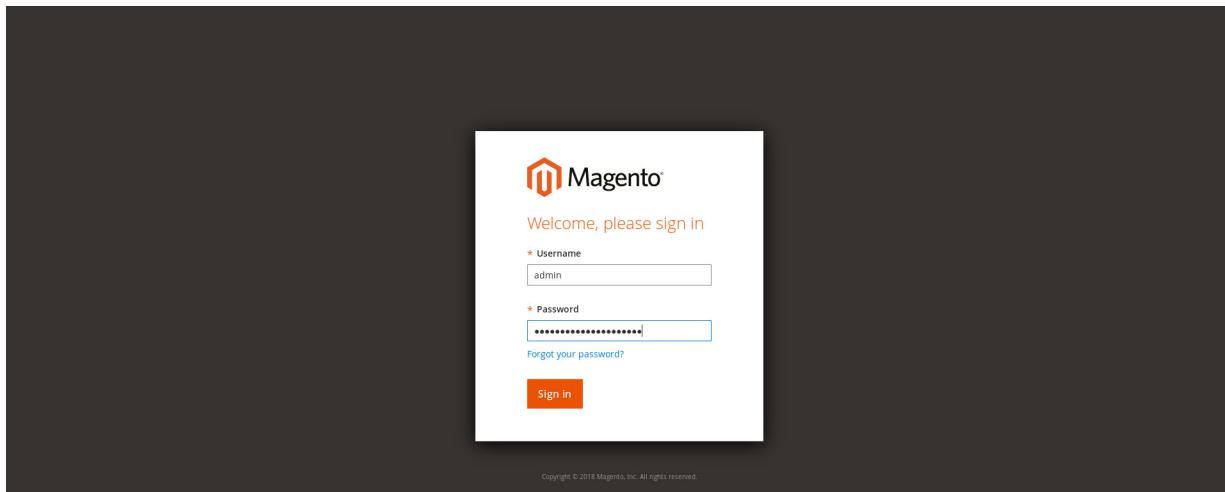
590 1. Determine the admin Uniform Resource Identifier (URI) by running the following command:

```
591     php bin/magento info:adminuri
```

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[ root@magento2 magento]# php bin/magento info:adminuri
Admin URI: /admin_14mzl4
[ root@magento2 magento]#
```

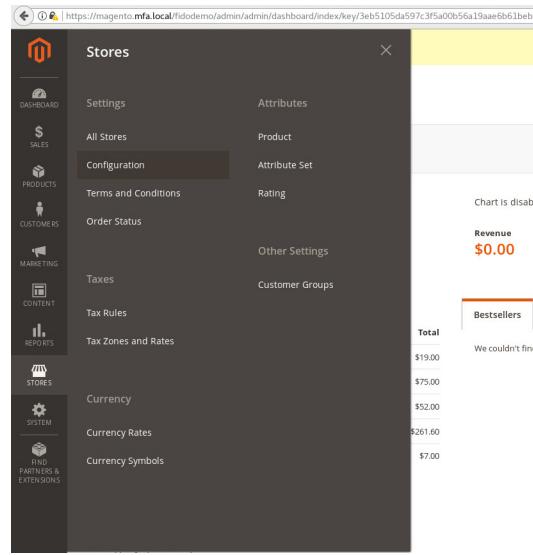
592

- 593 2. Navigate to the admin URI identified in [Section 2.2.5](#), Step 1, and sign in with the Magento
594 **Username** and **Password** created in [Section 2.2.4](#), Step 13 (the example implementation URI is
595 https://magento2.mfa.local/admin_14mzl4).



596

- 597 3. Proceed to the Configuration page: **STORES > Configuration**.



598

- 599 4. Click the **CUSTOMERS** drop-down from the menu in the **Configuration** page, and select **Customer Configuration**.
- 600

Configuration

A screenshot of the Configuration page. At the top, there is a 'Store View: Default Config' dropdown and a help icon. The main area is a tree view with categories: GENERAL, CATALOG, and CUSTOMERS. Under CUSTOMERS, 'Customer Configuration' is selected, while 'Newsletter', 'Wish List', 'Promotions', and 'Persistent Shopping Cart' are other options.

601

- 602 5. Click the **Password Options** drop-down.

Configuration

   Strongauth ▾

Store View: Default Config 

GENERAL Account Sharing Options 

CATALOG Online Customers Options 

CUSTOMERS Create New Account Options 

Newsletter 

Customer Configuration

Wish List 

Promotions 

Persistent Shopping Cart 

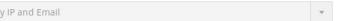
SALES CAPTCHA 

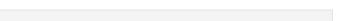
Save Config

603

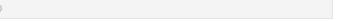
- 604 6. Uncheck the **Use system value** fields for the **Maximum Login Failures to Lockout Account** and
605 **Lockout Time (minutes)** to modify the settings for the **Password Options**.

Password Options

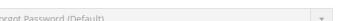
>Password Reset Protection Type  By IP and Email  Use system value

Max Number of Password Reset Requests  5  Use system value

Limit the number of password reset request per hour. Use 0 to disable.

Min Time Between Password Reset Requests  10  Use system value

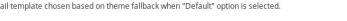
Delay in minutes between password reset requests. Use 0 to disable.

Forgot Email Template  Forgot Password (Default)  Use system value

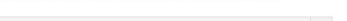
Email template chosen based on theme fallback when "Default" option is selected.

Remind Email Template  Remind Password (Default)  Use system value

Email template chosen based on theme fallback when "Default" option is selected.

Reset Password Template  Reset Password (Default)  Use system value

Email template chosen based on theme fallback when "Default" option is selected.

Password Template Email Sender  Customer Support  Use system value

Recovery Link Expiration Period  2  Use system value

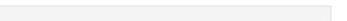
Please enter a number 1 or greater in this field.

Number of Required Character Classes  3  Use system value

Number of different character classes required in password: Lowercase, Uppercase, Digits, Special Characters.

Maximum Login Failures to Lockout Account  5  Use system value

Use 0 to disable account locking.

Minimum Password Length  8  Use system value

Please enter a number 1 or greater in this field.

Lockout Time (minutes)  20  Use system value

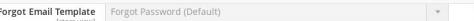
Account will be unlocked after provided time.

606

- 607 7. Click **Save Config** to save the changes made.

Configuration

Save Config

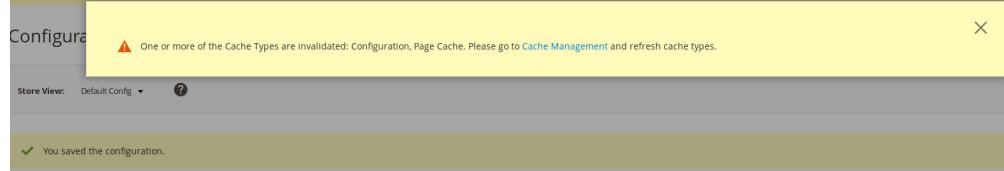
SALES  Forgot Password (Default)  Use system value

SERVICES  Remind Password (Default)  Use system value

ADVANCED

608

- 609 8. The following pop-up will appear, notifying you to refresh Cache Types. Click the **Cache Management** link in the message.



- 611 9. You will be redirected to the **Cache Management** page. Click **Flush Magento Cache** to resolve the **INVALIDATED** Cache Types.

Cache Management

Cache Management				
				Flush Cache Storage
				Flush Magento Cache
Refresh	Submit	13 records found		
Cache Type	Description	Tags	Status	
Configuration	Various XML configurations that were collected across modules and merged	CONFIG	INVALIDATED	
Layouts	Layout building instructions	LAYOUT_GENERAL_CACHE_TAG	ENABLED	
Blocks HTML output	Page blocks HTML	BLOCK_HTML	ENABLED	
Collections Data	Collection data files	COLLECTION_DATA	ENABLED	
Reflection Data	API interfaces reflection data	REFLECTION	ENABLED	
Database DDL operations	Results of DDL queries, such as describing tables or indexes	DB_DDL	ENABLED	
EAV types and attributes	Entity types declaration cache	EAV	ENABLED	
Customer Notification	Customer Notification	CUSTOMER_NOTIFICATION	ENABLED	
Page Cache	Full page caching	FPC	INVALIDATED	
Integrations Configuration	Integration configuration file	INTEGRATION	ENABLED	
Integrations API Configuration	Integrations API configuration file	INTEGRATION_API_CONFIG	ENABLED	
Translations	Translation files	TRANSLATE	ENABLED	
Web Services Configuration	REST and SOAP configurations, generated WSDL file	WEBSERVICE	ENABLED	

- 614 615 10. Upon completion of the flush, the page will reflect the changes.

Cache Management

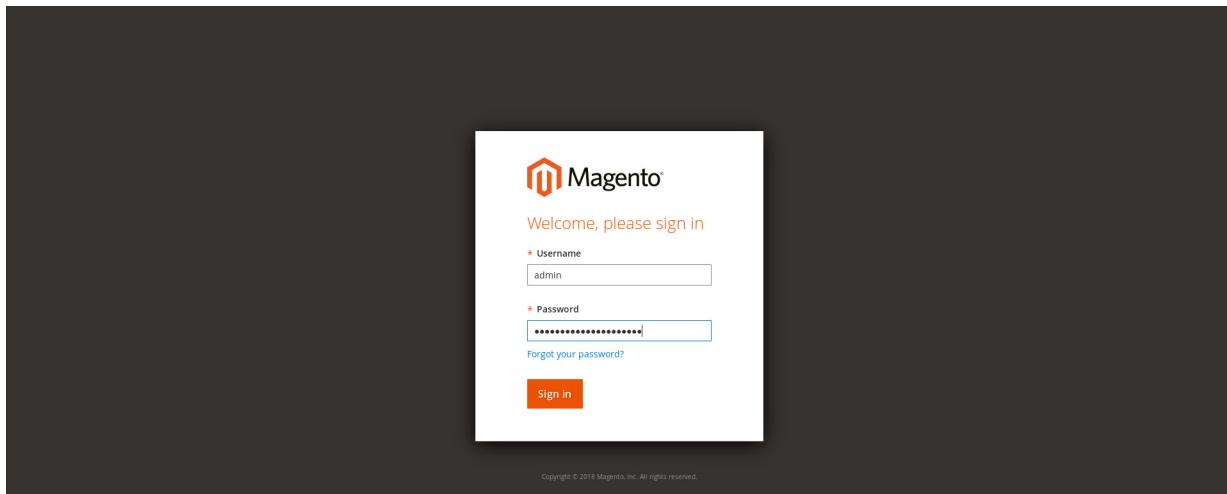
Cache Management				
				Flush Cache Storage
				Flush Magento Cache
Refresh	Submit	13 records found		
Cache Type	Description	Tags	Status	
Configuration	Various XML configurations that were collected across modules and merged	CONFIG	ENABLED	
Layouts	Layout building instructions	LAYOUT_GENERAL_CACHE_TAG	ENABLED	
Blocks HTML output	Page blocks HTML	BLOCK_HTML	ENABLED	
Collections Data	Collection data files	COLLECTION_DATA	ENABLED	
Reflection Data	API interfaces reflection data	REFLECTION	ENABLED	
Database DDL operations	Results of DDL queries, such as describing tables or indexes	DB_DDL	ENABLED	
EAV types and attributes	Entity types declaration cache	EAV	ENABLED	
Customer Notification	Customer Notification	CUSTOMER_NOTIFICATION	ENABLED	
Page Cache	Full page caching	FPC	ENABLED	
Integrations Configuration	Integration configuration file	INTEGRATION	ENABLED	
Integrations API Configuration	Integrations API configuration file	INTEGRATION_API_CONFIG	ENABLED	
Translations	Translation files	TRANSLATE	ENABLED	

616

617 2.2.6 Disabling Magento Guest Checkout

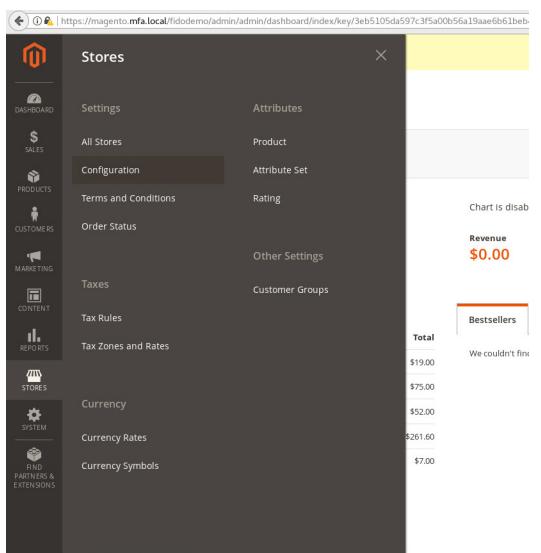
618 This section describes steps to disable Magento's guest checkout feature to ensure that purchasers
619 cannot choose to checkout as a guest.

- 620 1. Navigate to the admin URI identified in [Section 2.2.5](#), Step 1 (https://magento2.mfa.local/admin_14mzl4), and sign in with the **Username** and **Password** created in [Section 2.2.4](#), Step 13.



622

- 623 2. Proceed to the **Configuration** page: **STORES > Configuration**.



624

- 625 3. Click the **SALES** drop-down from the menu on the **Configuration** page, select **Checkout**, and ex-
626 pand the **Checkout Options**.

The screenshot shows the Magento Admin Configuration interface. On the left, there's a sidebar with categories: GENERAL, CATALOG, CUSTOMERS, and SALES. Under SALES, 'Sales', 'Sales Emails', 'PDF Print-outs', and 'Tax' are listed. Below these, 'Checkout' is highlighted with an orange border. The main content area has a header 'Checkout Options'. Underneath are sections for 'Shopping Cart', 'My Cart Link', 'Shopping Cart Sidebar', and 'Payment Failed Emails', each with a small circular icon to the right.

627

- 628 4. Uncheck the **Use system value** fields for the **Allow Guest Checkout** setting, and modify the settings to **No** for the **Checkout Options**.

The screenshot shows the 'Configuration' page with the 'SALES' category selected. Under 'Checkout Options', the 'Allow Guest Checkout' field is set to 'No' and the 'Use system value' checkbox is unchecked. Other fields like 'Enable OnePage Checkout', 'Enable Terms and Conditions', 'Display Billing Address On', and 'Maximum Number of Items to Display in Order Summary' are also shown with their respective values and checkboxes.

630

- 631 5. Click **Save Config**.
- 632 6. The following pop-up will appear, notifying you to refresh Cache Types. Click the **Cache Management** link in the message.



634

- 635 7. You will be redirected to the **Cache Management** page. Click **Flush Magento Cache** to resolve the **INVALIDATED** Cache Types.

Cache Management

   admin ▾

[Flush Cache Storage](#) [Flush Magento Cache](#)
[Refresh](#)[Submit](#)

13 records found

<input type="checkbox"/>	Cache Type	Description	Tags	Status
<input type="checkbox"/>	Configuration	Various XML configurations that were collected across modules and merged	CONFIG	INVALIDATED
<input type="checkbox"/>	Layouts	Layout building instructions	LAYOUT_GENERAL_CACHE_TAG	ENABLED
<input type="checkbox"/>	Blocks HTML output	Page blocks HTML	BLOCK_HTML	ENABLED

637

638 8. Upon completion of the flush, the page will reflect the changes.

Cache Management

   admin ▾

[Flush Cache Storage](#) [Flush Magento Cache](#)
✓ The Magento cache storage has been flushed.
[Refresh](#)[Submit](#)

13 records found

<input type="checkbox"/>	Cache Type	Description	Tags	Status
<input type="checkbox"/>	Configuration	Various XML configurations that were collected across modules and merged	CONFIG	ENABLED
<input type="checkbox"/>	Layouts	Layout building instructions	LAYOUT_GENERAL_CACHE_TAG	ENABLED
<input type="checkbox"/>	Blocks HTML output	Page blocks HTML	BLOCK_HTML	ENABLED

639

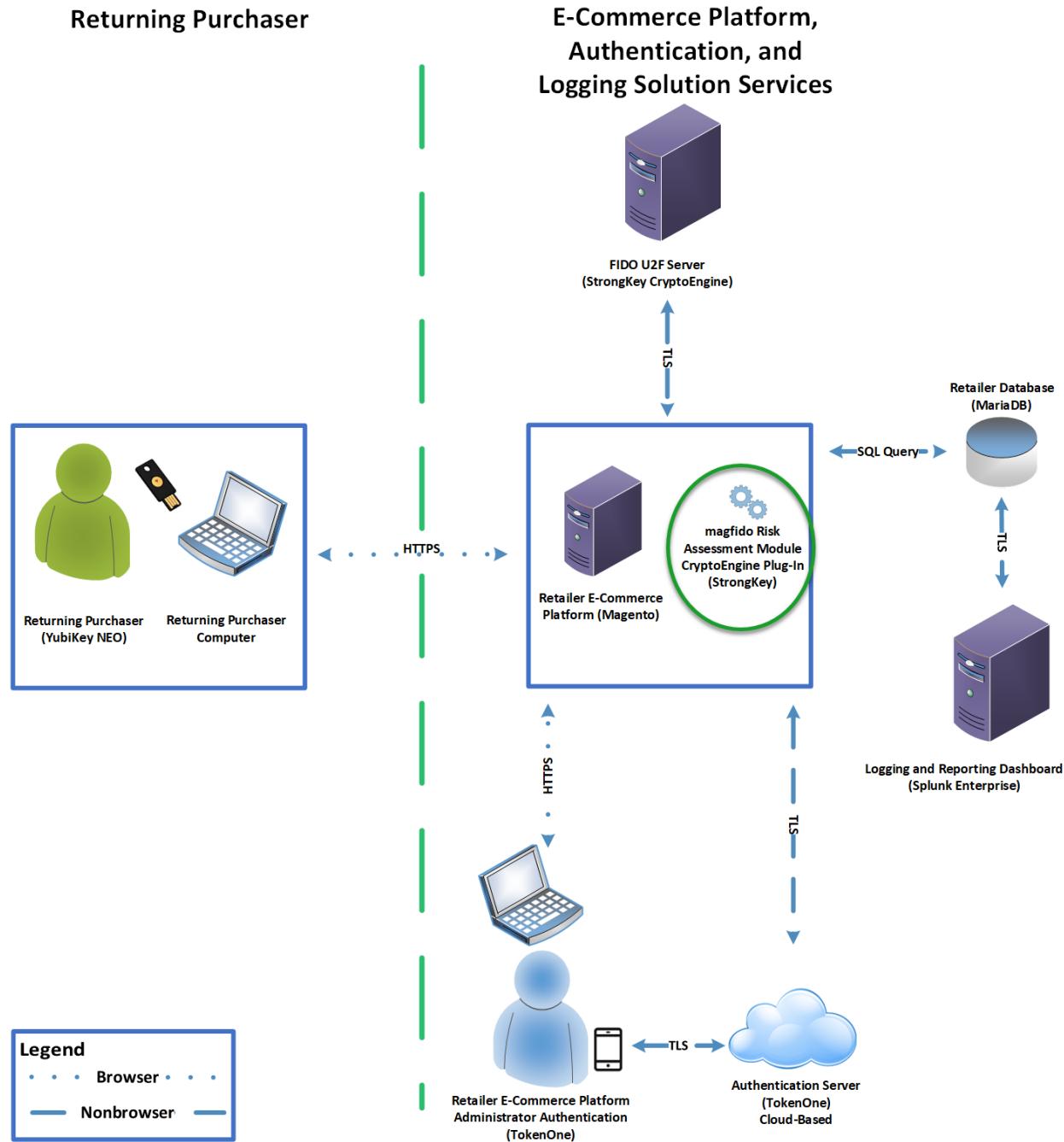
2.3 StrongKey magfido Module

641 This section of the guide provides installation and configuration guidance for the StrongKey magfido
 642 *FIDO U2FAuthenticator* module [\[6\]](#). While the core feature of the magfido module is to enable U2F
 643 authentication, the magfido module also allows registration of FIDO U2F Security Keys. Additional
 644 information on magfido and how the registration feature works can be found in [Appendix A](#).

2.3.1 StrongKey magfido Overview

646 The magfido module is used in the *cost threshold* example implementation build to examine the
 647 shopping cart's characteristics and to recommend whether MFA is required for the returning purchaser.
 648 The magfido module will modify the default behavior of Magento to register *FIDO U2FAuthenticators*,
 649 also known as FIDO Security Keys, and for FIDO authentication on purchases that exceed a total of \$25.
 650 The StrongKey magfido components that are installed by using the instructions in this section are
 651 illustrated in [Figure 2-3](#) (circled in green).

652 Figure 2-3 StrongKey magfido Module Components



653

654 **2.3.2 StrongKey magfido Installation and Configuration**

655 The installation procedure consists of the following steps.

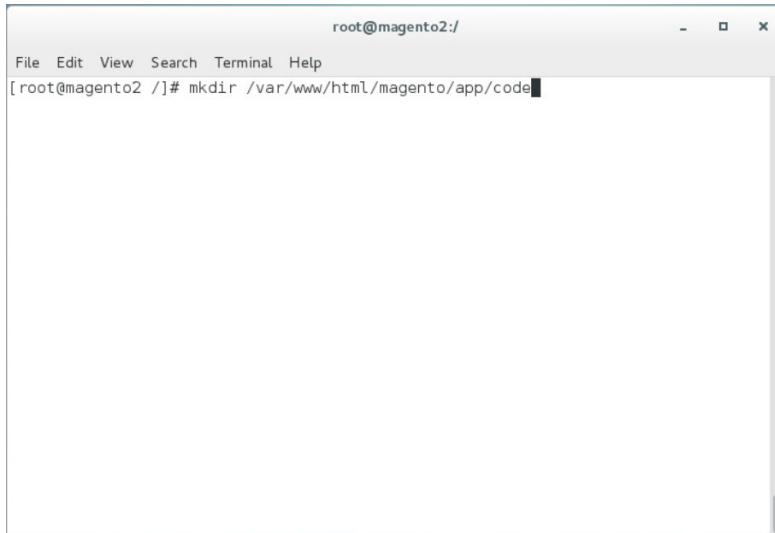
- 656 ▪ Download the software module to the Magento server where magfido will be installed.
657 ▪ Execute commands as root/administrator.
658 ▪ Perform post-installation configuration.

659 Navigate to the following site, and proceed to download the code:

660 <https://sourceforge.net/projects/magfido/>.

661 1. Create a code directory inside Magento's app folder by entering the following command:

662 `mkdir /var/www/html/magento/app/code`

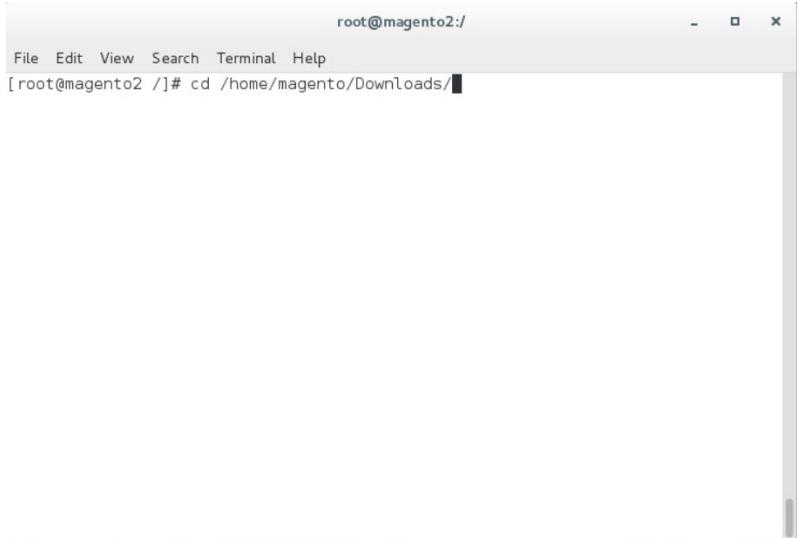


A screenshot of a terminal window titled "root@magento2:/" with a standard window title bar. The window contains a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu is a command line interface. The user has typed the command "[root@magento2 /]# mkdir /var/www/html/magento/app/code" and is pressing the Enter key. The cursor is positioned at the end of the command line.

663

664 2. Change your current directory to the Downloads directory by entering the following command:

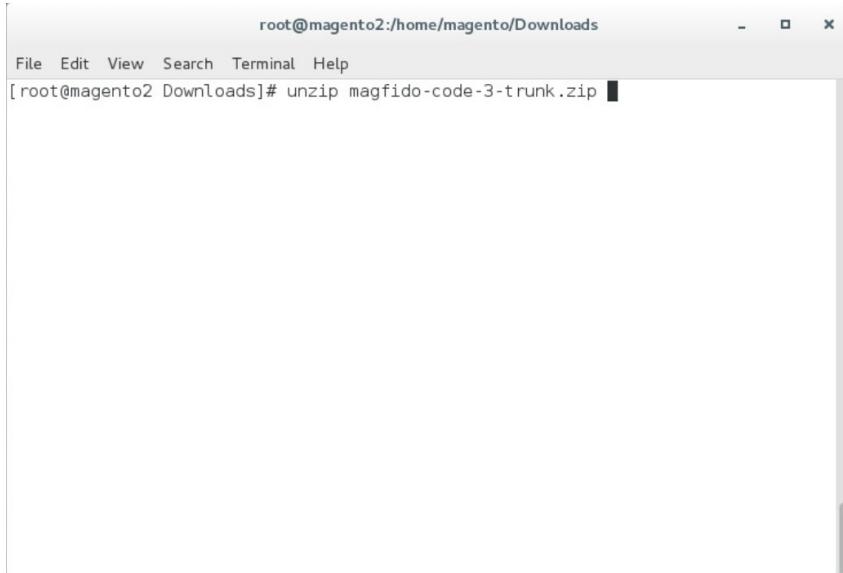
665 `cd /home/magento/Downloads/`



A screenshot of a terminal window titled "root@magento2:/". The window has a standard OS X-style title bar with icons for minimizing, maximizing, and closing. The menu bar below it includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane of the terminal shows the command "[root@magento2 /]# cd /home/magento/Downloads/". The terminal is currently inactive, indicated by a small black square icon in the bottom right corner of the main pane.

666

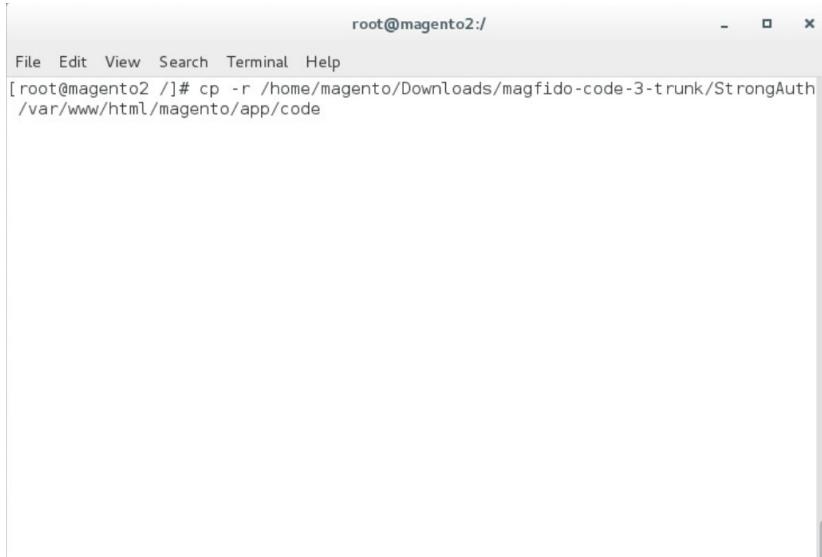
- 667 3. Unzip the *magfido-code-3-trunk.zip* by entering the following command:
668 unzip magfido-code-3-trunk.zip



A screenshot of a terminal window titled "root@magento2:/home/magento/Downloads". The window has a standard OS X-style title bar with icons for minimizing, maximizing, and closing. The menu bar below it includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane of the terminal shows the command "[root@magento2 Downloads]# unzip magfido-code-3-trunk.zip". The terminal is currently inactive, indicated by a small black square icon in the bottom right corner of the main pane.

669

- 670 4. Move the *StrongAuth_FIDO2FAuthenticator* module to the code directory by entering the following command:
671
672 cp -r home/magento/Downloads/magfido-code-3-trunk/StrongAuth
673 /var/www/html/magento/app/code



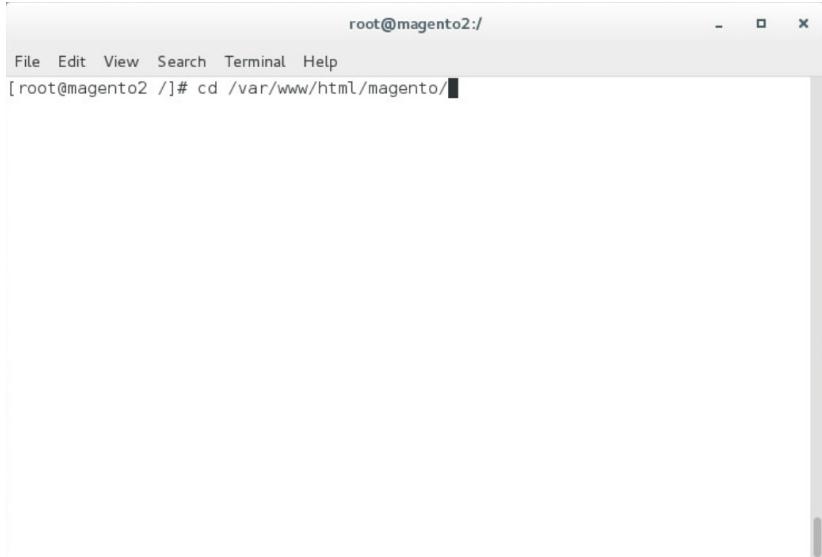
A screenshot of a terminal window titled "root@magento2:/" with a standard window control bar at the top. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane shows a command-line session:

```
root@magento2:~# cp -r /home/magento/Downloads/magfido-code-3-trunk/StrongAuth /var/www/html/magento/app/code
```

674

675 5. Change directories to the Magento directory by entering the following command:

676 cd /var/www/html/magento



A screenshot of a terminal window titled "root@magento2:/" with a standard window control bar at the top. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane shows a command-line session:

```
root@magento2:~# cd /var/www/html/magento/
```

677

678 6. Enable the *StrongAuth_FIDO2FAuthenticator* module by entering the following command:

679 php bin/magento module:enable StrongAuth_FIDO2FAuthenticator

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# php bin/magento module:enable StrongAuth_FIDO2FAuthent
icator
```

680

- 681 7. Register the *StrongAuth_FIDO2FAuthenticator* module by entering the following command:
- 682 php bin/magento setup:upgrade

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# php bin/magento setup:upgrade
```

683

- 684 8. Recompile dependencies by entering the following command:
- 685 php bin/magento setup:di:compile

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# php bin/magento setup:di:compile
```

686

687 9. Adjust the compiled file permissions by entering the following command:

```
688 chown -R apache:apache /var/www/html/magento && find var vendor pub/static
689 pub/media -type f -exec chmod u+w {} \; && find var vendor pub/static pub/media
690 -type d -exec chmod u+w {} \; && chmod u+x bin/magento
```

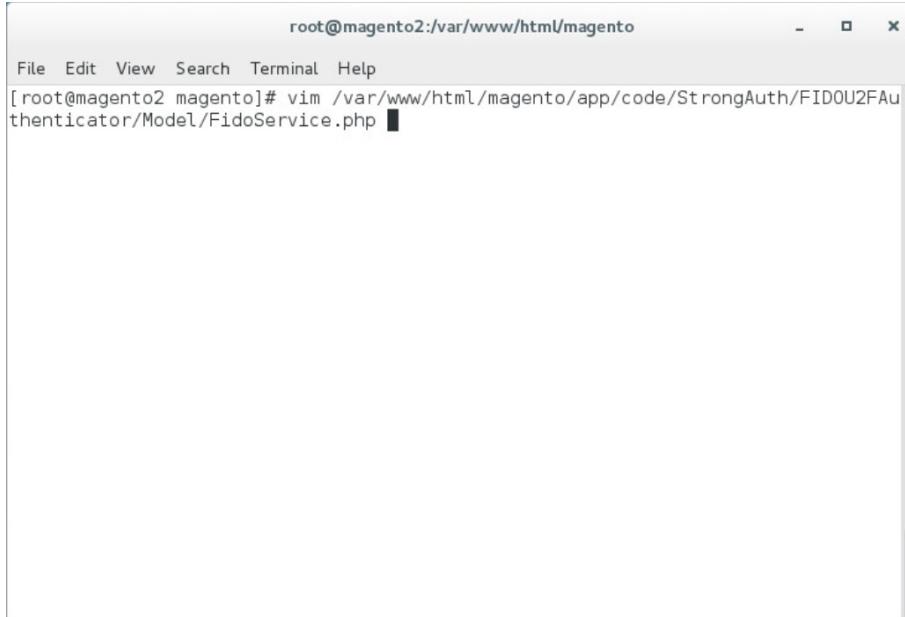
```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# chown -R apache:apache /var/www/html/magento && find va
r vendor pub/static pub/media -type f -exec chmod u+w {} \; && find var vendor p
ub/static pub/media -type d -exec chmod u+w {} \; && chmod u+x bin/magento
```

691

692 10. If SKCE is installed locally in your environment, then continue with the following steps:

693 a. Open *FidoService.php* by entering the following command:

694 Vim
695 /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/Fido
696 Service.php



```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# vim /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/FidoService.php
```

697

698 b. Modify the file to include the following information:

```

root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
namespace StrongAuth\FIDO2FAuthenticator\Model;

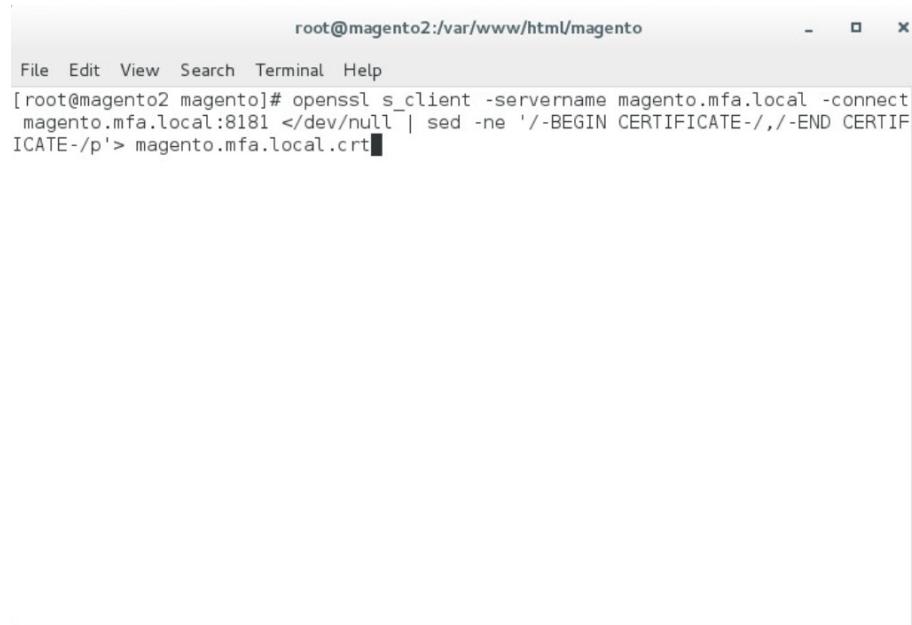
use StrongAuth\FIDO2FAuthenticator\Api\FidoServiceInterface;

class FidoService implements FidoServiceInterface
{
    const DID = "1";
    const SVCUSERNAME = "SVCFIDouser";
    const SVCPASSWORD = "Abcd1234!";
    const PROTOCOL = "U2F_V2";
    const VERSION = "1.0";
    const LOCATION = "unknown";
    const WSDL = "https://magento.mfa.local:8181/skfe/soap?wsdl";
    private $clientFactory;
    private $quoteRepository;

    public function __construct(\Magento\Framework\Webapi\Soap\ClientFactory $clientFactory, \Magento\Quote\Api\CartRepositoryInterface $quoteRepository) {
        $this->clientFactory = $clientFactory;
        $this->quoteRepository = $quoteRepository;
    }

699    public function preauthenticate($cartId) {
700
701        i. The DID parameter is the Domain ID of SKCE.
702
703        ii. The SVCUSERNAME parameter is the SKCE user responsible for authorizing
704            requests to the FIDO server.
705
706        iii. The SVCPASSWORD parameter is the password of the SKCE user.
707
708        iv. The PROTOCOL, VERSION, and LOCATION are parameters used for reference for
709            the FIDO server. They should be left as-is.
710
711        v. The WSDL (Web Services Description Language) parameter specifies the web ser-
712            vice endpoint with which the Magento server will communicate to send web-ser-
713            vice requests to the FIDO server. The default SKCE install will have the WSDL as
714            "https://<fully-qualified-domainname>:8181/skfe/soap?wsdl."
715
716        c. Retrieve a copy of the FIDO server's TLS digital certificate by entering the following
717            command (Note: This is a single command that must be executed on a single line.):
718
719            openssl s_client -servername <fully-qualified-domain-name> -connect
720            <fully-qualified-domain-name>:8181 </dev/null | sed -ne '/BEGIN
721            CERTIFICATE-/,/-END CERTIFICATE-/p' > <FQDN>.crt

```

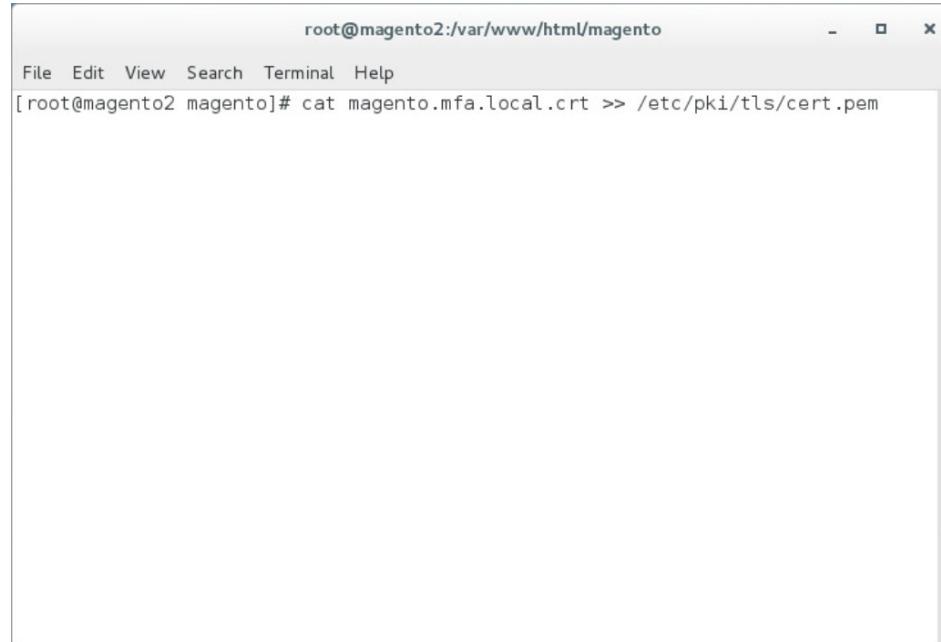


```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# openssl s_client -servername magento.mfa.local -connect magento.mfa.local:8181 </dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > magento.mfa.local.crt
```

715

716 d. Add the certificate to the list of trusted certificates by entering the following command:

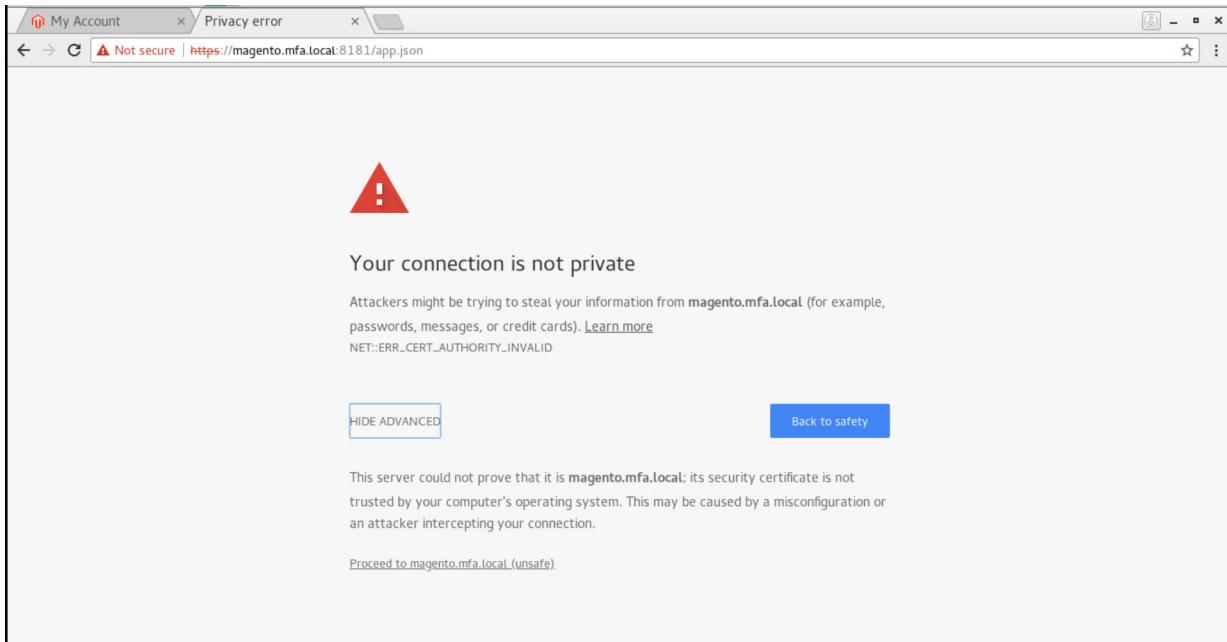
717 cat <fully-qualified-domain-name>.crt >> /etc/pki/tls/cert.pem



```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# cat magento.mfa.local.crt >> /etc/pki/tls/cert.pem
```

718

719 e. Open the Chrome browser and navigate to https://magento.mfa.local:8181/app.json.



720

721 i. A warning will appear, stating that “Your connection is not private.”

722 ii. Click **HIDE ADVANCED**.

723 iii. Click **Proceed to <fully-qualified-domain-name> (unsafe)**.

724 f. On your SKCE machine, edit the *app.json* file by entering the following command:

725 vim
726 usr/local/strongauth/payara41/glassfish/domains/domain1/docroot/app.json

```
# magento:/> vim usr/local/strongauth/payara41/glassfish/domains/domain1/docroot/app.json
```

727

728 g. Add the FQDN of the machine hosting the Magento application in the *ids* array, and save
729 the file.

730

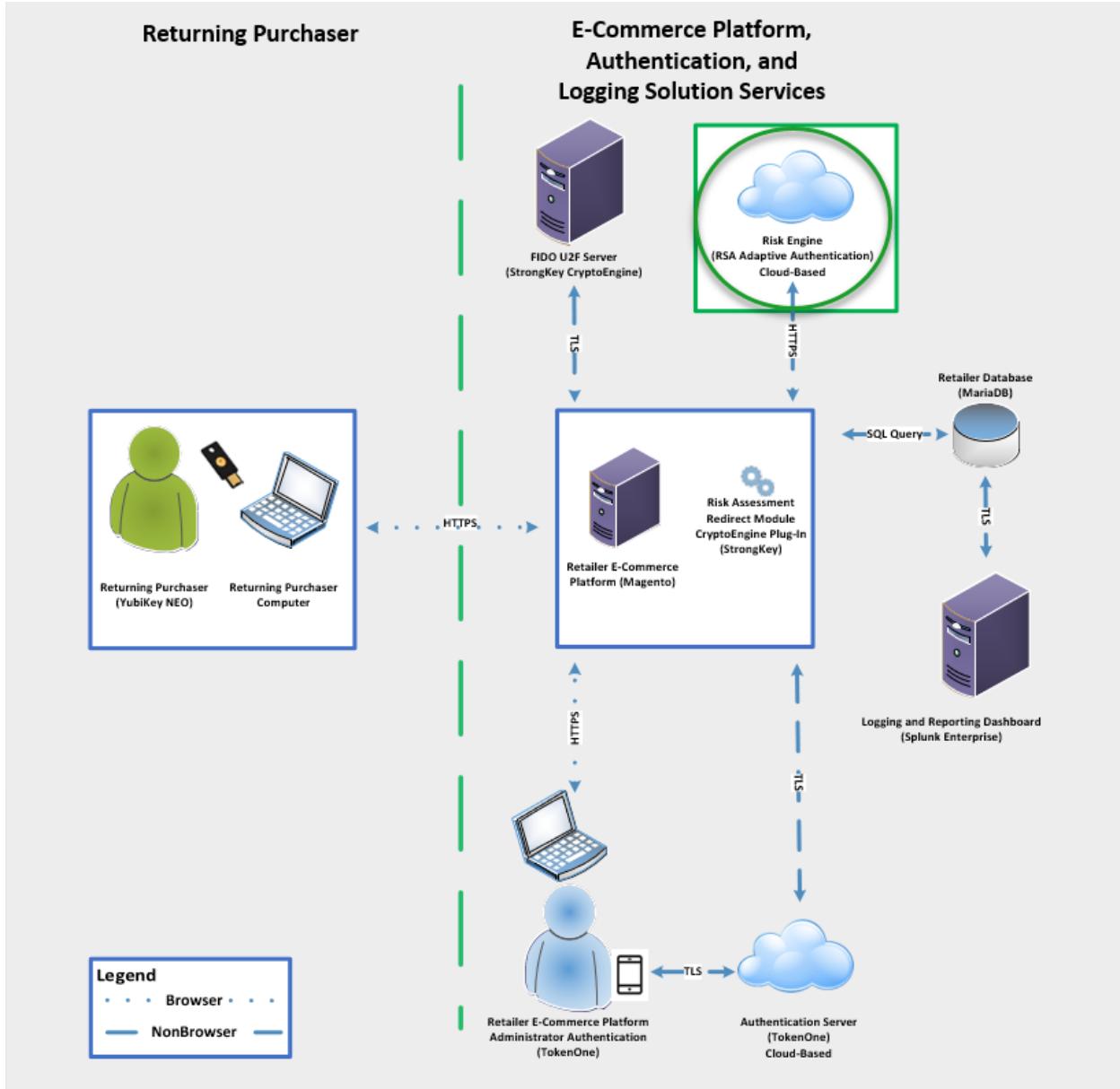
```
{  
    "trustedFacets": [  
        {"version": {"major": 1, "minor": 0},  
        "ids": [  
            "https://magento.mfa.local",  
            "https://magento.mfa.local:8181",  
            "https://magento2.mfa.local"  
        ]  
    ]  
}
```

731

2.4 RSA Adaptive Authentication

732 This section of the guide provides installation and configuration guidance for the RSA Adaptive
733 Authentication risk engine. The RSA Adaptive Authentication product performs a risk analysis and then
734 prompts the returning user to provide an MFA authenticator when required for the *risk engine* example
735 implementation build. The purpose of the RSA Adaptive Authentication is to minimize fraud with a low-
736 friction consumer experience. This example implementation uses the RSA Adaptive Authentication cloud
737 offering. The components that integrate Magento with RSA Adaptive Authentication are installed by
738 using the instructions in this section. The components are illustrated in [Figure 2-4](#) (circled in green).

739 Figure 2-4 RSA Adaptive Authentication Components



740

741 [2.4.1 RSA Overview](#)

742 RSA [7] offers an Adaptive Authentication [8] capability, which is part of the *risk engine* example
743 implementation.

744 The installation procedure consists of the following steps:

- 745 ▪ Preinstallation:
 - 746 • Download the RSA Project Library.
 - 747 • Configure Magento to accept additional extension attributes.
- 748 ▪ Installation and configuration:
 - 749 • Integrate RSA files into Magento.
 - 750 • Create policy in RSA Back Office.

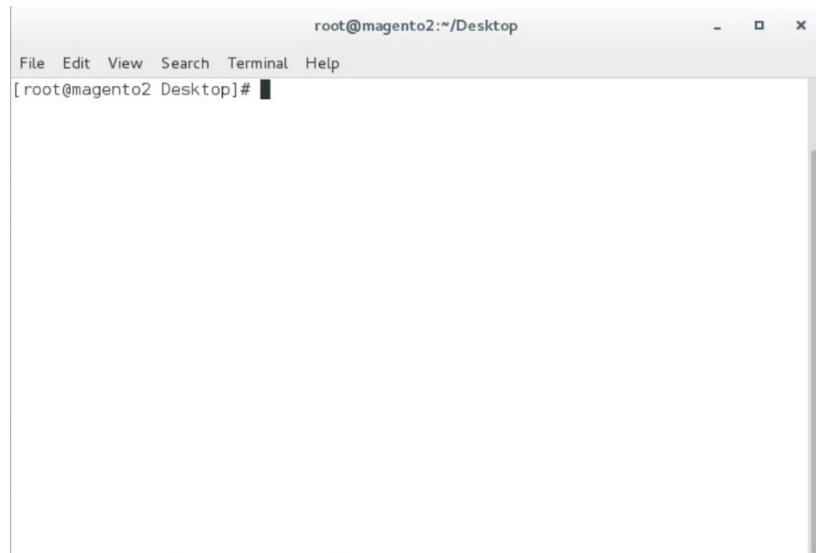
751 [2.4.2 RSA Preinstallation Steps](#)

752 Before beginning installation, perform the following steps.

- 753 ▪ Contact your RSA representative regarding access to RSA project library files (RSA.zip) and
754 RSA.php files. Download these files to the /home/magento/Downloads directory.
- 755 ▪ Configure Magento to accept additional extension attributes as outlined below.

756 This section will discuss how to add extension attributes to Magento to pass necessary information to
757 RSA Adaptive Authentication.

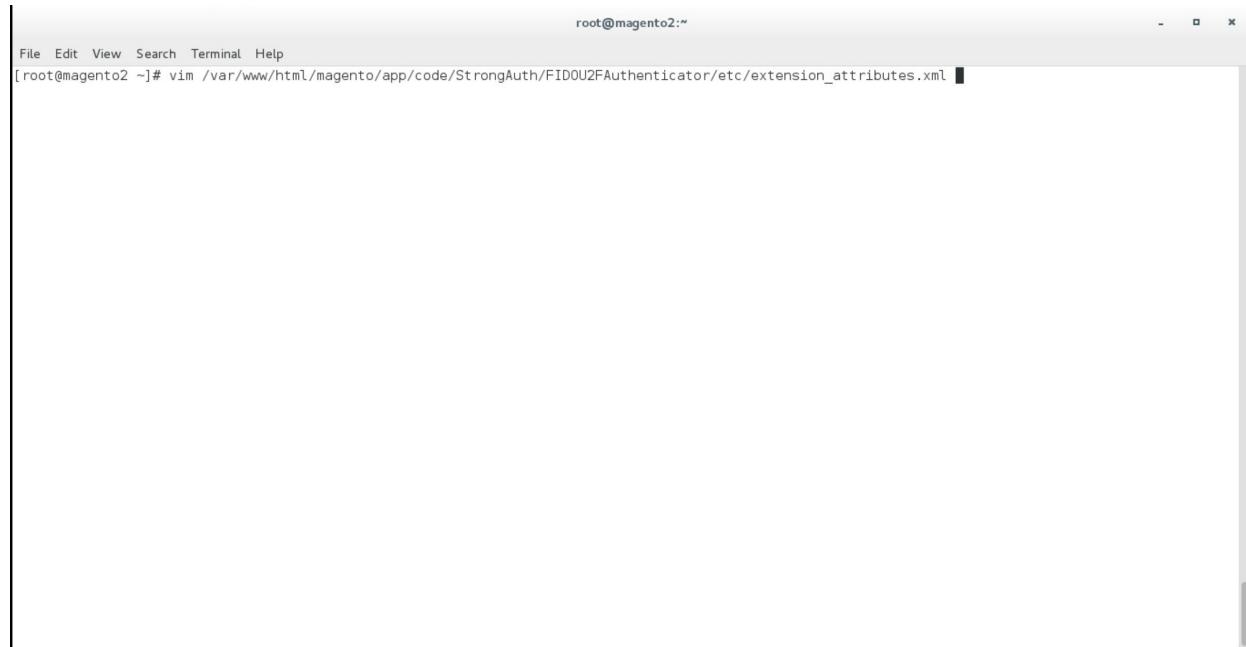
758 1. Open a terminal window.



759

760 2. To edit the file containing Magento's extension attributes, issue the following commands:

761 a. vim /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthentica-
762 tor/etc/extension_attributes.xml



The screenshot shows a terminal window titled 'root@magento2:~'. The window has a standard Linux-style interface with a title bar, menu bar, and scroll bars. The terminal content area displays the command: [root@magento2 ~]# vim /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthentica-tor/etc/extension_attributes.xml. The cursor is visible at the end of the command.

763

764 b. Press **i** to enter insertion mode.

765 3. Following Line 53, which contains <attribute code="signature" type="string" />, insert
766 the following lines (shown in the picture below):

767 <attribute code="email" type="string"/>
768 <attribute code="deviceprint" type="string"/>
769 <attribute code="cookie" type="string"/>
770 <attribute code="httplang" type="string"/>
771 <attribute code="useragent" type="string"/>
772 <attribute code="httpref" type="string"/>

```

root@magento2:~ - x
File Edit View Search Terminal Help
* $Date: 2018-02-02 14:42:01 -0800 (Fri, 02 Feb 2018) $
* $Revision: 381 $
* $Author: mishimoto $
* $URL:
*
* ****
*      888
*      888
*      888
* 88888b. .d88b. 888888 .d88b. .d8888b
* 888 "88b d88" "88b 888 d8P Y8b 88K
* 888 888 888 888 888 88888888 "Y8888b.
* 888 888 Y88..88P Y88b. Y8b. X88
* 888 888 "Y88P" "Y888 "Y8888 88888P"
*
* ****
* Tells Magento 2 that Payment information will have an attribute
* from our extension called signature.
*
*/
-->
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Api/etc/extension_attributes.xsd">
    <extension_attributes for="Magento\Quote\Api\Data\PaymentInterface">
        <attribute code="signature" type="string" />
        <attribute code="email" type="string"/>
        <attribute code="deviceprint" type="string"/>
        <attribute code="cookie" type="string"/>
        <attribute code="httpLang" type="string"/>
        <attribute code="useragent" type="string"/>
        <attribute code="httpref" type="string"/>
    </extension_attributes>
</config>
-- INSERT --

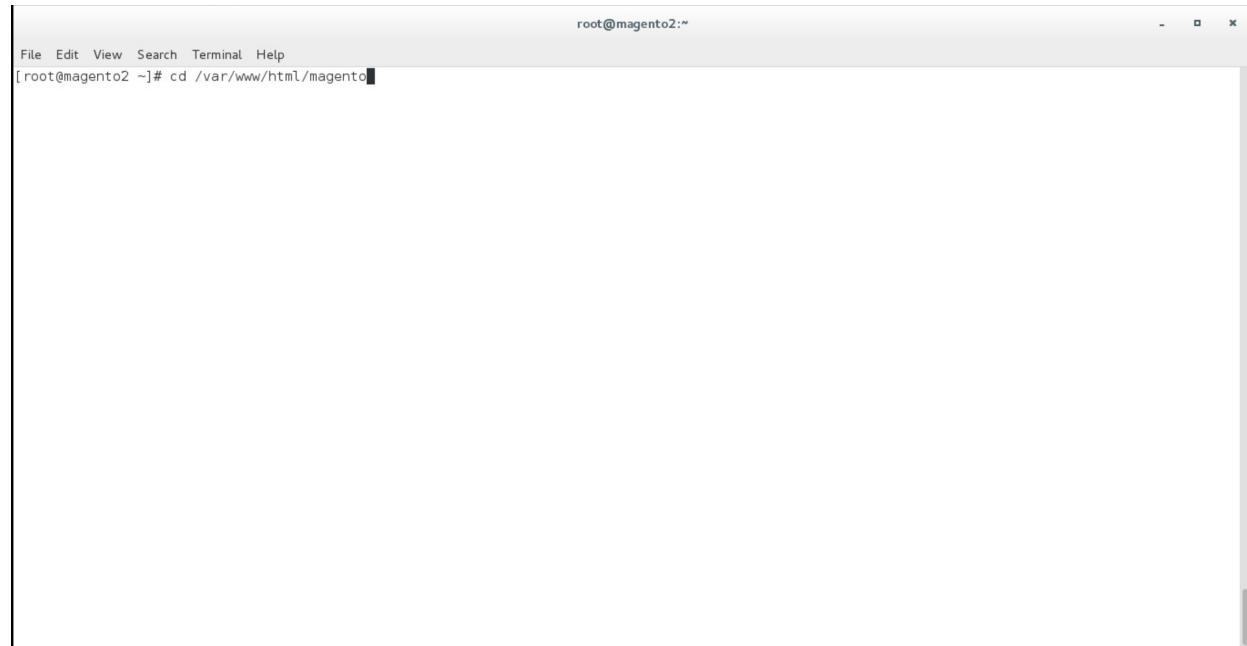
```

773

42,53

Bot

- 774 4. Press the Esc key to exit insert mode.
- 775 5. Save changes, and exit by entering the following command: :wq.
- 776 6. Return to the terminal window.
- 777 7. Change to the Magento folder by entering the following command:
- 778 cd /var/www/html/magento

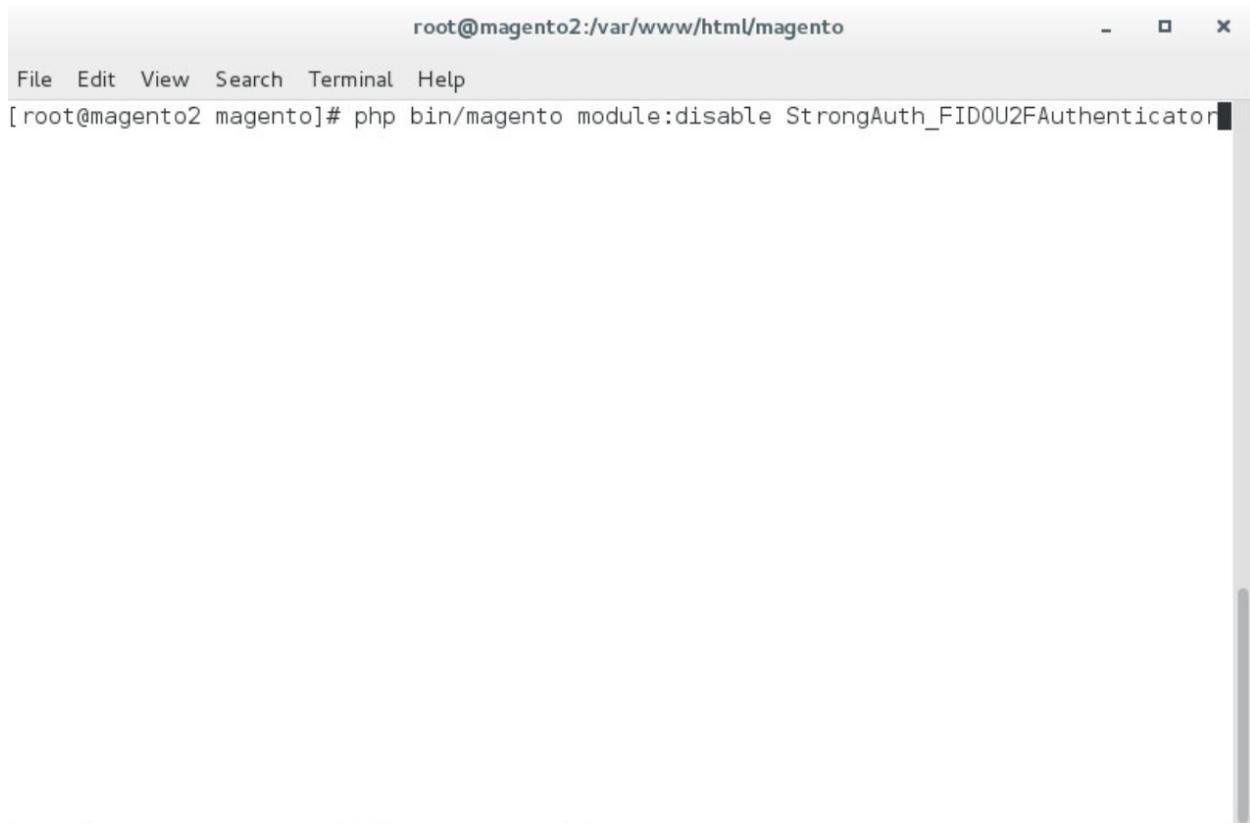


A screenshot of a terminal window titled 'root@magento2:~'. The window has a standard Linux-style interface with a menu bar at the top. In the main area, the command '[root@magento2 ~]# cd /var/www/html/magento' is visible, with the cursor positioned at the end of the path. The background of the terminal is light gray, and the text is black.

779

780 8. To recompile Magento to reflect the changes made to the extension attributes file, issue the fol-
781 lowing commands:

782 a. `php bin/magento module:disable StrongAuth_FIDO2FAuthenticator`



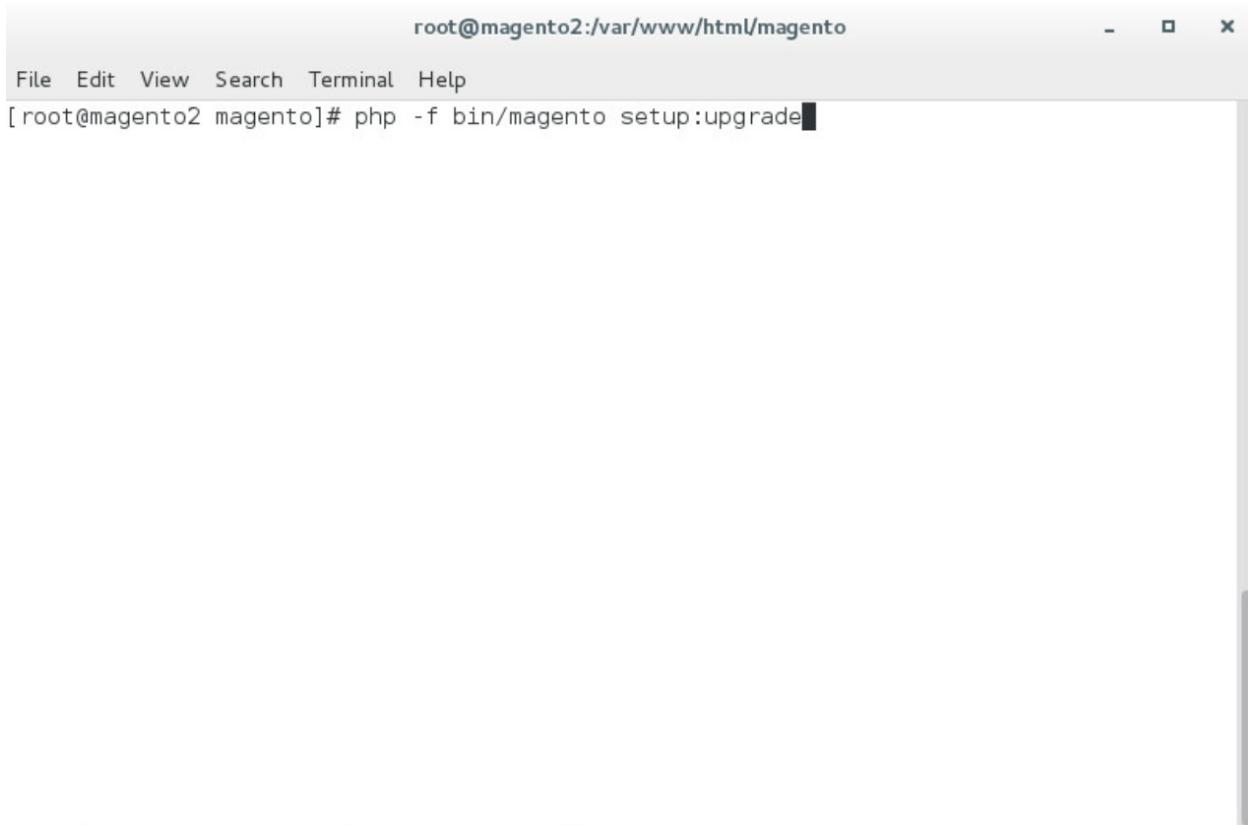
A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has standard Linux-style window controls at the top right. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane displays a command-line session:

```
root@magento2:/var/www/html/magento
[ root@magento2 magento]# php bin/magento module:disable StrongAuth_FIDO2FAuthenticator
```

783

784

b. `php -f bin/magento setup:upgrade`

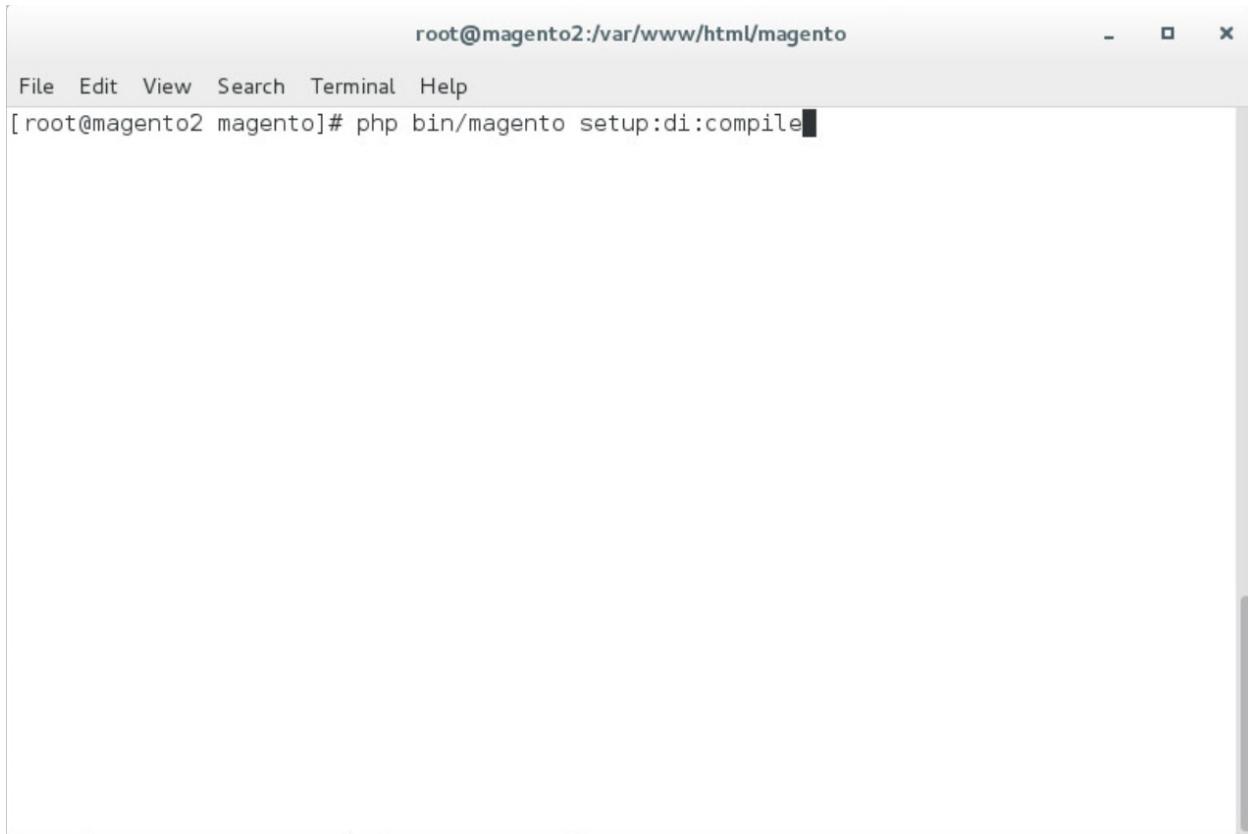


A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has a standard OS X-style title bar with icons for minimizing, maximizing, and closing. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane shows the command line: "[root@magento2 magento]# php -f bin/magento setup:upgrade". The cursor is positioned at the end of the command.

785

786

c. php bin/magento setup:di:compile

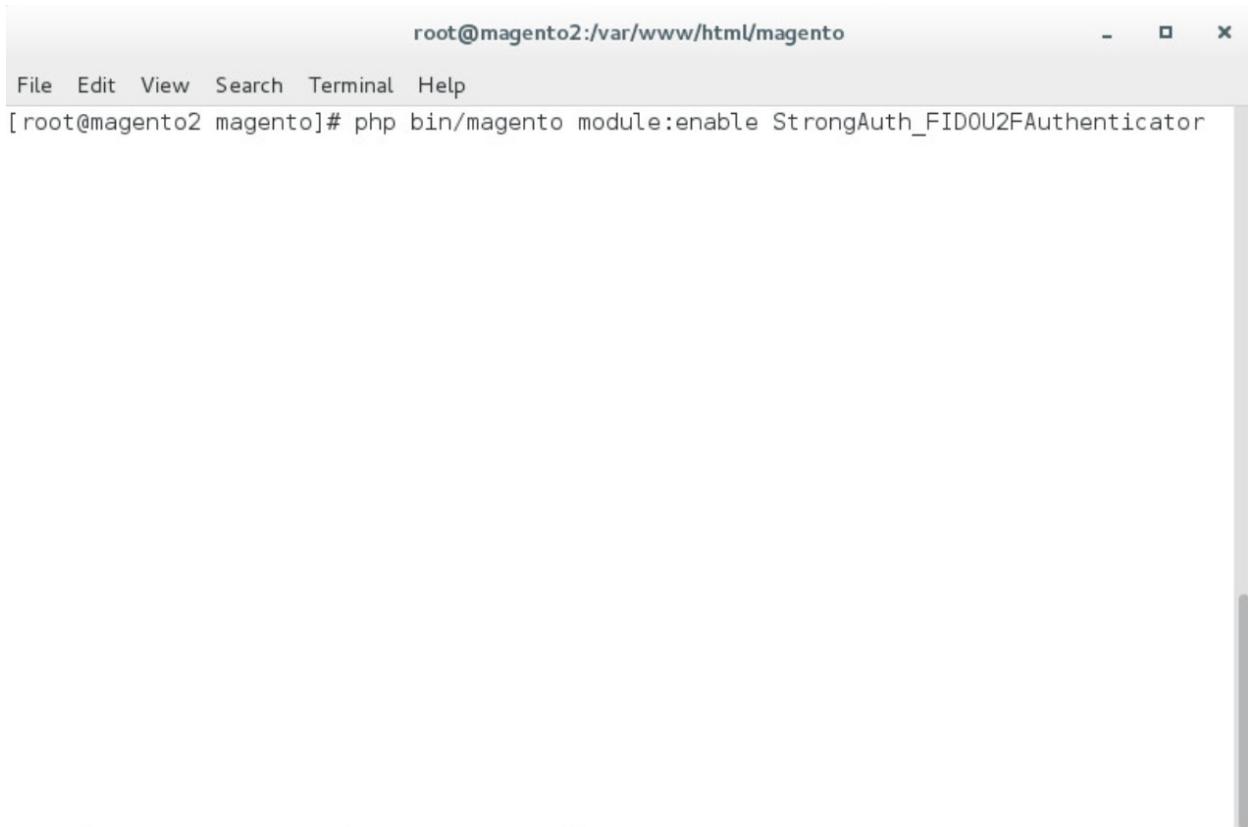


A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has a standard OS X-style title bar with icons for minimizing, maximizing, and closing. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane shows a command-line session where the user is executing a PHP command: "[root@magento2 magento]# php bin/magento setup:di:compile". The terminal is currently at the end of this command, indicated by a cursor at the end of the line.

787

788

d. `php bin/magento module:enable StrongAuth_FIDO2FAuthenticator`



A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has standard Linux-style window controls at the top right. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu is a command line prompt: "[root@magento2 magento]#". The user has entered the command "php bin/magento module:enable StrongAuth_FIDO2FAuthenticator". The terminal is set against a light gray background with a vertical scroll bar on the right side.

789

790

e. `php bin/magento setup:di:compile`

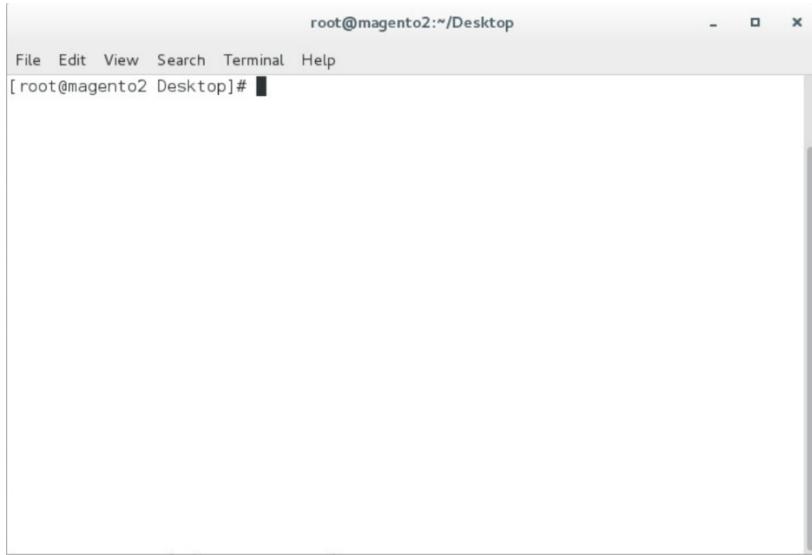
```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# php bin/magento setup:di:compile
```

791

792 2.4.3 Adaptive Authentication Installation and Configuration

793 This section provides a step-by-step installation guide for integrating RSA Adaptive Authentication.
794 Before you begin, make sure that you have received your RSA project libraries from your RSA
795 representative.

796 1. Open a terminal window.

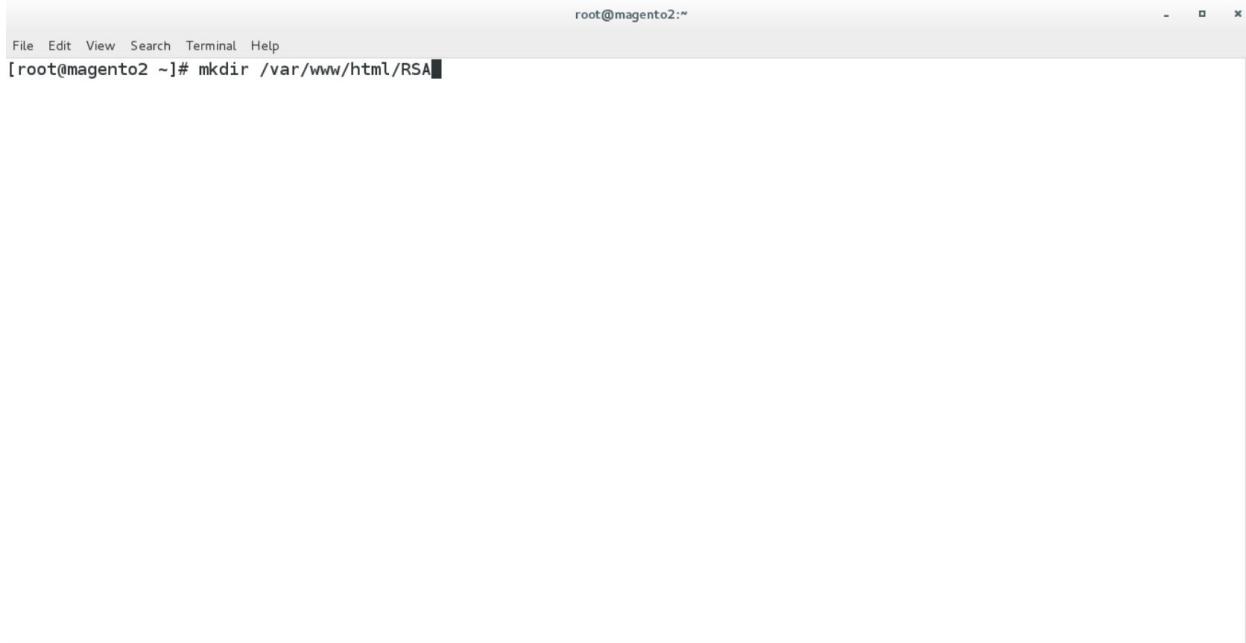


A screenshot of a terminal window titled "root@magento2:~/Desktop". The window has a standard title bar with icons for minimize, maximize, and close. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal is a white space with a cursor at the bottom-left corner, indicating it is ready for input. The prompt "[root@magento2 Desktop]#" is visible at the top of the terminal window.

797

798 2. Create a new directory by entering the following command:

799 Mkdir /var/www/html/RSA



A screenshot of a terminal window titled "root@magento2:~". The window has a standard title bar with icons for minimize, maximize, and close. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the command "mkdir /var/www/html/RSA" being typed into the input field. The cursor is positioned at the end of the command. The prompt "[root@magento2 ~]#" is visible at the top of the terminal window.

800

801 3. Obtain the RSA zip file from your RSA representative.

802 4. Change to the Downloads directory by entering the following command:

803 cd /home/magento/Downloads

```
root@magento2:~  
File Edit View Search Terminal Help  
[root@magento2 ~]# cd /home/magento/Downloads/
```

804

805 5. Unzip the RSA directory by entering the following command:

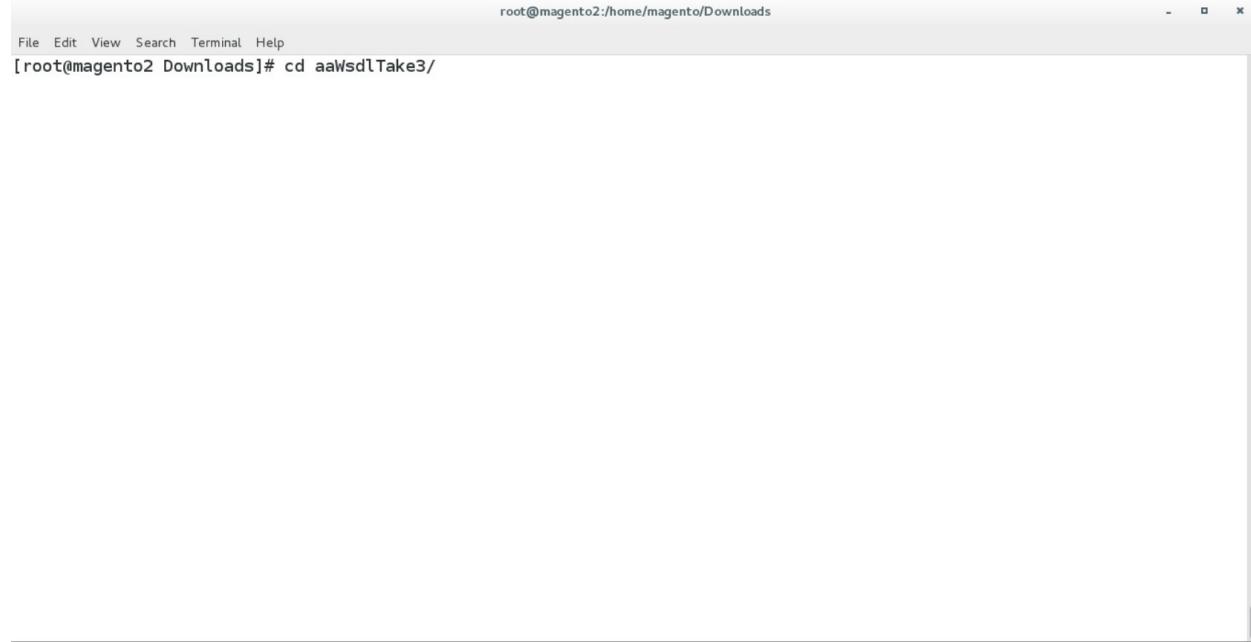
806 unzip RSA.zip

```
root@magento2:/home/magento/Downloads  
File Edit View Search Terminal Help  
[root@magento2 Downloads]# unzip RSA.zip
```

807

808 6. Change to the newly unzipped directory by entering the following command:

809 cd aaWsdlTake3/



A screenshot of a terminal window titled "root@magento2:/home/magento/Downloads". The window has a standard Linux-style interface with a menu bar at the top. In the terminal area, the command "cd aaWsdlTake3/" is being typed. The cursor is positioned after the final slash of the command.

810

811 7. Copy the contents of the API runtime directory to the RSA directory, which was created in Step 2
812 by entering the following command:

813 cp resources/aa13/aa70api-runtime/* /var/www/html/RSA/

```
root@magento2:/home/magento/Downloads/aaWsdlTake3
File Edit View Search Terminal Help
[root@magento2 aaWsdlTake3]# cp resources/aa13/aa70api-runtime/* /var/www/html/RSA/
```

814

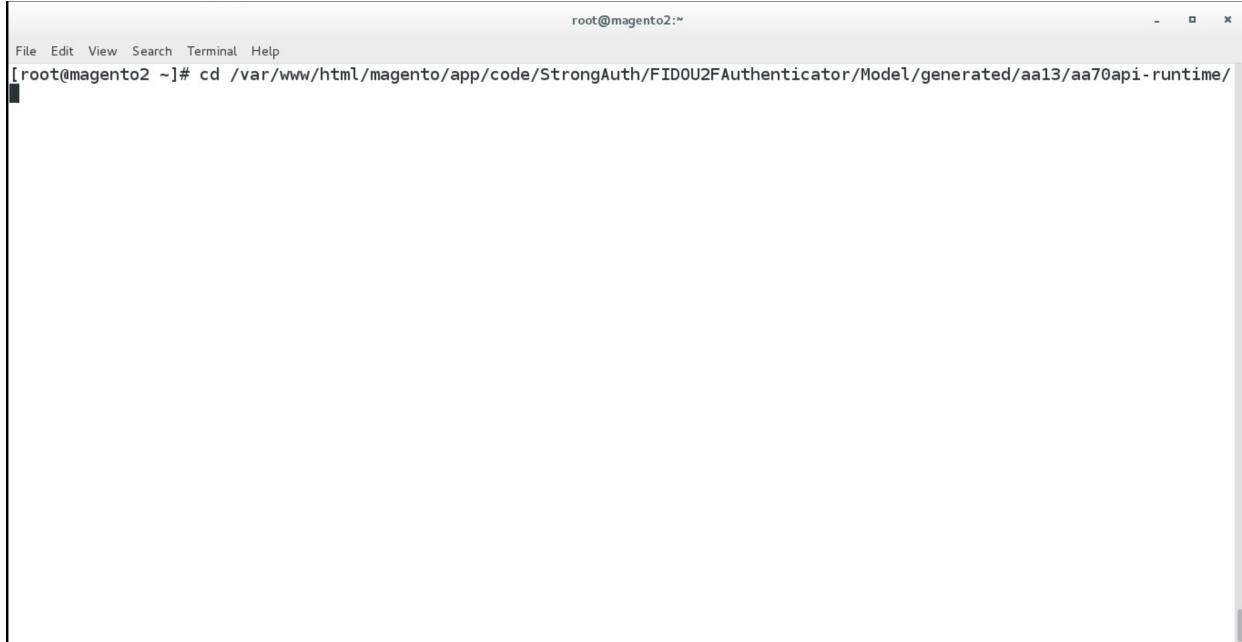
- 815 8. Copy the contents of the aaWsdlTake3 directory to the StrongAuth model directory by entering
816 the following command:
817 cp -R ./* /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/

```
root@magento2:/home/magento/Downloads/aaWsdlTake3
File Edit View Search Terminal Help
[root@magento2 aaWsdlTake3]# cp -R ./* /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/
```

818

819 9. Change to the generated RSA API runtime folder by entering the following command:

820 cd
821 /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/generated/
822 aa13/aa70api-runtime/



A screenshot of a terminal window titled "root@magento2:~". The window shows the command "cd /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/generated/aa13/aa70api-runtime/" being typed at the prompt. The terminal has a standard Linux-style interface with a menu bar and a scroll bar on the right.

823

824 10. Edit the Adaptive Authentication file by entering the following command:

825 vim AdaptiveAuthentication.php

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/generated/aa13/aa70api-runtime

File Edit View Search Terminal Help

[root@magento2 aa70api-runtime]# vim AdaptiveAuthentication.php

826

827 11. Make edits in the Adaptive Authentication file by pressing the **i** key to enter insert mode.

828 12. Change Line 297 of the document to the following line:

829 \$wsdl = 'http://magento2.mfa.local/RSA/AdaptiveAuthentication.wsdl';

```

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/generated/aa13/aa70api-runtime
File Edit View Search Terminal Help
 * @param array $options A array of config values
 * @param string $wsdl The wsdl file to use
 */
public function __construct(array $options = array(), $wsdl = null)
{
    foreach ($self::$classmap as $key => $value) {
        if (!isset($options['classmap'][$key])) {
            $options['classmap'][$key] = $value;
        }
    }
    $options = array_merge(array(
        'features' => 1,
    ), $options);
    if (!$wsdl) {
        $wsdl = 'http://magento2.mfa.local/RSA/AdaptiveAuthentication.wsdl';
    }
    parent::__construct($wsdl, $options);
}

/**
 * @param notify $parameters
 * @return void
 */
public function notify(notify $parameters)
{
    return $this->__soapCall('notify', array($parameters));
}
-- INSERT --

```

297, 70-77 81%

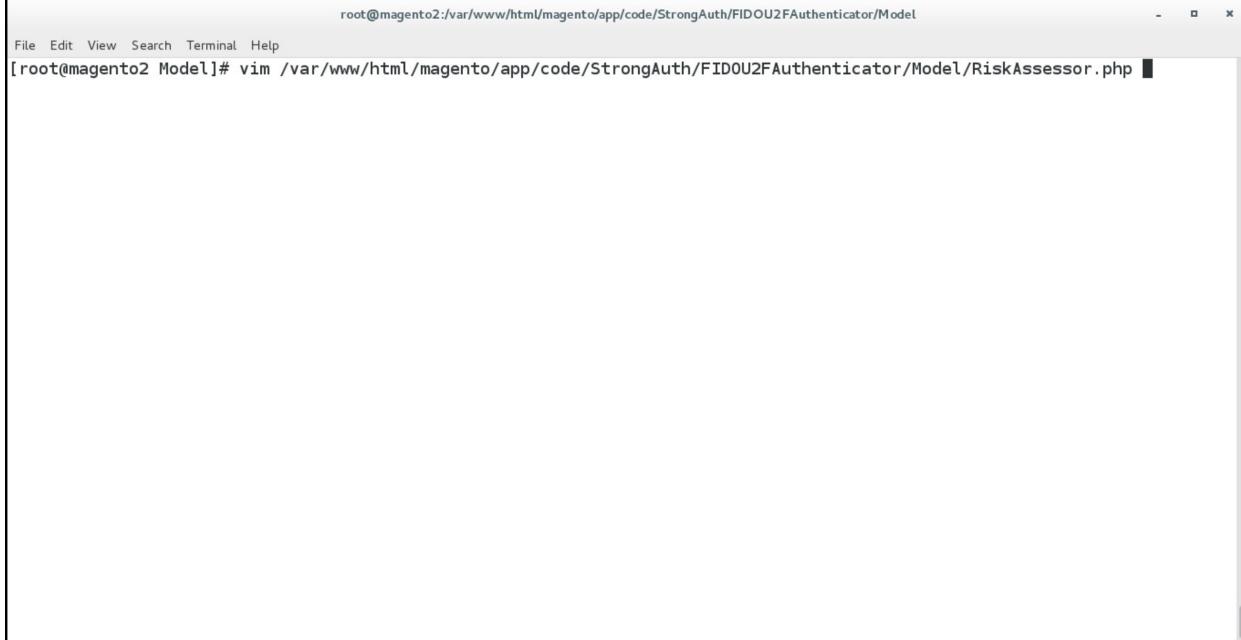
830

831 13. Press the Esc key to exit insert mode.

832 14. Save changes, and exit by entering the following command: :wq.

833 15. Edit the RSA Risk Assessor File by entering the following command:

834 vim
835 /var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Model/RiskAssess
836 or.php



The screenshot shows a terminal window titled 'root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Model'. The window has a standard Linux terminal interface with tabs for 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The command '[root@magento2 Model]# vim /var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Model/RiskAssessor.php' is visible at the bottom of the window. The rest of the screen is blank white space.

837

838 16. Press the i key to enter editor mode.

839 17. Make the following changes to the *RiskAssessor.php* file:

840 a. After Line 41, add the following two lines:

841 use RSA;
842 require_once('RSA.php');

```

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Model
File Edit View Search Terminal Help
*/
namespace StrongAuth\FIDO2FAuthenticator\Model;

use StrongAuth\FIDO2FAuthenticator\Api\RiskAssessorInterface;
use RSA; //add
require_once('RSA.php');//add

class RiskAssessor implements RiskAssessorInterface
{
    private $quoteRepository;

    public function __construct(\Magento\Quote\Api\CartRepositoryInterface $quoteRepository) {
        $this->quoteRepository = $quoteRepository;
    }

```

843

844 b. Change Line 55 to the following line:

845 Public function isFidoNeeded(\$cartId, \$email, \$deviceprint, \$cookie,
 846 \$httpplan, \$useragent, \$httpref)

```

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Model
File Edit View Search Terminal Help
private $quoteRepository;

public function __construct(\Magento\Quote\Api\CartRepositoryInterface $quoteRepository) {
    $this->quoteRepository = $quoteRepository;
}

#params in this instance is the cartId passed as a JSON string.
public function isFidoNeeded($cartId, $email, $deviceprint, $cookie, $httplang, $useragent, $httpref) { //add
    #If the user provided invalid information, force FIDO authentication

```

847

848 c. After Line 65, edit the following lines:

849 \$test = new RSA;
 850 \$amount = \$test->rsaAACall(\$cartId, \$email, \$deviceprint, \$cookie,
 851 \$httpplan, \$useragent, \$httpref);
 852 return \$amount;

```
root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model

File Edit View Search Terminal Help

if($cartId === null) {
    return true;
}
#Check that the cart exceeds $25 before requiring FIDO authentication
else {
    //document below
    $quote = $this->quoteRepository->getActive($cartId);
    $carttotal = $quote->getGrandTotal();
    $test = new RSA;
    $ammount= $test->rsaAACall($carttotal, $email, $deviceprint, $cookie, $httppling, $useragent, $httpref); //add
    return $ammount;
}

} //else
}

}

-- INSERT --
65,43-50 Bot
```

- 853
- 854 d. Press the **Esc** key to exit insert mode.
- 855 e. Save changes, and exit by entering the following command: `:wq`.
- 856 18. Open the *PIMOverrideFidoAuthenticate.php* file in the vim editor by entering the following command:
- 857
- 858 vim
- 859 `/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model/PIMOverrideFidoAuthenticate.php`
- 860



```
root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Model
File Edit View Search Terminal Help
[root@magento2 Model]# vim /var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Model/PIMOverrideFidoAuthenticate.php
```

861

862 19. Press the **i** key to enter editor mode.863 20. Make the following changes to the *PIMOverrideFidoAuthenticate.php* file:

864 a. Between Lines 68 and 72, edit the following lines:

```
865         extData = $paymentMethod->getExtensionAttributes();
866         if($this->riskAssessorFactory->create()->isFidoNeeded($cartId,$extData-
867             >getEmail(),$extData->getDeviceprint(),$extData->getCookie,$extData-
868             >getHttpLang(),$extData->getUseragent,$extData->getHttpref())) {
```

```

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model
File Edit View Search Terminal Help
) {
    $this->fidoServiceFactory = $fidoServiceFactory;
    $this->riskAssessorFactory = $riskAssessorFactory;
    parent::__construct($billingAddressManagement, $paymentMethodManagement, $cartManagement, $paymentDetailsFactory,
$cartTotalsRepository);
}
#Documentation Needed to add passed variables to savepayment order email...httpref
public function savePaymentInformationAndPlaceOrder(
    $cartId,
    \Magento\Quote\Api\Data\PaymentInterface $paymentMethod,
    \Magento\Quote\Api\Data\AddressInterface $billingAddress = null
) {
    $extData = $paymentMethod->getExtensionAttributes(); //add

    #Checks if Fido Authentication is needed
    if($this->riskAssessorFactory->create() ->isFidoNeeded($cartId,$extData->getEmail(),$extData->getDeviceprint(),$ext
Data->getCookie(),$extData->getHttpLang(),$extData->getUserAgent(),$extData->getHttpref())) {///add
        #If Fido Authentication is needed, verify that a signature was provided and that it is valid.
        $extensionData = $paymentMethod->getExtensionAttributes();
        if($extensionData === null || $extensionData->getSignature() === null) {
            throw new \Exception("No Signature provided");
        }
        $result = $this->fidoServiceFactory->create() ->authenticate($cartId, json_decode($extensionData->getSignature(
)));
        if(strpos($result->return, "Successfully") === false) {
            throw new \Exception($result->return);
        }
        else {
            #Save the payment information and place the order only if the signature was valid.
-- INSERT --

```

72,222 85%

869

870 b. Press the Esc key to exit insert mode.

871 c. Save changes, and exit by entering the following command: :wq.

872 21. Open the RSA RiskAssessor Controller file by entering the following command:

```

873 vim
874 /var/www/html/magento/StrongAuth/FIDOU2FAuthenticator/Controller/Index/Riskasse
875 ssor.php

```



root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Model
File Edit View Search Terminal Help
[root@magento2 Model]# vim /var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/Controller/Index/RiskAssessor.php

876

877 22. Press the **i** key to enter editor mode.

878 23. Make the following changes to the *RiskAssessor.php* file:

879 a. Change Line 60 to the following line:

```
880       $result = $this->riskAssessorFactory->create() -  
881       >isFidoNeeded($params['cartId'], $params['email'],  
882       $params['deviceprint'], $params['cookie'], $params['httpLang'],  
883       $params['useragent'], $params['httpref']);
```

```

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/Model
File Edit View Search Terminal Help
* or not.
*/
namespace StrongAuth\FIDOU2FAuthenticator\Controller\Index;
use Magento\Framework\App\Action\Context;
use StrongAuth\FIDOU2FAuthenticator\Model\RiskAssessorFactory;
use Magento\Framework\Controller\Result\JsonFactory;
class RiskAssessor extends \Magento\Framework\App\Action\Action
{
    protected $riskAssessorFactory;
    protected $jsonFactory;
    public function __construct(Context $context, RiskAssessorFactory $riskAssessorFactory, JsonFactory $jsonFactory) {
        parent::__construct($context);
        $this->riskAssessorFactory = $riskAssessorFactory;
        $this->jsonFactory = $jsonFactory;
    }
    #Calls the isFidoNeeded method of the RiskAssessor Model. cartId is passed to the model to allow it to make decisions
    #based on the items in the "shopping cart" (and the customer associated with the cart).
    public function execute() {
        $params = $this->getRequest()->getPostValue();
        $result = $this->riskAssessorFactory->create()->isFidoNeeded($params['cartId'],$params['email'],$params['deviceprint'],$params['cookie'],$params['httplang'],$params['useragent'],$params['httpref']);//add
        $resultJson = $this->jsonFactory->create();
        return $resultJson->setData($result);
    }
}
?>
-- INSERT --

```

884

60,3

Bot

885 b. Press the Esc key to exit insert mode.

886 c. Save changes, and exit by entering the following command: :wq.

887 24. Open the RSA JavaScript Override file by entering the following command:

```

888 vim
889 /var/www/html/magento/StrongAuth/FIDOU2FAuthenticator/view/frontend/web/js/defa
890 ult-payment-override.js

```

```
root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/view/frontend/web/js
File Edit View Search Terminal Help
[root@magento2 js]# vim /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/view/frontend/web/js/default-payment-override.js
```

891

892 25. Press the **i** key to enter editor mode.

893 26. Make the following changes to the *default-payment-override.js* file:

894 a. Add the following two lines after Line 57:

895 'StrongAuth_FIDOU2FAuthenticator/js/lib/hashtable',
896 'StrongAuth_FIDOU2FAuthenticator/js/lib/rsa'

897

```

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/view/frontend/web/js
File Edit View Search Terminal Help
* appended to the order information and then sent to the server.
*/
define([
    'jquery',
    'Magento_Checkout/js/action/place-order',
    'Magento_Checkout/js/model/payment/additional-validators',
    'Magento_Checkout/js/action/redirect-on-success',
    'Magento_Ui/js/modal/modal',
    'mage/url',
    'Magento_Checkout/js/model/quote',
    'fidoCommon',
    'fidou2f',
    'StrongAuth_FIDOU2FAuthenticator/js/lib/hashtable',//add
    'StrongAuth_FIDOU2FAuthenticator/js/lib/rsa'//add

    ],
    function($, placeOrderAction, additionalValidators, redirectOnSuccessAction, modal, url, quote, common, U2f, hash, rsa) {
    //use strict';

    return function(targetModule) {
        return targetModule.extend({
            //Overrides the default placeOrder function
            placeOrder: function(data, event){
                console.log("Place Order Pressed");
                //Performs some client side validations that exist in the default placeOrder function
                var self = this;
                if(event) {
                    event.preventDefault();
                }
                if(this.validate() && additionalValidators.validate()) {
                    this.isPlaceOrderActionAllowed(false);
                }
            }
        });
    };
-- INSERT --

```

57,15 17%

898

b. Change Line 83 to the following line:

899
900
901
902

```

Data: {cartId: quote.getQuoteId(), email : window.customerData.email,
deviceprint : encode_deviceprint(), cookie: document.cookie, httplang :
window.navigator.language, useragent : navigator.userAgent, httpref :
document.referrer},

```

903

```

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/view/frontend/web/js
File Edit View Search Terminal Help
placeOrder: function(data, event){
    console.log("Place Order Pressed");
    //Performs some client side validations that exist in the default placeOrder function
    var self = this;
    if(event) {
        event.preventDefault();
    }
    if(this.validate() && additionalValidators.validate()) {
        this.isPlaceOrderActionAllowed(false);

        //Makes a call to the Magento server to determine if FIDO Authentication is needed
        $.ajax({
            type: 'POST',
            url: url.build('fidou2fauthenticator/index/riskassessor/'),
            data: {cartId : quote.getQuoteId(), email : window.customerData.email, deviceprint : encode_device
print(), cookie : document.cookie, httplang : window.navigator.language, useragent : navigator.userAgent, httpref : docume
nt.referrer}, //add
            dataType: 'json'
        }).then(function(isFidoNeeded) {
            console.log('Printing stuff above');
            console.log('FIDO Authentication needed: ' + isFidoNeeded);

            //If FIDO Authentication isn't needed, perform the default behavior
            //Note: The server also performs these checks on its side, so even
            //if a malicious user overrides the client side code, the server will
            //block the purchase.
            if(!isFidoNeeded) {
                self.getPlaceOrderDeferredObjectOverride(null) //changed
            }
        });
    }
-- INSERT --

```

83,264 26%

904

c. Change Line 95 to the following line:

905

```
self.getPlaceOrderDeferredObjectOverride(null)
```

```
root@Magento2:/var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/view/frontend/web/js
File Edit View Search Terminal Help
    dataType: 'json'
}); then(function(isFidoNeeded) {
  console.log('Printing stuff above');
  console.log('FIDO Authentication needed: ' + isFidoNeeded);

  //If FIDO Authentication isn't needed, perform the default behavior
  //Note: The server also performs these checks on its side, so even
  //if a malicious user overrides the client side code, the server will
  //block the purchase.
  if(!isFidoNeeded) {
    self.getPlaceOrderDeferredObjectOverride(null) //add
      .fail(function() {
        self.isPlaceOrderActionAllowed(true);
        console.log(data);

      })
      .done(function() {
        self.afterPlaceOrder();
        if(self.redirectAfterPlaceOrder) {
          redirectOnSuccessAction.execute();
        }
      });
  }
  //If FIDO Authentication is needed:
} else {
-- INSERT --
95,81 32%
```

906

907

d. After Line 268, add the following lines:

908

```
Data['extension_attributes']['email'] = window.customerData.email;
Data['extension_attributes']['deviceprint'] = encode_deviceprint();
Data['extension_attributes']['cookie'] = document.cookie;
Data['extension_attributes']['httplang'] = window.navigator.language;
Data['extension_attributes']['useragent'] = navigator.userAgent;
Data['extension_attributes']['httpref'] = document.referrer;
```

```

root@magento2:/var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/view/frontend/web/js
File Edit View Search Terminal Help
}
else {
    return false;
}
},
//Overrides the default getPlaceOrderDeferredObjectOverride function to append the signature data to the data
sent to the server.
getPlaceOrderDeferredObjectOverride: function(response) {
    console.log("Combining signature data with order information");
    var data = this.getData();
    if(data['extension_attributes'] === undefined) {
        data['extension_attributes'] = {};
    }
    data['extension_attributes']['signature'] = JSON.stringify(response);
    data['extension_attributes']['email'] = window.customerData.email; //add
    data['extension_attributes']['deviceprint'] = encode_deviceprint();
    data['extension_attributes']['cookie'] = document.cookie;
    data['extension_attributes']['htplang'] = navigator.language;
    data['extension_attributes']['useragent'] = navigator.userAgent;
    data['extension_attributes']['httpref'] = document.referrer;
    console.log("Combining signature data success");
    console.log(data);
    return $.when(placeOrderAction(data, this.messageContainer));
}
);
};

-- INSERT --

```

914

268, 86

Bot

- 915 e. Press the **Esc** key to exit insert mode.
- 916 f. Save changes, and exit by entering the following command: `:wq`.
- 917 27. Download the RSA JavaScript files from your RSA representative.
- 918 28. Make the following change to the Downloads directory:
- 919 `cd /home/magento/Downloads`

```
root@magento2:~  
File Edit View Search Terminal Help  
[root@magento2 ~]# cd /home/magento/Downloads/
```

920

921 29. Unzip the contents of the RSA JavaScript folder by entering the following command:

922 unzip RSA_Scripts.zip

```
root@magento2:/home/magento/Downloads  
File Edit View Search Terminal Help  
[root@magento2 Downloads]# unzip RSA_Scripts.zip
```

923

924 30. Move to the newly unzipped scripts folder by entering the following command:

925 cd scripts/



A screenshot of a terminal window titled "root@magento2:/home/magento/Downloads". The window has a standard Linux-style interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar. In the terminal area, the command "[root@magento2 Downloads]# cd scripts/" is visible, with the cursor at the end of the command line.

926

927 31. Copy the *rsa.js* and *hashtable.js* files to StrongAuth front-end JavaScript directory by entering
928 the following commands:

929 a. cp rsa.js /var/www/html/magento/app/code/StrongAuth/FIDO2FAuthentica-
930 tor/view/frontend/web/js/lib/



root@magento2:/home/magento/Downloads/scripts

File Edit View Search Terminal Help

```
[root@magento2 scripts]# cp rsa.js /var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/view/frontend/
```

931

- 932 b. cp hashtable.js /var/www/html/magento/app/code/StrongAuth/FIDO2FAuthen-
933 ticator/view/frontend/web/js/lib/



root@magento2:/home/magento/Downloads/scripts

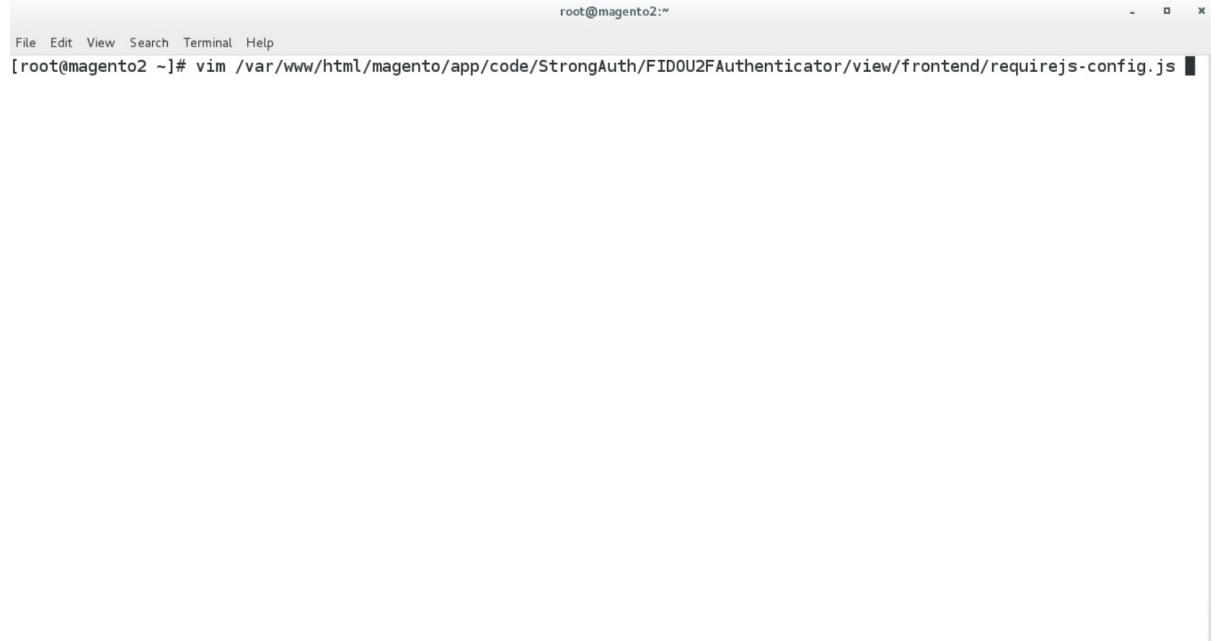
File Edit View Search Terminal Help

```
[root@magento2 scripts]# cp hashtable.js /var/www/html/magento/app/code/StrongAuth/FIDO2FAuthenticator/view/frontend/
```

934

935 32. Open the StrongAuth JavaScript required file by entering the following command:

936 vim
937 /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/view/frontendreq
938 uirejs-config.js



The screenshot shows a terminal window titled 'root@magento2:~'. The window contains the command '[root@magento2 ~]# vim /var/www/html/magento/app/code/StrongAuth/FIDOU2FAuthenticator/view/frontend/requirejs-config.js'. The terminal is running in root mode on a Linux system named 'magento2'.

939

940 33. Press the i key to enter editor mode.

941 34. Make the following edits to the *requirejs-config.js* file:

942 a. After Line 41, insert the following lines:

943 " hashtable" : "StrongAuth_FIDOU2FAuthenticator/js/lib/hastables",
944 "rsa" : "StrongAuth_FIDOU2FAuthenticator/js/lib/rsa"

```

root@magento2:~
File Edit View Search Terminal Help
*
* ****
* Imports the 3rd party Javascript libraries into RequireJS.
* In addition, overrides the default Javascript that is run
* when clicking the "Place Order" button.
*(Note) for Practice Guide Documentation Needed to add hashtable and rsa lines to path
*/
var config = {
  paths: {
    "fidoCommon" : "StrongAuth_FIDO2FAuthenticator/js/lib/common",
    "fidoU2f" : "StrongAuth_FIDO2FAuthenticator/js/lib/u2f-api",
    "hashtable" : "StrongAuth_FIDO2FAuthenticator/js/lib/hashtables",
    "rsa" : "StrongAuth_FIDO2FAuthenticator/js/lib/rsa"
  },
  shim: {
    'fidoU2f' : {
      exports: 'u2f'
    }
  },
  config: {
    mixins: {
      'Magento_Checkout/js/view/payment/default': {
        'StrongAuth_FIDO2FAuthenticator/js/default-payment-override' : true
      }
    }
  }
};
-- INSERT --

```

945

41, 76

Bot

946

b. Press the **Esc** key to exit insert mode.

947

c. Save changes, and exit by entering the following command: `:wq`.

948

2.4.4 RSA Adaptive Authentication Policy Creation

949

1. Open a web browser and navigate to the back-office URL supplied by your RSA representative.

950

The screenshot shows the RSA Adaptive Authentication login interface. At the top left is the RSA logo. Below it, the page title is "Adaptive Authentication". A "Login" form is centered, containing fields for "User Name" and "Password", both marked with red asterisks indicating they are required. A "Login" button is below the fields. A note at the bottom left says "* Required Field". At the bottom right of the page, there is a copyright notice: "Copyright © 2018 EMC Corporation. All Rights Reserved.".

951

- 952 2. Enter your RSA-supplied login credentials.
- 953 3. Open the **Policy Management Manage Rules** page by clicking **Policy Management > Manage Rules**.
- 954
- 955 4. Click **New**.

The screenshot shows the "Manage Rules" page under the "Policy Management" tab. The top navigation bar includes "Policy Management", "Administration", "Customer Service", and "Reports". On the right, there are user information ("Logged in as: Admin | Logout") and organization details ("Organization NCCoE"). The main content area is titled "Manage Rules" and contains a sub-instruction: "Manage rules using the table below. To edit a rule, click on the Rule Name." Below this is a table with the following columns: Order, Rule Name, Event Type, Current Status, Pending Status, Action, and Date Modified. The table header includes buttons for "New", "Delete", and "Status". Above the table, it says "1 items found" and "Showing 25 per page".

956

- 957 5. Under the **General** tab, edit the required fields with the following information:
- 958 a. **Rule Name:** Payment over 50

- 959 b. **Status:** Production
- 960 c. **Event Type:** PAYMENT
- 961 d. **Order:** 2
- 962 e. **Sample Size:** 100

Edit Rule

1: General 2: Conditions 3: Actions Summary

Define the general details for this rule.

Rule Details

*Rule Name: Payment over 50

Description:

*Status: Production

Comment:

*Event Type: [?]

- FAILED_CHANGE_PASSWORD_ATTEMPT
- FAILED_LOGIN_ATTEMPT
- FAILED_OLB_ENROLL_ATTEMPT
- OLB_ENROLL
- OPEN_NEW_ACCOUNT
- OPTIONS_TRADE
- PAYMENT
- READ_SECURE_MESSAGE

*Order: [?]

*Sample Size: % [?]

* Required Field

- 963
- 964 6. Click **Next**.
- 965 7. Under the **Conditions** tab, fill out the form with the following information:
- 966 a. **Select Category:** Transaction Details
- 967 b. **Select Fact:** Transaction Amount in USD
- 968 c. **Select Operator:** Greater than or Equal to
- 969 d. **USD:** 50

Edit Rule

1: General **2: Conditions** **3: Actions** **Summary**

Build the condition(s) for this rule using categories, facts, and operators. You must add at least one condition. Each condition must contain at least one expression.

Rule Conditions

Condition 1

Expression 1

Transaction Details → Transaction Amount in USD → Greater than or Equal to 50 USD

Remove Expression | Duplicate Expression

Join Multiple Expression By OR | Add New Expression

Add New Condition

Back Next Save & Exit Cancel

970

971 8. Click **Next**.972 9. Under the **Action** tab, fill out the form with the following information:973 a. **Action:** Challenge974 b. **Authentication Method(s):** EXTERNAL_METHOD1

New Rule

1: General **2: Conditions** **3: Actions** Summary

Define the action to occur when the rule conditions are met.

Rule Actions

* Action: Challenge

* Authentication Method(s):

Available Method(s): EXTERNAL_METHOD1, KBA, OOBBIOMETRICS, OOBPHONE, OOBSMS, OTP

Selected Method(s): [?]

Create Case:

When authentication fails [?] When authentication succeeds [?]

Back Next Save & Exit Cancel

975

976 c. **Create Case:** Leave the box checked for **When authentication fails**.977 10. Click **Next**.978 11. Review the new rule under the **Summary** tab.

New Rule

1: General 2: Conditions 3: Actions Summary

Review the rule before closing the wizard. Edit the rule as needed.

Rule Details

Rule Name: Payment Over 50
 Rule ID:
 Created By:
 Description:
 Status: Production
 Comment:
 Event Type: PAYMENT
 Rule Order: 1
 Inherited by All Organizations: No
 Sample Size: 100 %

Rule Conditions

If (Transaction Amount in USD **Greater than** 50 USD)

Rule Actions

Actions: Deny
 Create Case: Yes

Back Finish Cancel

979

980 12. Click **Finish**.981 13. To put the rule into production, click **Status > Approve Status**.982 14. In the **Approve Status** window, click **Approve**.

Approve Status

Review the rule status details and add any relevant comment before you approve the status change.

Rule Name: Payment Over 50
 Current Status: Work In Progress
 Pending Status: Production [?]
 Change Request: admin , 2018-06-01 11:00 (EST): No Comment
 Comment:

Approve Cancel

983

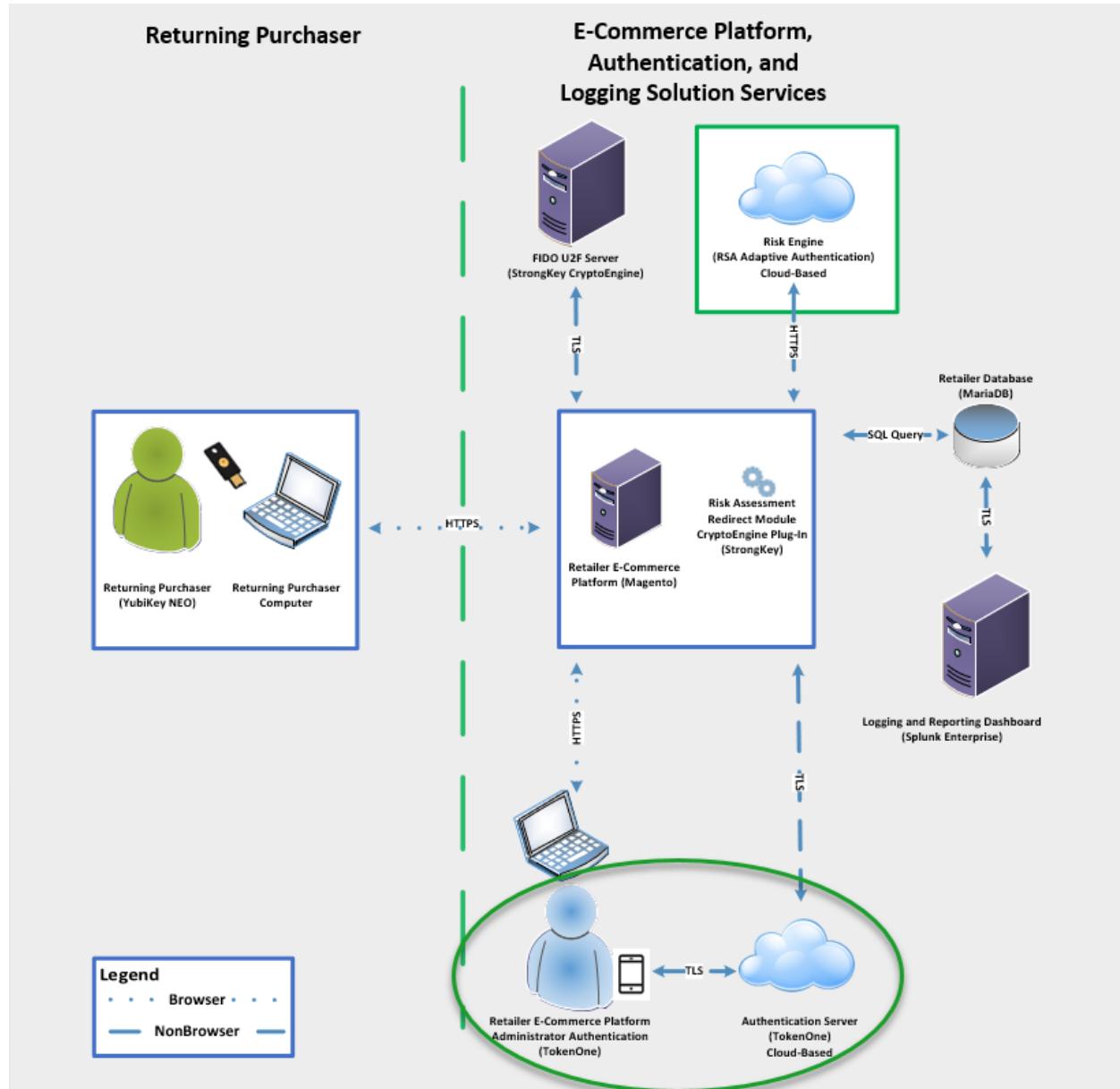
984

2.5 TokenOne

985 This section provides installation and configuration guidance for TokenOne's authentication capability
 986 [\[9\]](#). TokenOne's authentication product is used by the retailer e-commerce platform administrator when
 987 they are managing the Magento e-commerce platform. TokenOne developed a Magento connector that
 988 both the *cost threshold* and *risk engine* example implementations use. The TokenOne authentication

989 components that are installed and configured in this section are illustrated in [Figure 2-5](#) (circled in
 990 green).

991 **Figure 2-5 TokenOne Authentication Components**



992

993 **2.5.1 TokenOne Overview**

994 TokenOne allows software-based authentication through a one-time personal identification number
995 (PIN). The Magento Admin URI portal has been configured to use Second Factor Authentication with
996 TokenOne. When accessing Magento with TokenOne's authentication capability, the user's numeric PIN
997 is not entered, transmitted, or stored, but the corresponding letter code—which is entered when
998 accessing Magento—is different every time that the user accesses the system. The TokenOne
999 smartphone application is not push-button. The user always enters the code in the Magento
1000 administration interface.

1001 The installation procedure consists of the following steps:

- 1002 ▪ Preinstallation:
 - 1003 • Download the TokenOne application
 - 1004 • Download the TokenOne module.
- 1005 ▪ Installation and configuration:
 - 1006 • Download the TokenOne module.
 - 1007 • Integrate the TokenOne module into Magento.
 - 1008 • Test connectivity and authentication.

1009 **2.5.2 Preinstallation Steps**

1010 Before beginning installation, ensure that the following steps are completed:

- 1011 ▪ Download and install the TokenOne mobile application from either the Apple App Store or the
1012 Google Play Store.
- 1013 ▪ Speak with your TokenOne representative to receive the *TokenOne10.zip* file.
- 1014 ▪ Download the *TokenOne10.zip* file to the */home/magento/Downloads* directory.

1015 **2.5.3 TokenOne Installation and Configuration**

1016 To begin installation, perform the following steps:

- 1017 1. Open a terminal window.

```
root@magento2:~/Desktop
File Edit View Search Terminal Help
[root@magento2 Desktop]#
```

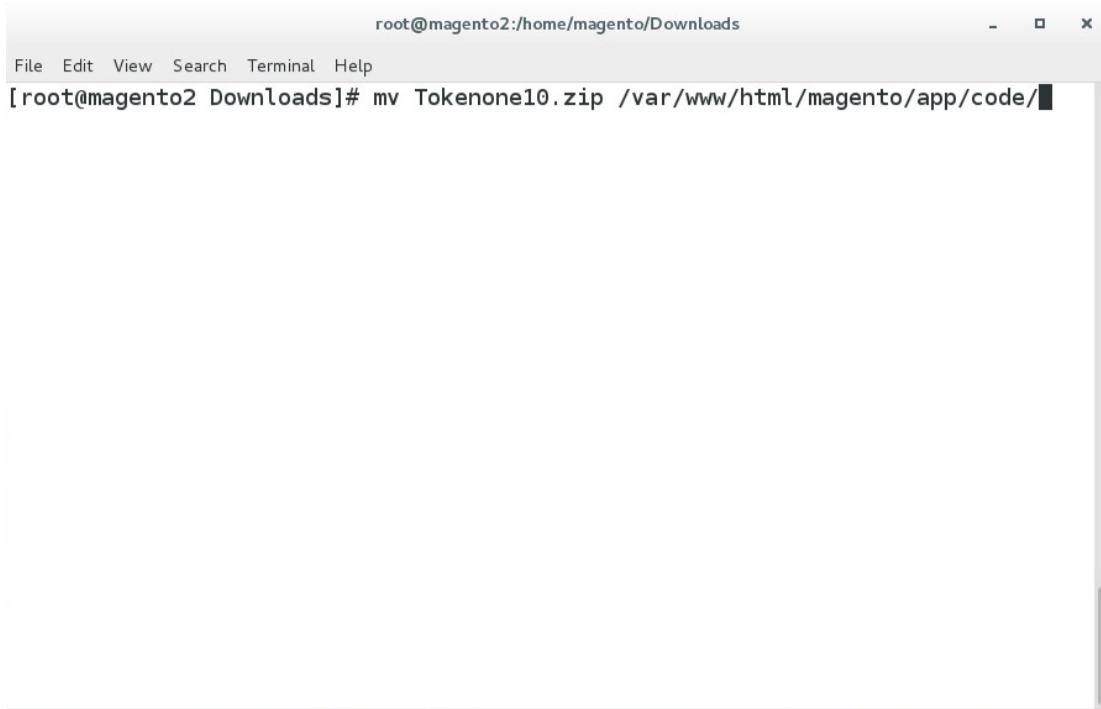
1018

- 1019 2. Change to the Downloads directory by entering the following command:
- 1020 cd /home/magento/Downloads

```
root@magento2:~
File Edit View Search Terminal Help
[root@magento2 ~]# cd /home/magento/Downloads/
```

1021

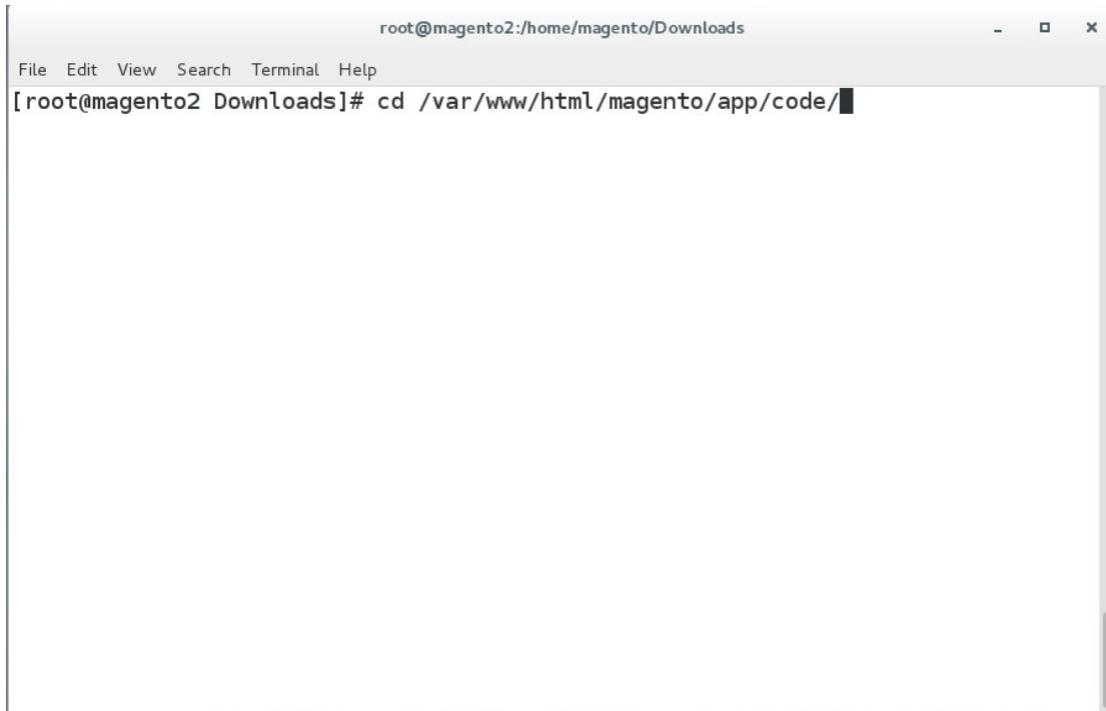
- 1022 3. Move to the *Tokenone10.zip* file to the Magento application code directory by entering the fol-
- 1023 lowing command:
- 1024 mv Tokenone10.zip /var/www/html/magento/app/code/



A screenshot of a terminal window titled "root@magento2:/home/magento/Downloads". The window has a standard Linux-style interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar. The main area of the terminal shows the command: [root@magento2 Downloads]# mv Tokenone10.zip /var/www/html/magento/app/code/. The command is partially typed, with the final part "/var/www/html/magento/app/code/" visible in the input field.

1025

-
- 1026 4. Change to the Magento application directory by entering the following command:
1027 cd /var/www/html/magento/app/code/

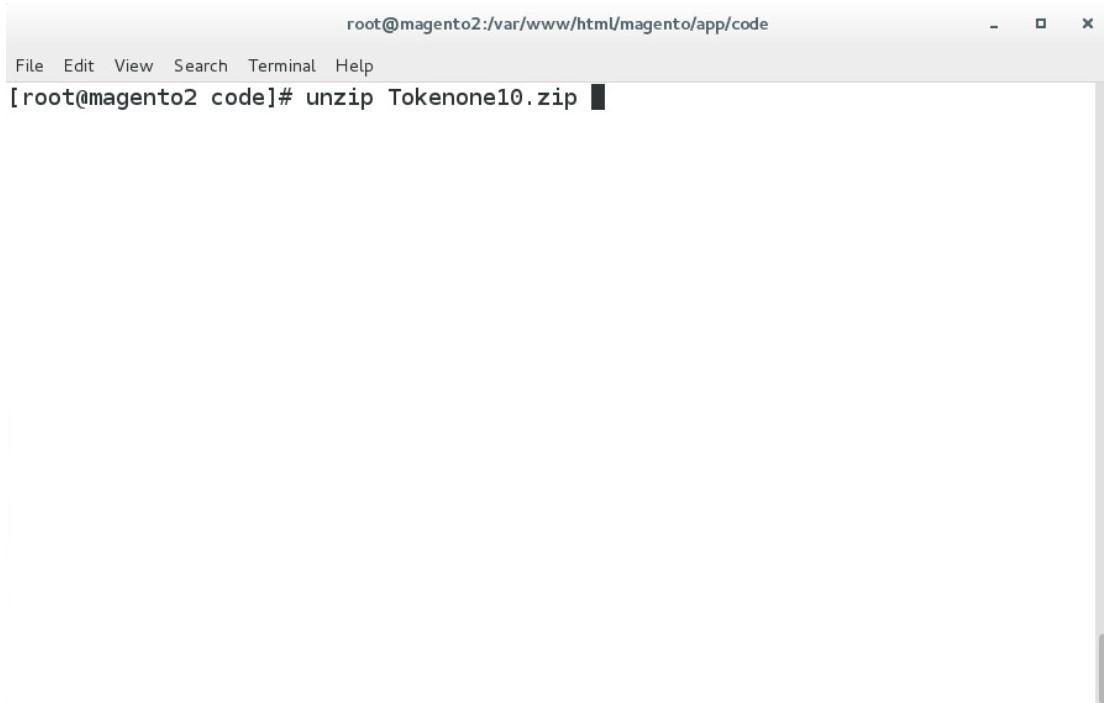


A screenshot of a terminal window titled "root@magento2:/home/magento/Downloads". The window has a standard Linux-style interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar. The main area shows a command line prompt: "[root@magento2 Downloads]# cd /var/www/html/magento/app/code/". The terminal is currently empty below the prompt.

1028

1029 5. Unzip the TokenOne zip file by entering the following command:

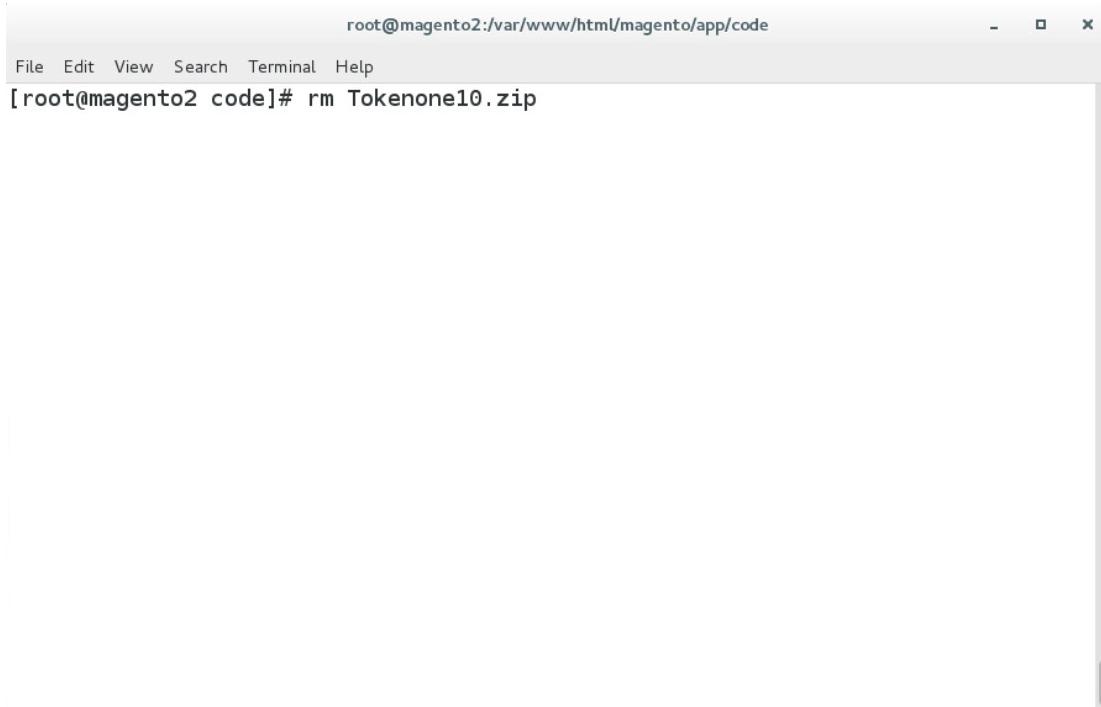
1030 unzip Tokenone10.zip



A screenshot of a terminal window titled "root@magento2:/var/www/html/magento/app/code". The window has a standard OS X-style title bar with icons for minimize, maximize, and close. The menu bar below it includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane of the terminal shows the command "[root@magento2 code]# unzip Tokenone10.zip" entered by the user. The terminal is running on a Mac OS X system.

1031

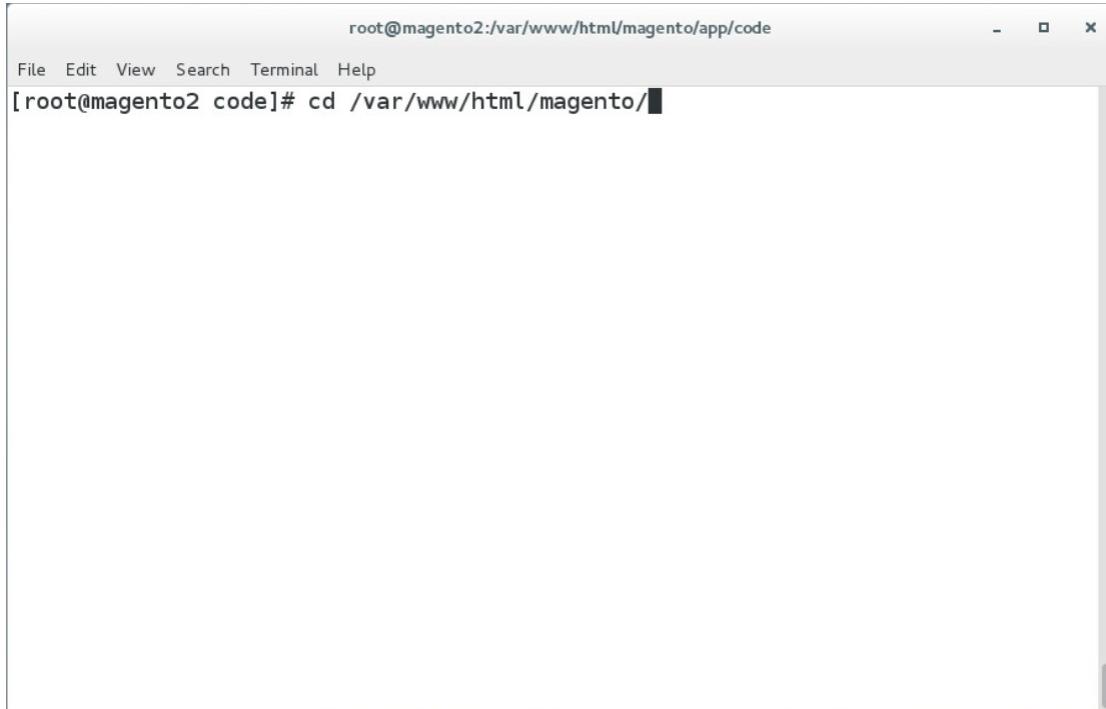
-
- 1032 6. Remove the zip file from the code directory by entering the following command:
1033 rm Tokenone10.zip



A screenshot of a terminal window titled "root@magento2:/var/www/html/magento/app/code". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the command "[root@magento2 code]# rm Tokenone10.zip" being typed in. The terminal is set against a light gray background.

1034

1035 7. Change to the Magento web server directory by entering the following command:
1036 cd /var/www/html/magento/



The screenshot shows a terminal window with the following details:

- Terminal title: root@magento2:/var/www/html/magento/app/code
- Menu bar: File Edit View Search Terminal Help
- Command line: [root@magento2 code]# cd /var/www/html/magento/

1037

1038 8. Enable the TokenOne module by entering the following command:

1039 php bin/magento module:enable Tokenone_TwoFactorAuth

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# php bin/magento module:enable Tokenone_TwoFactorAuth
```

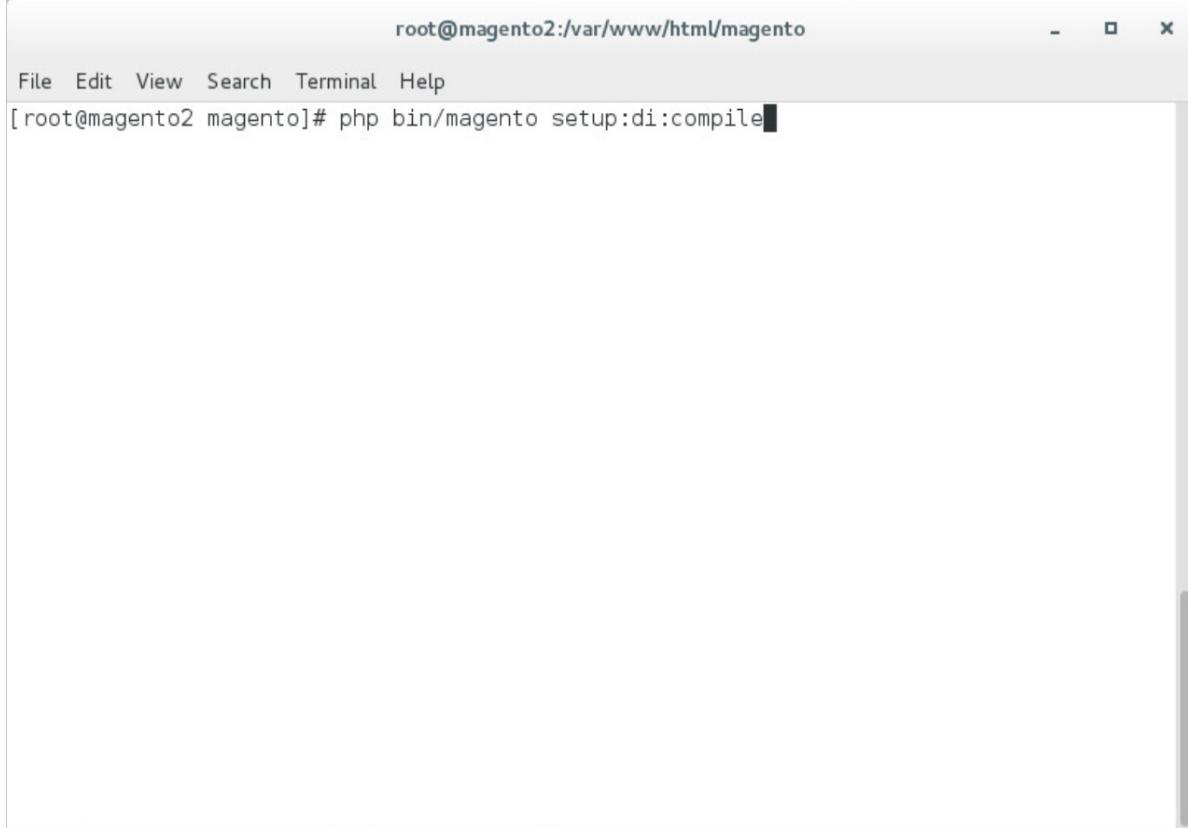
1040

- 1041 9. To upgrade Magento to reflect the newly enabled module, enter the following command:
1042 php bin/magento setup:upgrade

```
root@magento2:/var/www/html/magento
File Edit View Search Terminal Help
[root@magento2 magento]# php bin/magento setup:upgrade
```

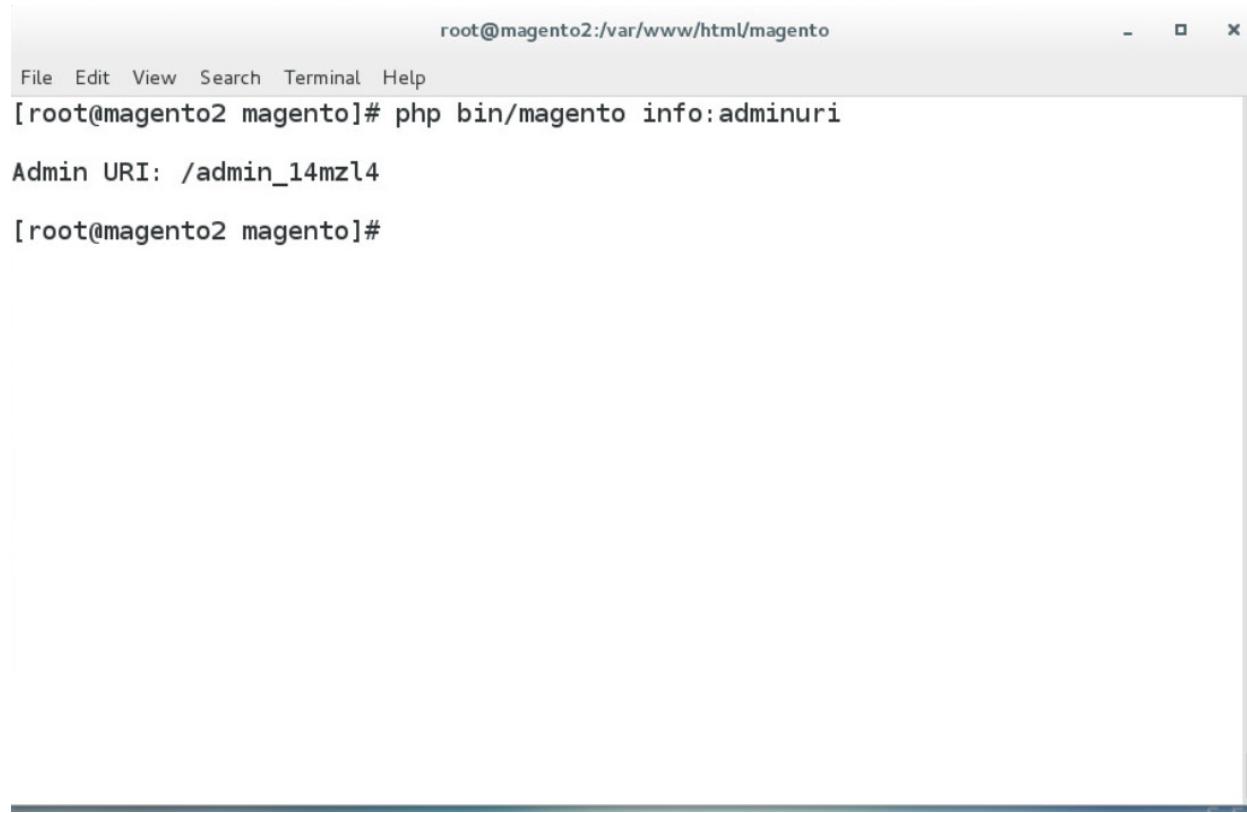
1043

- 1044 10. Recompile Magento to reflect the changes, by entering the following command:
1045 php bin/magento setup:di:compile



A screenshot of a terminal window titled "root@magento2:/var/www/html/magento". The window has standard OS X-style window controls at the top right. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane shows the command line: "[root@magento2 magento]# php bin/magento setup:di:compile". The cursor is positioned at the end of the command.

- 1046
1047 11. To find the Magento admin URI, enter the following command:
1048 php bin/magento info:adminuri



The screenshot shows a terminal window titled 'root@magento2:/var/www/html/magento'. The window has a standard OS X-style title bar with icons for minimizing, maximizing, and closing. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main pane displays the command 'php bin/magento info:adminuri' followed by its output: 'Admin URI: /admin_14mzl4'. The prompt '[root@magento2 magento]#' is shown at the bottom.

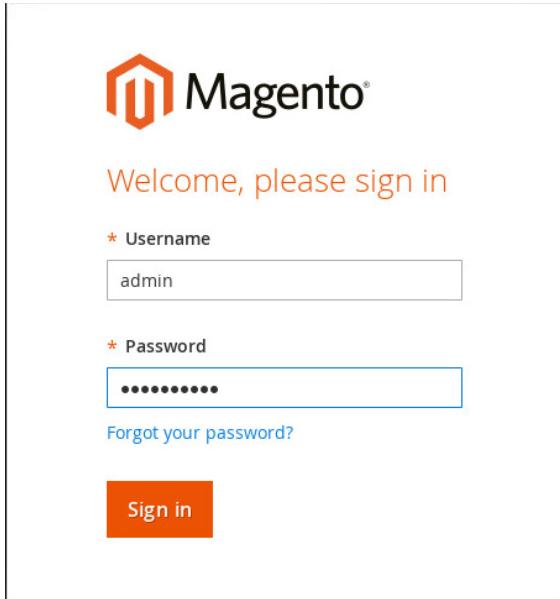
1049

1050 Note the URI that is output from the command. It will be used for TokenOne provisioning.

1051 2.5.4 TokenOne Provisioning

1052 Once TokenOne has been installed, administrators will be required to use TokenOne to log into the
1053 administration portal. The first time that an administrator logs into the portal, they will be required to
1054 provision and link their TokenOne authenticator with the system by using the following steps:

- 1055 1. Open a web browser and navigate to https://magento2.mfa.local/magento/admin_14mzl4.
1056 2. Sign into the admin portal.



1057

- 1058 3. Once the administrator has signed into the Magento admin portal, a TokenOne splash screen
1059 will appear with steps to create an account.

 **Magento®**

TokenOne Multi-Factor Authentication Registration

To complete the registration process, follow the steps below:

Step 1. Open the TokenOne application and click the Set Up Your Account Button

Step 2. Download the TokenOne application by searching for TokenOne in the app store for your phone

Step 3. Scan the QR code below*



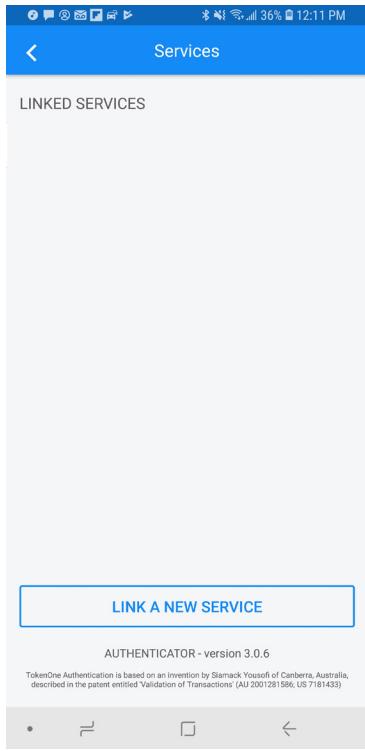
Step 4. To create your pin, click on the button below and follow instructions

Confirm

1060

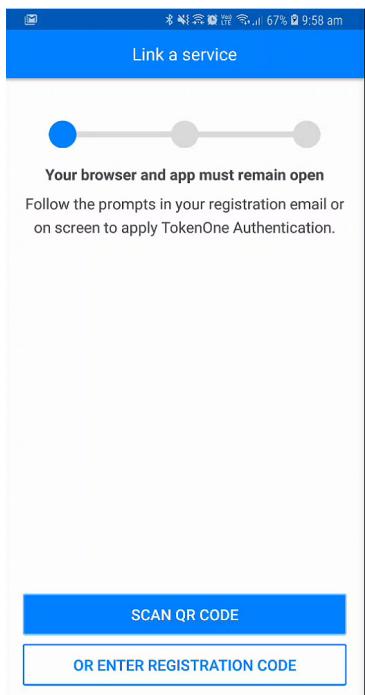
1061

4. Open the TokenOne mobile application and click **LINK A NEW SERVICE**.



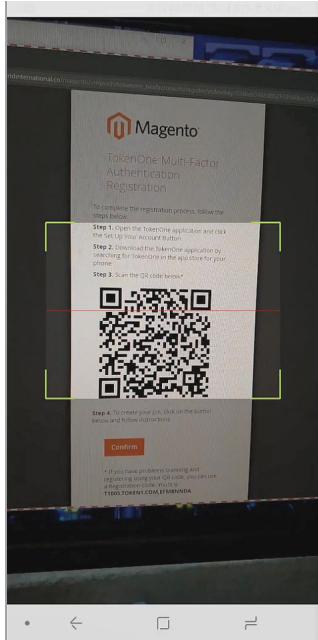
1062

1063 5. Click SCAN QR CODE.



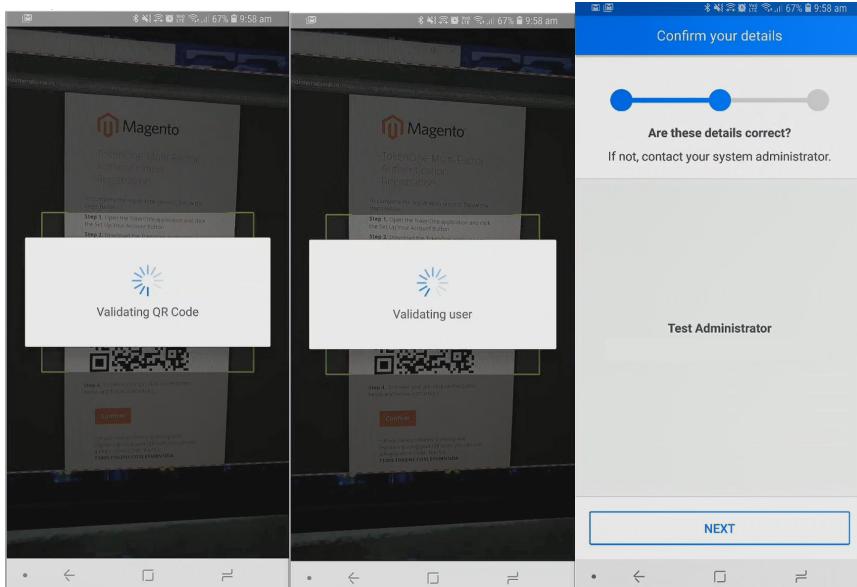
1064

1065 6. Capture the Quick-Response (QR) code that is displayed on the Magento site.



1066

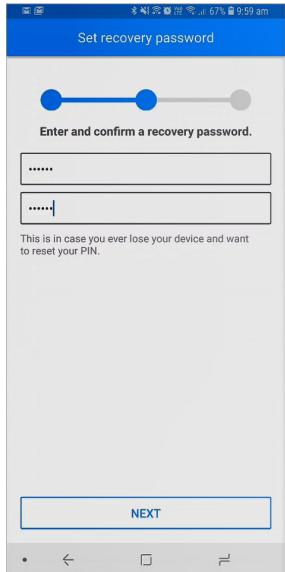
- 1067 7. Upon scanning the QR code, the phone will then be profiled and registered.
 1068 8. Follow the prompts on the smartphone to complete the registration.



1069

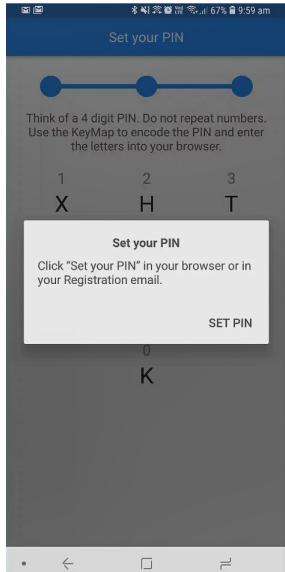
- 1070 9. Click **NEXT**.

1071 10. Create a recovery password for the account.



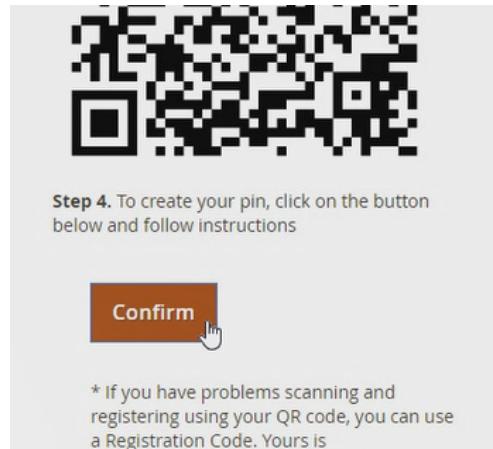
1072

1073 11. Click **NEXT**. Once the phone has been profiled and the account provisioned, you will be
1074 prompted to set your user PIN.



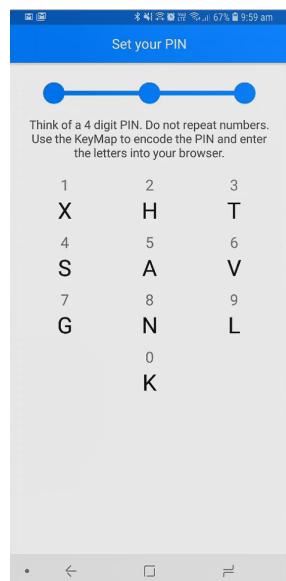
1075

1076 12. Click **SET PIN** on the phone, and click **Confirm** on your computer.



1077

- 1078 13. Use the KeyMap on the phone screen to encode your user PIN into a letter code. A KeyMap is
 1079 simply a sheet of 10 letters, each with a corresponding number (0 to 9). Match the numbers of
 1080 your PIN to the corresponding letters. This is your one-time letter code. For example, if your PIN
 1081 is 2610, then your one-time letter code is HVXK.



1082

- 1083 14. Enter the letters corresponding to your PIN into the Magento admin panel, and click **Submit**. Re-
 1084 peat the process to confirm your PIN.

Create your secret PIN

Step 1. Think of a hard to guess 4 to 6 digit PIN
Step 2. Use the KeyMap on your phone to convert the numbers to letters
Step 3. Enter your CODE (the letters representing your encoded PIN) below

* Enter your CODE:

Submit

Need support? Contact info@tokenone.com

Confirm your PIN

Please wait until a new KeyMap is displayed on your phone

Step 1. Do not do anything until a new KeyMap is displayed on your phone
Step 2. Use the KeyMap on your phone to convert the numbers to letters
Step 3. Enter your CODE (the letters representing your encoded PIN) below

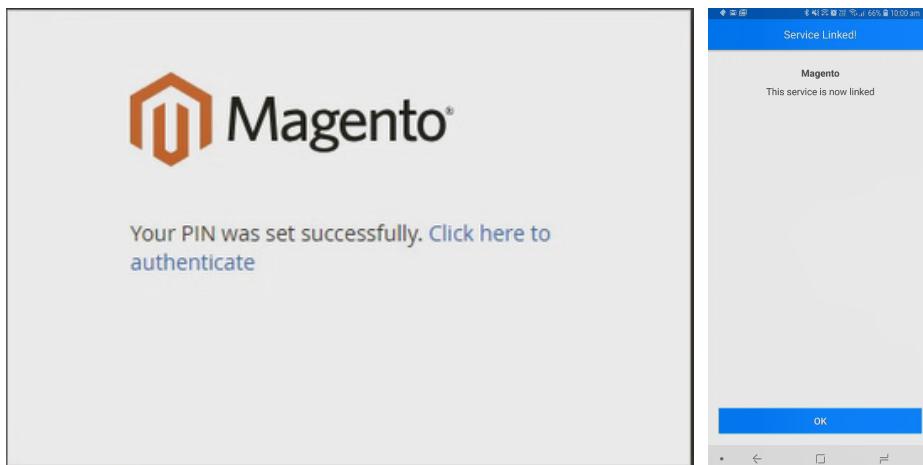
* Enter your CODE:

Submit

Need support? Contact info@tokenone.com

1085

- 1086 15. Do not turn off your phone during this process. Wait until the smartphone application indicates
 1087 that the account has been registered.



1088

1089 2.5.5 Administrator Login with TokenOne Authentication

1090 To log into the Magento administration portal by using TokenOne authentication, perform the following
 1091 steps:

- 1092 1. Open a web browser and navigate to https://magento2.mfa.local/magento/admin_14mzl4.
 1093 2. Sign into the admin portal.



The image shows the Magento sign-in page. It features the Magento logo at the top left. Below it, the text "Welcome, please sign in" is displayed. There are two input fields: one for "Username" containing "admin" and another for "Password" containing a series of asterisks. A blue link "Forgot your password?" is located below the password field. At the bottom is a large orange "Sign in" button.

1094

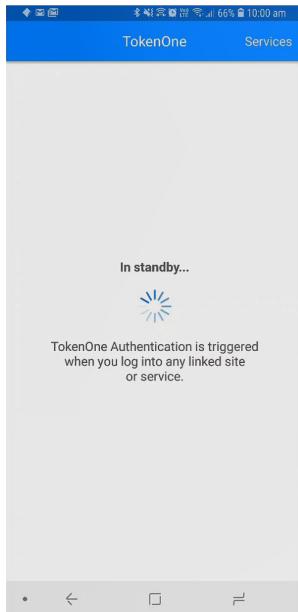
- 1095 3. Magento will prompt for the TokenOne **CODE**.



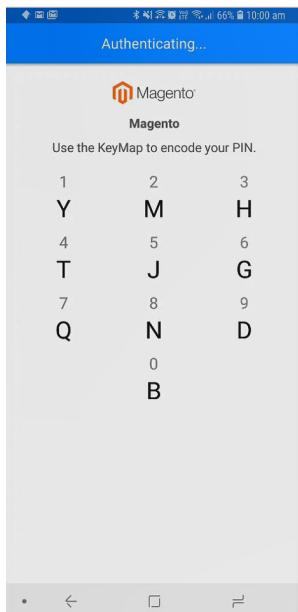
The image shows the Tokenone Auth page. It features the Magento logo at the top left. Below it, the text "Tokenone Auth" is displayed. A label "★ CODE:" is followed by a text input field. At the bottom is a large orange "Confirm" button.

1096

- 1097 4. Open the TokenOne mobile application on your smartphone.
- 1098 5. An **In standby...** screen will appear while the service verifies that you are using the correct regis-
1099 tered device.



- 1100
- 1101 6. Once your device is verified, a unique KeyMap will appear.



1102

- 1103 7. Match the numbers of your PIN to the corresponding letters. This is your one-time letter code.
1104 For example, if your PIN is **2610**, then your one time letter code is **MGYB**.
1105 8. Enter the letter code into the administration panel, and click **Confirm**.

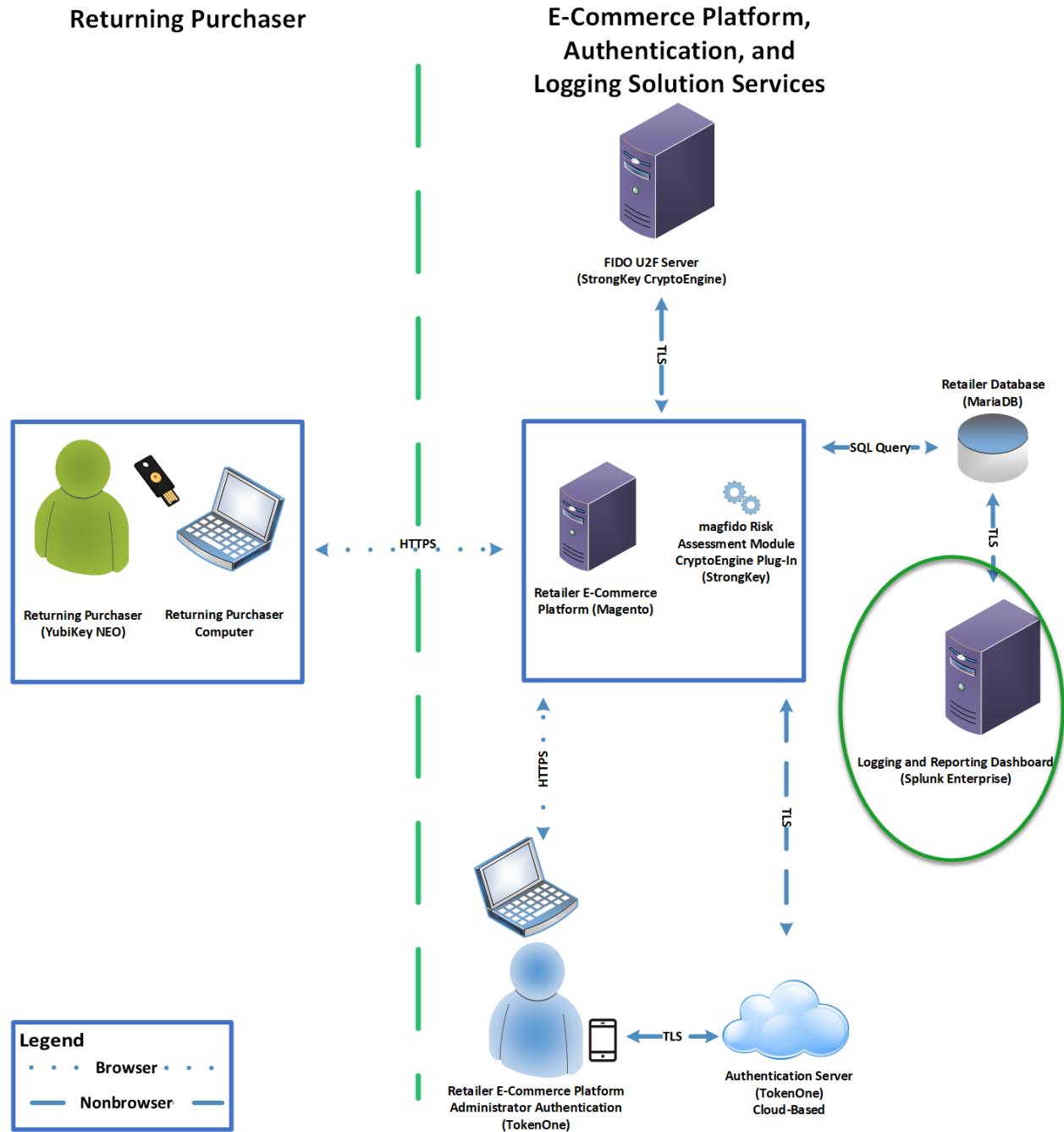


1106

1107 **2.6 Splunk Enterprise**

1108 This section provides installation and configuration guidance for Splunk's Enterprise product. Splunk
1109 Enterprise is used in both the *cost threshold* and *risk engine* example implementation builds to process
1110 and display authentication logging information. In addition to installing and configuring Splunk
1111 Enterprise and its supporting components, this section also provides step-by-step guidance on
1112 developing dashboard displays of the logged information. The locations of the Splunk components that
1113 are installed by using the instructions in this section are illustrated in [Figure 2-6](#) (circled in green).

1114 Figure 2-6 Splunk Enterprise Components



1115

1116 **2.6.1 Splunk Technologies Overview**

1117 Splunk [10] technologies enable computer log and data collection, parsing, and display. Splunk
1118 Enterprise [11], along with two enabling capabilities, was used in both example implementations:

- 1119 ▪ Splunk Enterprise [11], where data was collected, parsed, and displayed by using dashboards
- 1120 ▪ Splunk Universal Forwarder [12], which was installed on systems from which we collected data,
1121 forwarding the information to Splunk Enterprise
- 1122 ▪ Splunk DB Connect [13], which was used to import structured data for analysis, indexing, and
1123 visualization into Splunk Enterprise in the example implementation

1124 **2.6.2 Splunk Enterprise**

1125 **2.6.2.1 Overview**

1126 Splunk Enterprise [11] enables monitoring and analyzing data from multiple sources. Splunk Enterprise
1127 can receive data from many sources, and then respond to data queries and provide dashboard displays
1128 of the data that has been provided to it.

1129 For both example implementations, we used Splunk Enterprise to ingest a variety of log types from the
1130 retail e-commerce platform server. Once the data was collected by Splunk Enterprise, it could then be
1131 parsed and displayed by using prebuilt rules or custom criteria. For both example implementations, we
1132 displayed information as described in [Section 2.6.5](#).

1133 **2.6.2.2 Splunk Enterprise Requirements**

1134 System requirements required to support the use of Splunk Enterprise can be found here:
1135 <http://docs.splunk.com/Documentation/Splunk/6.6.1/Installation/Systemrequirements>.

1136 **2.6.2.3 Splunk Enterprise: Prepare for Installation**

1137 To prepare your environment for an on-premises installation, follow this guidance:

1138 Windows:

1139 <http://docs.splunk.com/Documentation/Splunk/6.6.1/Installation/PrepareyourWindowsnetworkforSplunkinstallation>

1141 **2.6.2.4 Splunk Enterprise Installation**

1142 You will need a Splunk account to download Splunk Enterprise. The account is free and can be set up at
1143 https://www.splunk.com/page/sign_up.

1144 Download Splunk Enterprise from https://www.splunk.com/en_us/download/splunk-enterprise.html.
1145 Splunk Enterprise was installed on a Windows instance. The installation instructions can be found here:
1146 <http://docs.splunk.com/Documentation/Splunk/6.6.1/Installation/InstallonWindows>.

1147 **2.6.3 Splunk Universal Forwarder**

1148 ***2.6.3.1 Splunk Universal Forwarder Overview***

1149 The Splunk Universal Forwarder collects data to be used by Splunk Enterprise. Splunk Universal
1150 Forwarder allows Splunk Enterprise to collect data from remote sources and send it for indexing. To use
1151 this capability, Splunk Universal Forwarder must be installed on each system from which you want to
1152 collect data.

1153 We used Splunk Universal Forwarder to collect data from Magento and forward it to Splunk Enterprise.
1154 Once the data was delivered to Splunk Enterprise, the data provided by the Splunk Universal Forwarder
1155 was used to analyze purchaser authentication trends and to populate the dashboard displays.

1156 ***2.6.3.2 Splunk Universal Forwarder Requirements***

1157 System requirements required to support the use of Splunk Universal Forwarder can be found here:
1158 <http://docs.splunk.com/Documentation/Forwarder/6.6.1/Forwarder/Systemrequirements>.

1159 ***2.6.3.3 Splunk Universal Forwarder: Prepare for Installation***

1160 Before you can forward data to Splunk Enterprise, you must enable forwarding and receiving on Splunk
1161 Enterprise. Instructions can be found here:
1162 <http://docs.splunk.com/Documentation/Forwarder/6.6.1/Forwarder/EnableaReceiver>.

1163 ***2.6.3.4 Splunk Universal Forwarder: Installation***

1164 The Splunk Universal Forwarder can be installed on different operating system platforms. The following
1165 subsections provide instructions for installing the Splunk Universal Forwarder on both Linux and
1166 Windows.

1167 ***2.6.3.4.1 Installing Splunk Universal Forwarder on Linux***

1168 Detailed Splunk Universal Forwarder installation instructions can be found here:
1169 http://docs.splunk.com/Documentation/Forwarder/6.6.1/Forwarder/Installanixuniversalforwarder#Instal l_all_the_universal_forwarder_on_Linux.

1171 The following steps are an abridged version of the preceding installation link:

- 1172 1. You will need a splunk.com account to download the Splunk Universal Forwarder on Linux. Ac-
1173 count setup is free and can be done here: https://www.splunk.com/page/sign_up.
- 1174 2. Once you have an account, the Splunk Universal Forwarder for Linux is free and can be down-
1175 loaded from here: http://www.splunk.com/en_us/download/universal-forwarder.html.
- 1176 3. Having the latest operating system version is recommended for installations. For both example
1177 implementations, we used the latest CentOS OS version 2.6+ kernel Linux distributions (64-bit).
1178 For the example implementation, we installed on CentOS by selecting the file that ends in .tgz
1179 and placed it on the target Linux machine. This is an example:

1180 `splunkforwarder-7.0.1-2b5b15c4ee89-linux-x86_64.tgz`

- 1181 4. Untar the file downloaded to the opt/ directory:

1182 `tar zxvf <splunk_package_name.tgz> -C /opt`

- 1183 5. Change to the /opt/splunkforwarder/bin directory:

1184 `cd /opt/splunkforwarder/bin`

- 1185 6. Start the universal forwarder:

1186 `./splunk start`

- 1187 7. Enable boot start of the universal forwarder:

1188 `./splunk enable boot-start`

1189 2.6.3.4.2 Configure Splunk Forwarder on Linux

1190 More information about adding a forwarder can be found at

1191 <http://docs.splunk.com/Documentation/Forwarder/6.6.1/Forwarder/Configuretheuniversalforwarder>.

- 1192 1. Change to the /opt/splunkforwarder/bin directory:

1193 `cd /opt/splunkforwarder/bin`

- 1194 2. Run script to configure the forwarder to connect to the Splunk Enterprise server:

1195 `./splunk add forward-server loghost:7777 -auth admin:change`

1196 2.6.3.4.3 Installing Splunk Universal Forwarder on Windows

- 1197 1. You will need a splunk.com account to download the Splunk Universal Forwarder on Windows.

1198 An account is free and can be set up here: https://www.splunk.com/page/sign_up.

- 1199 2. Once you have an account, the Splunk Universal Forwarder for Windows is free and can be
1200 downloaded from here: http://www.splunk.com/en_us/download/universal-forwarder.html.

1201 3. You want the latest version for operating system version Windows (64-bit). Because this down-
1202 load will be installed on Windows, select the file that ends in .msi. This is an example:

1203 `spunkforwarder-7.0.0-00f5bb3fa822-x64-release.msi`

1204 2.6.4 Splunk DB Connect

1205 Splunk DB Connect facilitates database information imports, exports, lookups, and multiple data source
1206 combinations [13], [14].

1207 2.6.4.1 Overview

1208 Splunk DB Connect provides a solution for integrating database information with Splunk Enterprise
1209 queries and reports. It allows for structured data-collection from databases, which can be leveraged in
1210 analysis.

1211 Splunk DB Connect was used to import structured data from Magento's MySQL database instance. This
1212 enabled us to leverage information in the database within the Splunk Enterprise deployment.

1213 2.6.4.2 Splunk DB Connect Requirements

1214 Splunk DB Connect requires that the Java Runtime Environment (JRE) is installed on the Splunk
1215 Enterprise search head. The JRE can be installed from here:

1216 <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>.

1217 You must install a driver for the database that you are planning to connect to the Splunk DB Connect
1218 application. Splunk DB Connect supports a list of drivers that can define other databases. MariaDB is not
1219 included in the list of predefined databases. As MariaDB is a branch of MySQL, we downloaded the
1220 MySQL Java Connector from the following location ([Section 2.6.4.4](#), Step 6 provides installation
1221 directions for the Java Connector): <https://dev.mysql.com/downloads/connector/j/>.

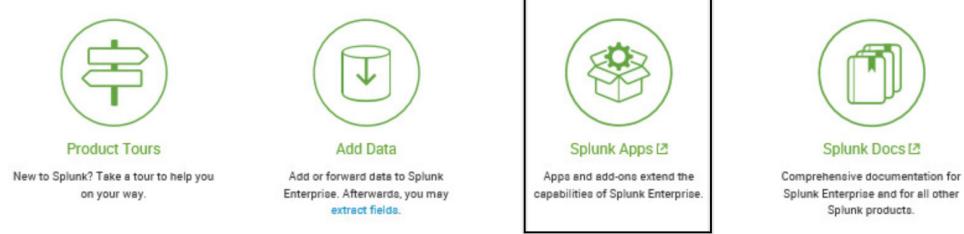
1222 2.6.4.3 Splunk DB Connect Installation

1223 This section describes the steps required to install the Splunk DB Connect application onto your single-
1224 instance deployment of Splunk. Additional guidance can be found here:

1225 <https://docs.splunk.com/Documentation/DBX/3.1.2/DeployDBX/AboutSplunkDBConnect>.

- 1226 1. Navigate to the Splunk Enterprise home page, and click the **Splunk Apps** icon.

Explore Splunk Enterprise



1227

- 1228 2. Type “db connect” into the search bar to locate the Splunk DB Connect application.

Browse More Apps

db connect X

CATEGORY

- DevOps
- IT Operations
- Security, Fraud & Compliance
- Business Analytics
- IoT & Industrial Data
- Utilities

1229

- 1230 3. Once the **Splunk DB Connect** application is located, click **Install**.

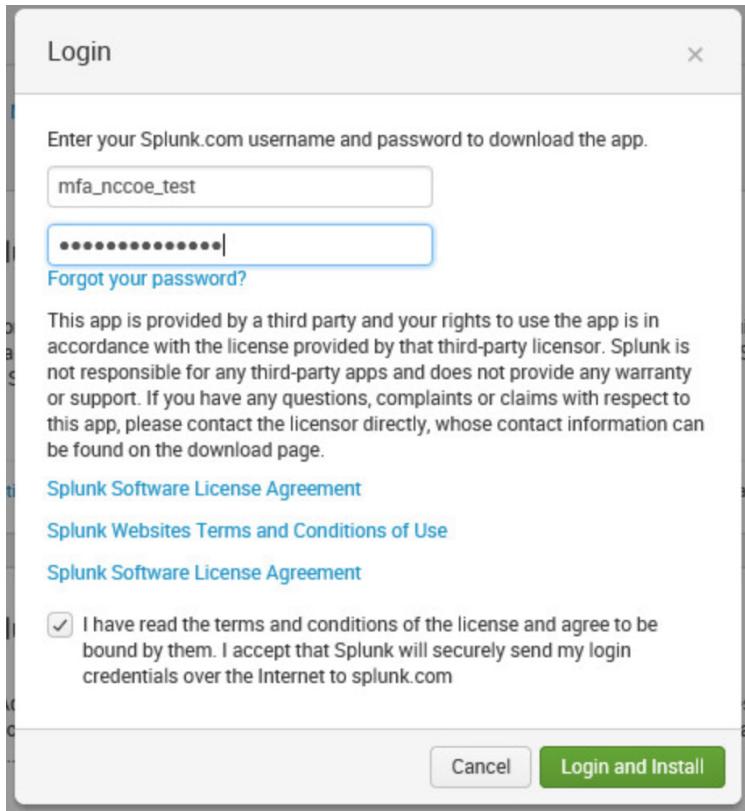
DBX Splunk DB Connect Install

Splunk DB Connect is the best solution for working with databases from Splunk. It can help you quickly integrate structured data sources with your Splunk real-time machine data collection. Supports DB2/Linux, Informix, MemSQL, MySQL, AWS Aurora, Microsoft SQL Server, Oracle, PostgreSQL, AWS RedShift, SAP SQL Anywhere, Sybase ASE, Sybase IQ, and Ter... [More](#)

Category: [Utilities](#), [Business Analytics](#) | Author: [Splunk Inc.](#) | Downloads: 60282 | Released: 3 years ago | Last Updated: 5 months ago | [View on Splunkbase](#)

1231

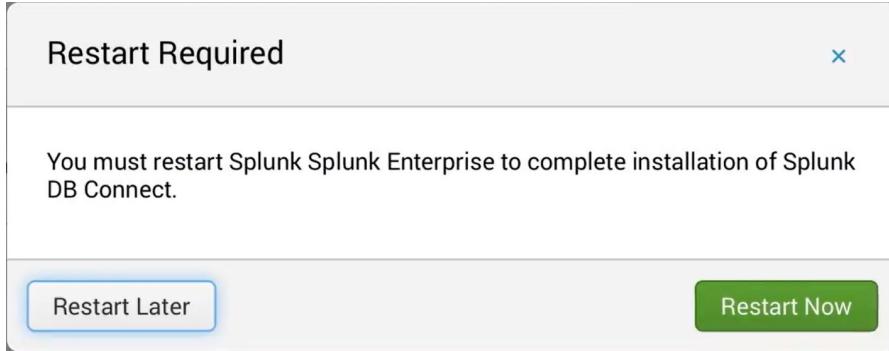
- 1232 4. Log in and accept the terms and conditions by using your splunk.com user account and credentials (not the Splunk Enterprise instance credentials) and then by clicking **Login and Install**.
- 1233



1234

1235

5. Click **Restart Now**.



1236

1237
1238

6. Log in after reboot, with the Splunk Enterprise instance credentials that were created during the installation of Splunk Enterprise.

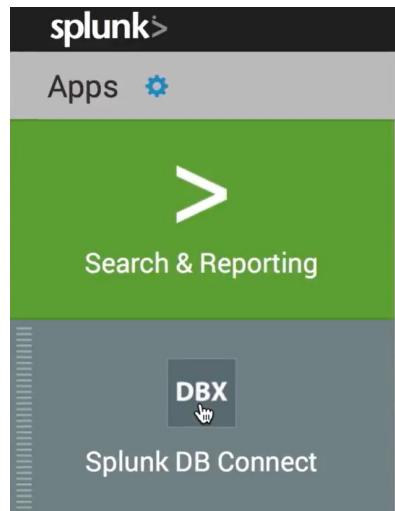


1239

2.6.4.4 Setup

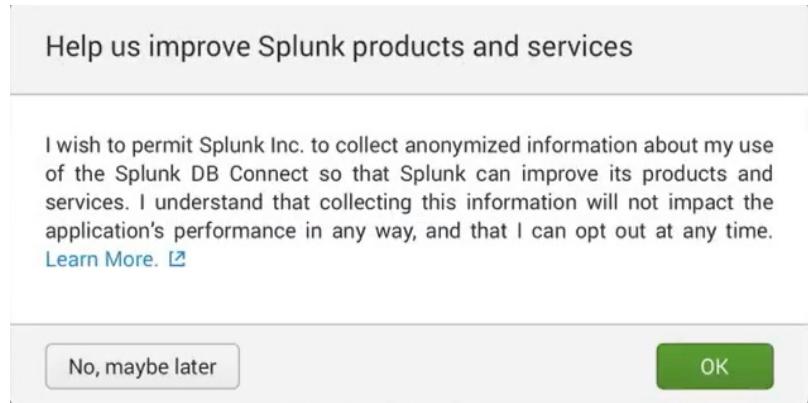
1241 This section describes the initial setup process that will follow the installation of Splunk DB Connect.

- 1242 1. On the home page, navigate to **Splunk DB Connect** in the **Apps** sidebar.



1243

- 1244 2. Select whether to send Splunk information about your use of Splunk DB Connect.



1245

- 1246 3. Click **Setup** to begin the configuration process.

Welcome to DB Connect!



Connect

Link to your databases



Transport

Retrieve, index and export your data



Transform

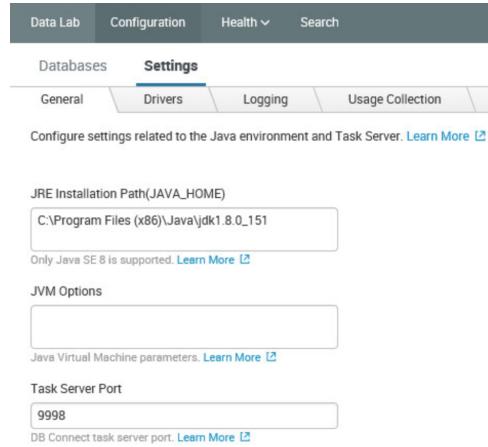
Enrich and work with your data

DB Connect requires some basic settings to work properly. [Skip Setup](#)

[Setup](#)

1247

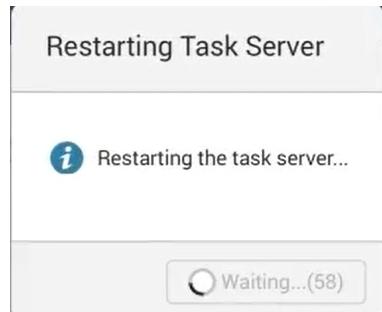
- 1248 4. Specify the **JRE Installation Path (JAVA_HOME)**.



1249

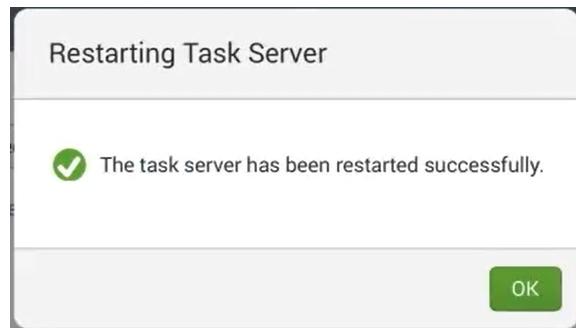
1250 a. Click **Save** to confirm general configurations.

1251 b. Task server restart will occur.



1252

1253 c. Once the restart completes, click **OK**.



1254

1255 5. Proceed to set up drivers for the database in the **Drivers** tab: **Configuration > Settings > Drivers**.

1256 6. Search for the database that you are using.

Driver Name	Installed	Version
MySQL	No	5.1

- 1257
- 1258 a. If your driver is not installed, Splunk DB Connect will show **No** for **Installed**. If that is the case, perform Step i below to move the connector into a new directory to enable configuring Splunk DB Connect.
- 1259 i. Move the MySQL Java Connector downloaded in [Section 2.6.4.2](#) to the following directory:
- 1260 `C:\Program Files\Splunk\etc\apps\splunk_app_db_connect\drivers`
- 1261 b. To specify a database that isn't predefined, follow the Splunk documentation located here: <https://docs.splunk.com/Documentation/DBX/3.1.2/DeployDBX/AboutSplunkDB-Connect>.
- 1262 7. Click **Reload**. The status of the driver should reflect that it was installed.

Driver Name	Installed	Version
MySQL	Yes	5.1

- 1263
- 2.6.4.5 Creating Identities**
- 1264 Before connecting Splunk DB Connect to your database, an identity is needed to establish the connection. This section details creating an identity that leverages database credentials, which will be used by Splunk DB Connect to access your database.
- 1265 1. Navigate to the **Identities** tab: **Configuration > Databases > Identities**.
- 1266 2. Click **New Identity**.

1275

1276 3. Configure the **Settings** for your **New Identity**.

1277

1278 a. Specify a unique **Identity Name**.1279 b. Enter the **Username** and **Password** that are used to access your database.1280 c. Click **Save**.

1281 4. You will now see the new identity that you created, listed in the table of identities.

1282

1283 ***2.6.4.6 Creating Connections***1284 This section details how to create a database connection for Splunk DB Connect to use. This provides the
1285 information that the software needs to connect to your remote database.1286 1. Navigate to the **Connections** tab: **Configuration > Databases > Connections**.

1287 2. Click **New Connection**.

The screenshot shows the Splunk DB Connect interface. At the top, there are tabs for Data Lab, Configuration, Health, and Search. On the right, it says "Splunk DB Connect". Below these, there are two main tabs: "Databases" (which is selected) and "Settings". Under "Databases", there are two sub-tabs: "Connections" (selected) and "Identities". A search bar at the top says "Search by Connection Name". To the right of the search bar is a link "A database connection object contains the necessary information for connecting to a remote database. Learn More" and a green "New Connection" button.

1288

1289 3. Configure the **Settings** for your **New Connection**.

The screenshot shows the "New Connection" configuration page. At the top, there are tabs for "Settings" (selected) and "Permissions". The "Connection Name" field is filled with "Magento_DB". The "Identity" dropdown is set to "magento_users". The "Connection Type" dropdown is set to "MySQL". The "Timezone" dropdown is set to "US/Eastern -05:00". Below the timezone dropdown, there is a note: "The time zone used by DB Connect to read time-related fields. By default the JVM time zone setting is used." followed by a "Learn More" link. On the right side, there are "Cancel" and "Save" buttons. At the bottom right, there is a note: "Activate Windows Go to System in Control Panel to activate Windows." followed by a dropdown arrow.

1290

- 1291 a. Uniquely name your connection in the **Connection Name** field.
- 1292 b. Select the **Identity** created in [Section 2.6.4.5](#).
- 1293 c. Select the type of database being connected, in the **Connection Type** field.
- 1294 d. Specify the **Timezone**.

1295 4. Configure the **JDBC URL Settings**.

JDBC URL Settings	JDBC URL Preview
Host	<input type="text" value="jdbc:mysql://magento.mfa.local:3306/magento"/>
Port	<input type="text" value="3306"/>
Default Database	<input type="text" value="magento"/>
<p>The usage and meaning of this parameter varies between database vendors. Learn More</p>	
<input type="checkbox"/> Enable SSL	This is a DB driver flag and may not be supported by all JDBC drivers. Learn More
Advanced Settings	
<input type="checkbox"/> Read Only	Use a read-only database connection to ensure that data cannot be altered. This is a DB driver flag and not guaranteed to work for all drivers.
Fetch Size	<input type="text" value="Optional"/>
<p>The number of rows to return at a time from the database.</p>	

1296

- 1297 a. Enter the database's hostname in the **Host** field.

1298 b. Specify the **Port** that your database uses for remote connections.

1299 c. Specify the **Default Database** to be used.

1300 d. Click **Save**.

1301 Note: If you receive an error when attempting to save the connection, be sure to check
1302 that the database to which you are attempting to connect is configured for remote
1303 connections.

1304 5. You will now see the new connection that you created, listed in the table of connections.

Data Lab Configuration Health Search Splunk DB Connect

Databases Settings

Connections Identities

Search by Connection Name A database connection object contains the necessary information for connecting to a remote database. [Learn More](#)

New Connection

Connection Name	Identity	Connection Type	App	Status	Sharing	Actions
Magento_DB	magento_users	MySQL	Splunk DB Connect	<input checked="" type="button"/> Enabled	App Permissions	Edit Clone Delete

1305

1306 2.6.4.7 Creating Inputs

- 1307 This section details how to ingest data from your database by using inputs. We demonstrated the
1308 creation of an input that pulled customer account information from the Magento database.

1309 1. Navigate to the **Inputs** tab: **Data Lab > Inputs**.

1310 2. Click **New Input**.

The screenshot shows the Splunk Data Lab interface with the 'Inputs' tab selected. A modal dialog titled 'New Input' is open. At the top of the dialog is a green progress bar with a green dot at the start. Below it is a button labeled 'Set SQL Query'. The main area of the dialog is titled 'Choose Table' and contains the following fields:

- Connection:** Magento_DB
- Catalog:** magento
- Schema:** Select...
- Table:** Customer_entity

To the right of these fields is a vertical sidebar with the title 'SQL' at the top, followed by a large black rectangular area.

1311

1312 3. Choose the table for your **New Input**.

The 'New Input' dialog is still open, showing the 'Choose Table' section. The 'Table' field now contains 'Customer_entity'. The sidebar on the right is visible but mostly blacked out.

1313

1314 a. Select the **Connection** created in [Section 2.6.4.6](#).

1315 b. Select the Default Database created in [Section 2.6.4.6](#), Step 4c, as the **Catalog**.

1316 c. Search for and select the **Table** from which input is to pull data. We selected the **Cus-**
1317 **tomer_entity** table.

1318 4. Preview the data.

Preview Data

SQL Editor

```
1  SELECT * FROM `magento`.`customer_entity`
```

entity_id	disabled	enable_auto_group_change	email	entity_id	failures_num	first_failure
1	0	1973-12-15	roni_cost@example.com	1	0	
2	0		nccoe@example.com	2	3	2018-01-13:22:31
	0		a@a.com	3	0	A
3	0		jdoe@mfa.test.com	4	0	G W

1319

1320 5. Click **Execute SQL** to review the results of the query.

1321 6. Select the **Input Type**.

Settings

Template

Select...

Input Type

Batch Rising

1322

1323 **Batch or Rising:** **Batch** indexes all of the table's data every time that it runs, whereas **Rising** uses
 1324 a checkpoint to update the data that it collects from the table. We selected **Rising**.

1325 7. Configure the settings for the Rising input type.

Rising Column

entity_id

Checkpoint Value

0

Timestamp

Current Index Time Choose Column

Query Timeout

30

Enter the number of seconds to wait for the query to complete. The default is 30 if you leave it blank.

1326

- 1327 a. Specify the column of your table to be used as the **Rising Column**. We selected **entity_id**.
- 1329 b. Enter the **Checkpoint Value** of the entry where you want your Rising Input to begin updating. This will dynamically update as the query is executed over time. We entered **0** to begin input at the first entity created.
- 1332 c. Select the **Timestamp** for Splunk to index this data. We selected **Current Index Time**.
- 1333 d. **Query Timeout:** Enter the number of seconds to wait for the query to complete. We entered **30**.
- 1335 8. Click **Next**.

New Input

Set SQL Query Set Properties Complete

< Next Cancel

Choose Table

Preview Data

Connection

Magento_DB

SQL Editor

Format

```
1 SELECT * FROM `magento`.`customer_entity` WHERE entity_id > ?
2 ORDER BY entity_id ASC
```

1336

- 1337 9. Set Properties for the New Input.

New Input

Set SQL Query Set Properties Complete

Basic Information

Name: magento_customer_entity

Description: Customer info

Application: Splunk DB Connect

Parameter Settings

Max Rows to Retrieve: 0
Enter the maximum number of rows to retrieve with each query. If you set this to 0 or leave it blank, it will be unlimited. [Learn More](#)

Fetch Size: 300
Enter the number of rows to return at a time from the database. The default is 300 if you leave it blank.

Execution Frequency: 30
Enter the number of seconds or a valid cron expression e.g. 0 18 * * * (every day at 6PM).

Metadata

Host: Optional
The host defined on the connection will be used if you leave it blank.

Source: Optional
The input name will be used if you leave it blank.

Source Type: mysqld-5

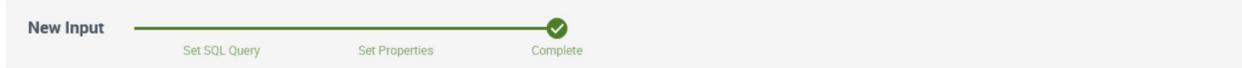
Index: main

1338

- 1339 a. Enter a unique **Name** for the input. We named our instance **magento_customer_entity**.
- 1340 b. Enter a **Description** for the type of data being input from the table.
- 1341 c. Select the **Application** context. We selected **Splunk DB Connect**.
- 1342 d. Enter the **Max Rows to Retrieve** with each query. We entered the default, **0**.
- 1343 e. Enter the **Fetch Size**. This specifies the number of rows to be returned with each input query. We entered the default, **300**.
- 1344 f. Enter the **Execution Frequency**. This specifies how frequently, in seconds, to execute the query for this input. We entered **30**.
- 1345 g. Enter a **Source Type** for the data being queried by this input. Note: This can be pre-defined, or a new type can be created in this field. We entered the predefined **mysqld-5**.
- 1346 h. Select the **Index** field, and enter **main**.

1350 i. Click **Finish**.

1351 10. The following screen will appear upon completion. Click **Back to List**.



Done!

Name: magento_customer_entity
Connection: Magento_DB
App: splunk_app_db_connect
Status: Enabled

Add More Back to List

Activate Windows
Go to System in Control Panel to activate Windows.

1352

1353 11. You will now see the new input that you created, listed in the table of inputs.

The screenshot shows the Splunk Data Lab interface with the 'Inputs' tab selected. A table lists the input configuration. The first row shows:

Name	Connection	Template	App	Status	Actions
magento_customer_entity	Magento_DB	-	Splunk DB Connect	Enabled	Edit Clone Find Events Delete

1 input in total.

1354

2.6.4.8 Creating Database Lookups

1355 This section describes creating a new database lookup. Database lookups allow you to extend the data being input from your external database into the Splunk Search Processing Language (SPL) queries. It allows events gathered from logs to be correlated with the information pulled from your database. This example correlates the entity_id returned in SPL queries to user emails stored in the database.

1360 1. Navigate to the **Lookups** tab: **Data Lab > Lookups**.

1361 2. Click **New Lookup**.

The screenshot shows the Splunk Data Lab interface with the 'Lookups' tab selected. A table lists the lookup configuration. The first row shows:

Name	Connection	App	Status	Actions
------	------------	-----	--------	---------

A green 'New Lookup' button is visible on the right side of the interface.

1362

1363 3. Navigate to **Set Reference Search**, and select the field of interest to be mapped to the lookup.

	entity_id	_raw	_time
1	4	2018-01-10 06:27:01 350, entity_id="4", website_id="1", email="jdoe@mfa.test.com", group_id="1", store_id="1", created_at="2018-01-10 06:29:28.0", updated_at="2018-01-10 06:29:28.0", is_active="1", disable_auto_group_change="0", created_in="Default Store View", firstname="John", lastname="Doe", password_hash="416bffe7d76f626002c9150b4f8769f03df2b49a739267edbae521d08d609f1.xbRPwaCnpB6RLeAHmVv78p30Mxe8MjxW.1", rp_token="c4daa220505e7be606a364f5ab6fa194", rp_token_created_at="2018-01-10 14:29:28.0"	2018-01-10 09:27:01.350
2	3	2018-01-09 10:12:01 065, entity_id="1", website_id="1", group_id="1", store_id="1", created_at="2018-01-05 11:52:31.0", updated_at="2018-01-05 11:55:51.0", is_active="1", disable_auto_group_change="0", created_in="Default Store View", firstname="A", lastname="A", password_hash="f0c0d5093db1cf96b92fb1a5bda27a5c3c2992238c779ccb02458dc2b27aa2d.wewhTj51SE5V0OaqkybFP077Fc2MN1Z.1", rp_token="3ce45c41c48d6f012a31eeff090752ff6", rp_token_created_at="2018-01-05 19:52:32.0", failures_num="0"	2018-01-09 13:12:01.065
3	2	2018-01-09 10:12:01 064, entity_id="2", website_id="1", email="nccoe@example.com", group_id="1", store_id="1", created_at="2017-10-31 12:14:33.0", updated_at="2018-01-03 09:01:12.0", is_active="1", disable_auto_group_change="0", created_in="Default Store View", firstname="nccoe", lastname="nccoe", password_hash="db9f2ab196e6fe0cc1e9ed7e3abd6653139407d44b3ba07c003a53a8c7568NdHJa7rdC4YSRHk0H2EdohILQWIV.1", rp_token="54dfae2a29504f2cb364ef762449fe3bf", rp_token_created_at="2017-10-31 19:14:34.0", default_shipping="2", failures_num="6", first_failure="2018-01-02 07:07:36.0"	2018-01-09 13:12:01.064
4	1	2018-01-09 10:12:01 044, entity_id="1", website_id="1", email="roni_cost@example.com", group_id="1", store_id="1", created_at="2017-10-18 14:17:55.0", updated_at="2017-10-18 14:18:56.0", is_active="1", disable_auto_group_change="0", created_in="Default Store View", firstname="Veronica", lastname="Costello", dob="1973-12-15", password_hash="a1dbfdc62f5d07572d9f6838f8febf86a9eaecdcd0d7c43a02d5905daf7ccb:c3abk1FRos18bIUPCznWml0xJ6OoAp:1", rp_token="c4dfb17b70d8f8fc4f23ff7890e7ff68bc", rp_token_created_at="2017-10-18 21:17:56.0", default_billing="1", default_shipping="1", gender="2", failures_num="0"	2018-01-09 13:12:01.044

1364

1365 a. We entered a new **Search**.

1366 b. Click **Next**.

1367 4. Navigate to **Set Lookup SQL**.

	confirmation	created_at	created_in	default_billing	default_shipping	disable_auto_group_change
1	2017-10-18 14:17:55.0	Default Store View		1	1	0
2	2017-10-31 12:14:33.0	Default Store View			2	0
3	2018-01-05 11:52:31.0	Default Store View				0
4	2018-01-10 06:29:28.0	Default Store View			3	0

1368

1369 a. Specify a **Connection** by using information from the connection, which was created in [Section 2.6.4.6](#).

1370 b. Specify the **Catalog**.

- 1372 c. Enter the **Table**.
- 1373 d. Click **Execute SQL** to view the results of the query created.
- 1374 e. Click **Next**.
- 1375 5. Navigate to **Field Mapping**.

New Lookup

Set Reference Search Set Lookup SQL **Field Mapping** Set Properties Complete < Next Cancel

Search Fields Mapping

Map your selected search results fields to table columns.

Search Fields	Match	Table Columns
entity_id	→	entity_id

Add Search Field ▾

Lookup Fields

Add your table columns as new Splunk fields.

Table Columns	AS	Aliases
email	→	email

Add Column ▾

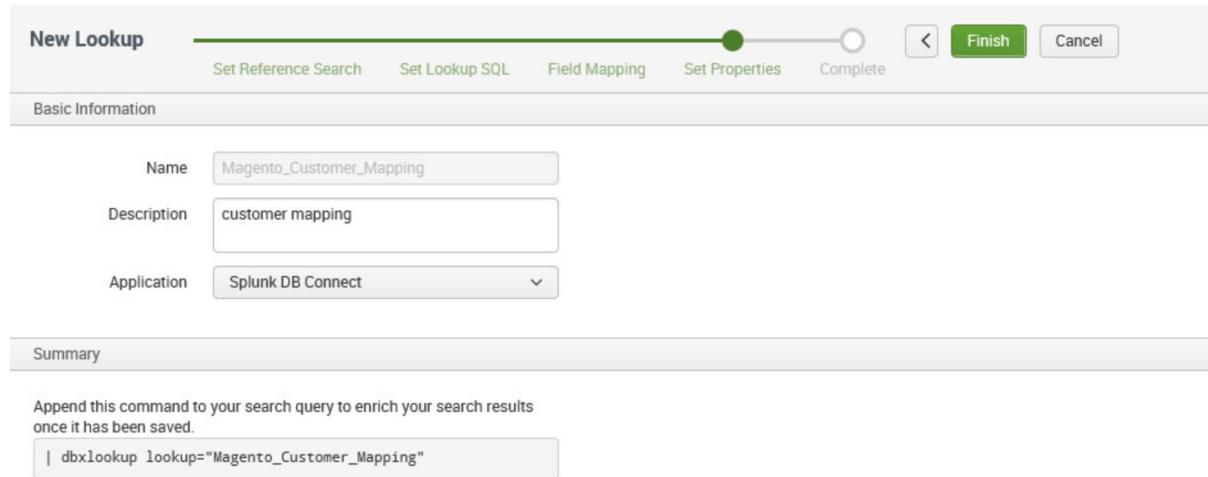
Preview Results

Preview lookup results with the following SPL

```
(...) | dbxlookup connection="Magento_DB" query="SELECT * FROM `magento`.`customer_entity`" "entity_id" AS "entity_id" OUTPUT "email" AS "email"
```

Open In Search ↗

- 1376
- 1377 a. Click **Add Search Field**.
- 1378 b. Select the **Search Fields** to be mapped to the database. We selected **entity_id**.
- 1379 c. Select the **Table Columns** to which the field maps in the database. We selected **entity_id**.
- 1380
- 1381 d. Click **Add Column**.
- 1382 e. Select the **Table Columns** to be returned as Splunk fields. We selected **email**.
- 1383 f. Enter an **Alias** for the field. We chose to leave the name of the field as **email**.
- 1384 g. Click **Next**.
- 1385 6. Navigate to **Set Properties**.



- 1386
- 1387 a. Enter a unique **Name** for the lookup. We named our instance **Magento_Customer_Mapping**.
- 1388
- 1389 b. Enter a **Description** for the type of new lookup being created.
- 1390 c. Select the **Application** context. We selected **Splunk DB Connect**.
- 1391 d. The **Summary** contains the command to be appended to your SPL searches to leverage
- 1392 the lookup:
- 1393 | dbxlookup lookup="Magento_Customer_Mapping"
- 1394 e. Click **Finish**.
- 1395 7. The following screen will appear upon completion. Click **Back to List**.



Done!

Activate Windows
Go to System in Control Panel to activate Windows.

1396

1397 8. You will now see the new lookup that you created, listed in the table of lookups.

The screenshot shows the 'Lookups' table in the Splunk interface. The table has columns for Name, Connection, App, Status, and Actions. There is one entry in the table:

Name	Connection	App	Status	Actions
Magento_Customer_Mapping	Magento_DB	Splunk DB Connect	Enabled	Edit Clone Delete

Below the table, it says '1 lookup in total.'

1398

1399 2.6.5 Splunk Enterprise Queries and Dashboards

1400 Splunk Enterprise reports, alerts, and dashboards are powered by queries written in the Splunk SPL.
 1401 These queries are used to perform the analytics responsible for capturing events, identifying trends, and
 1402 detecting anomalies. Once a query is written, it can be saved as a report, an alert, or a dashboard panel.
 1403 The following queries were developed for both example implementations and were also saved as Splunk
 1404 Enterprise dashboards to provide a central viewing location.

1405 2.6.5.1 *Query: Total Attempted Single-Factor Authentications*

1406 The following search query traverses the logs aggregated from the Magento server. The query uses
 1407 multiple data sources relating to the same access log to detect when access to a customer account is
 1408 attempted via single-factor credentials. The output of the query shows the total events per hour.

```
1409 host="magento.mfa.local" source ="/var/log/httpd/*" sourcetype=access_common 302
1410 "/fidodemo/customer/account/loginPost" earliest=1 latest=now | stats count by
1411 date_hour
```

1412 [2.6.5.2 Query: Failed Single-Factor Authentications Within Past Five Minutes](#)

1413 The following search query traverses the logs aggregated from the Magento server, specifically the
1414 database logs. This log returns information, including failed login attempts per entity ID. With the
1415 database lookup created in [Section 2.6.4.8](#), the query below maps the entity ID to the respective email
1416 address reporting when a customer account has failed to be logged in via single-factor credentials. The
1417 output of the query shows failed logins, per email address, within a five-minute interval.

```
1418 source="/usr/local/strongauth/mariadb-10.1.22/log/mysqld.log" failures_num!="0" |  
1419 rex field=entity_id "'?(?<entity_id>[\d\.]+)'?" | dbxlookup  
1420 lookup="Magento_Customer_Mapping" earliest=-5m latest=now | eventstats | stats count  
1421 by email
```

1422 [2.6.5.3 Query: Attempted Single-Factor Authentications in Past Five Minutes](#)

1423 The following search query traverses the logs aggregated from the Magento server. The query uses
1424 multiple data sources relating to the same access log to detect when access to a customer account is
1425 attempted via single-factor credentials. The output of the query shows the failed login, per IP address,
1426 within a five-minute interval.

```
1427 host="magento.mfa.local" source ="/var/log/httpd/*" sourcetype=access_common 302  
1428 "/fidodemo/customer/account/loginPost" earliest=-5m latest=now | stats count by IP
```

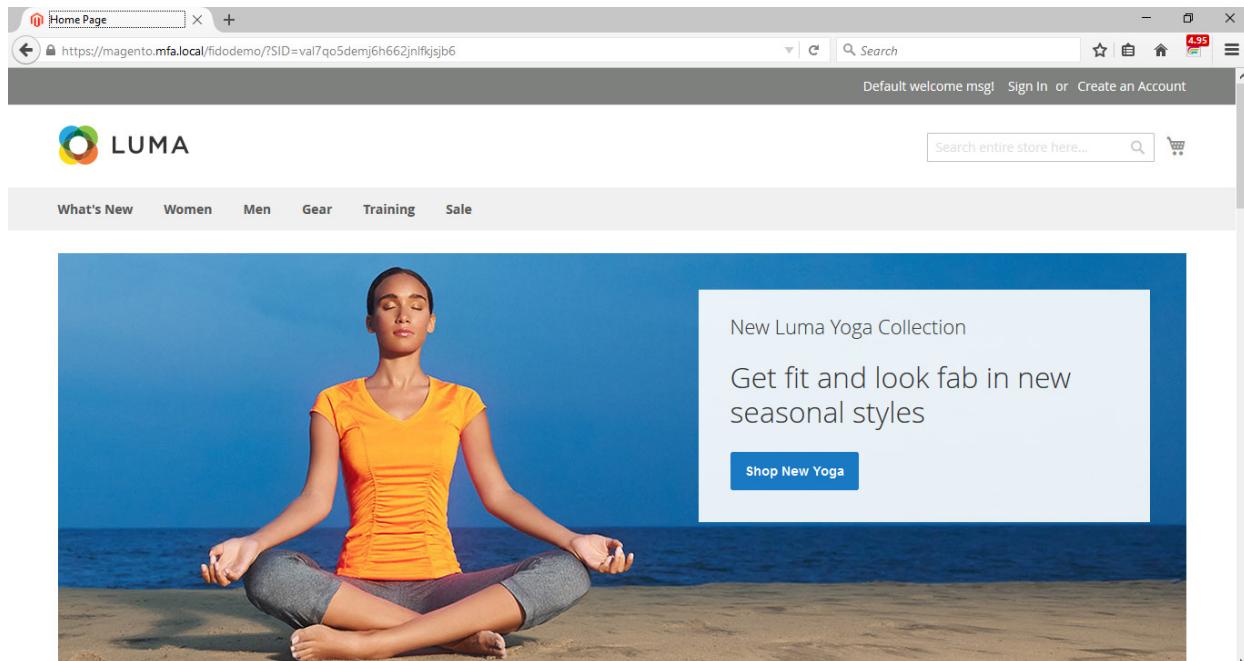
1429 [2.7 Testing FIDO Key Registration and Checkout](#)

1430 Once installed and configured, the example implementation can configure accounts, and the build can
1431 be tested. To test the implementation, an example customer account was created. Example processes
1432 for customer account creation, FIDO key registration, and FIDO checkout are detailed in the following
1433 subsections.

1434 [2.7.1 Creating an Example Magento Customer Account](#)

1435 This section outlines how to create example customer accounts. The accounts are created using a web
1436 browser interface.

- 1437 1. To begin, **open a web browser** and navigate to <https://magento.mfa.local/fidodemo>.



1438

- 1439 2. Click **Create an Account**.
- 1440 3. Fill out the form as shown in the example below.
 - 1441 a. **First Name:** John
 - 1442 b. **Last Name:** Doe
 - 1443 c. **Email:** jdoe@mfa.test.com
 - 1444 d. **Password:** Password!

Create New Customer Account

Personal Information

First Name *
John

Last Name *
Doe

Sign Up for Newsletter

Sign-in Information

Email *
jdoe@mfa.test.com

Password *

Password Strength: Weak

Confirm Password *

Create an Account

1445

- 1446 4. After entering the required information, click **Create an Account**.
- 1447 5. Upon successful account creation, you will be taken to the **Account Dashboard** page, where details of the account that was created are visible.

My Account

Welcome, John! John Doe

LUMA

Search entire store here...

What's New Women Men Gear Training Sale

✓ Thank you for registering with Main Website Store.

Account Dashboard

- Account Information
- Address Book
- My Downloadable Products
- My Orders
- Stored Payment Methods
- Newsletter Subscriptions
- Billing Agreements
- My Product Reviews
- My Wish List

Compare Products

You have no items to compare.

My Dashboard

Account Information

Contact Information
John Doe
jdoe@mfa.test.com
[Edit](#) | [Change Password](#)

Newsletters
You don't subscribe to our newsletter.
[Edit](#)

Address Book [Manage Addresses](#)

Default Billing Address
You have not set a default billing address.
[Edit Address](#)

Default Shipping Address
You have not set a default shipping address.
[Edit Address](#)

1449

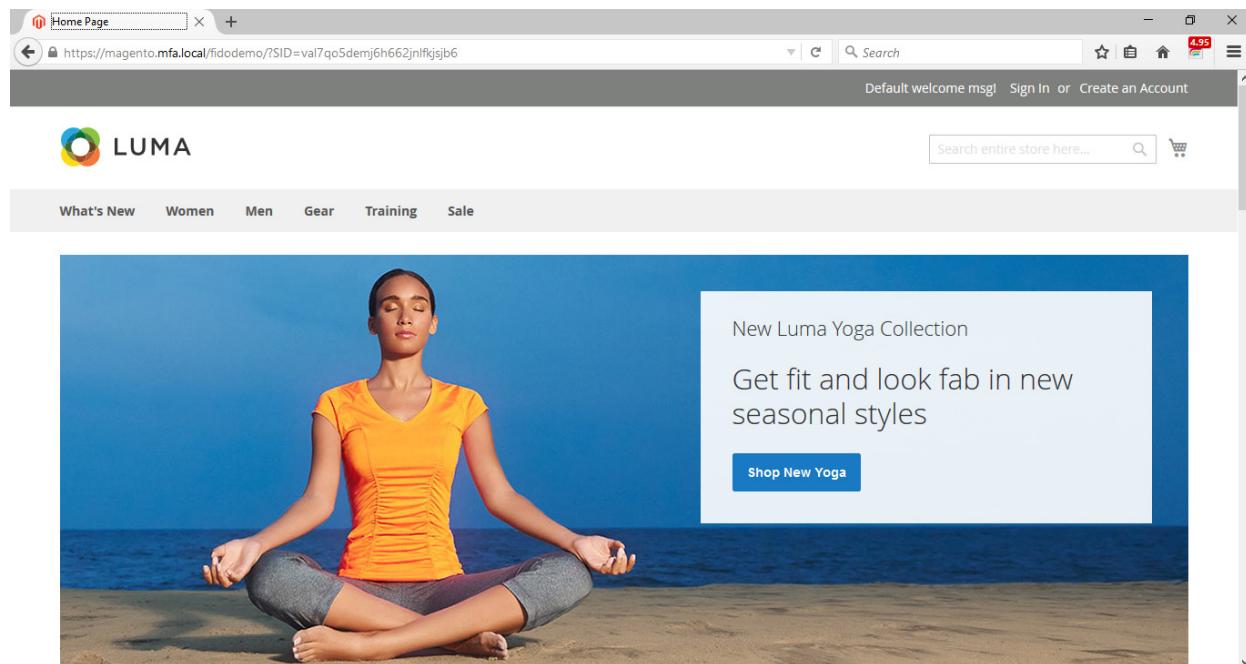
1450 **2.7.2 FIDO Key Registration**

1451 This section provides information for associating the FIDO key with the purchaser's account that was
1452 created in [Section 2.7.1](#). The account holder will need their FIDO key to complete the registration
1453 process.

1454 1. To begin, open a web browser and navigate to <https://magento.mfa.local/fidodemo>.

1455 Note: You need to have already created a Magento Example Customer Account. If you have not
1456 done so, please refer to [Section 2.7.1](#).

1457 2. Click **Sign In**.



1458

1459 3. Fill out the **Email** and **Password** for the example customer account that was created in
1460 [Section 2.7.1](#).



LUMA

What's New Women Men Gear Training Sale

Search entire store here...

Customer Login

Registered Customers

If you have an account, sign in with your email address.

Email *

Password *

Sign In

[Forgot Your Password?](#)

* Required Fields

New Customers

Creating an account has many benefits: check out faster, keep more than one address, track orders and more.

Create an Account

1461

1462 a. Email: jdoe@mfa.test.com

1463 b. Password: Password!

1464 4. Click **Sign In**.

1465 5. On the **Account Dashboard** page, click **Register FIDO Security Key**.

Welcome, John! John Doe USD - US Dollar

Search entire store here...

LUMA

What's New Women Men Gear Training Sale

Thank you for registering with Main Website Store.

My Dashboard

Fido2 Certified

Fido2 FIDO Security Key Registration

Register a FIDO Security Key to protect your purchases with FIDO strong-authentication.

Register FIDO Security Key

Number of registered Security Keys: 1

Account Information

Contact Information

John Doe
jdoe@mfa.test.com
[Edit](#) | [Change Password](#)

Newsletters

You don't subscribe to our newsletter.
[Edit](#)

Compare Products

You have no items to compare.

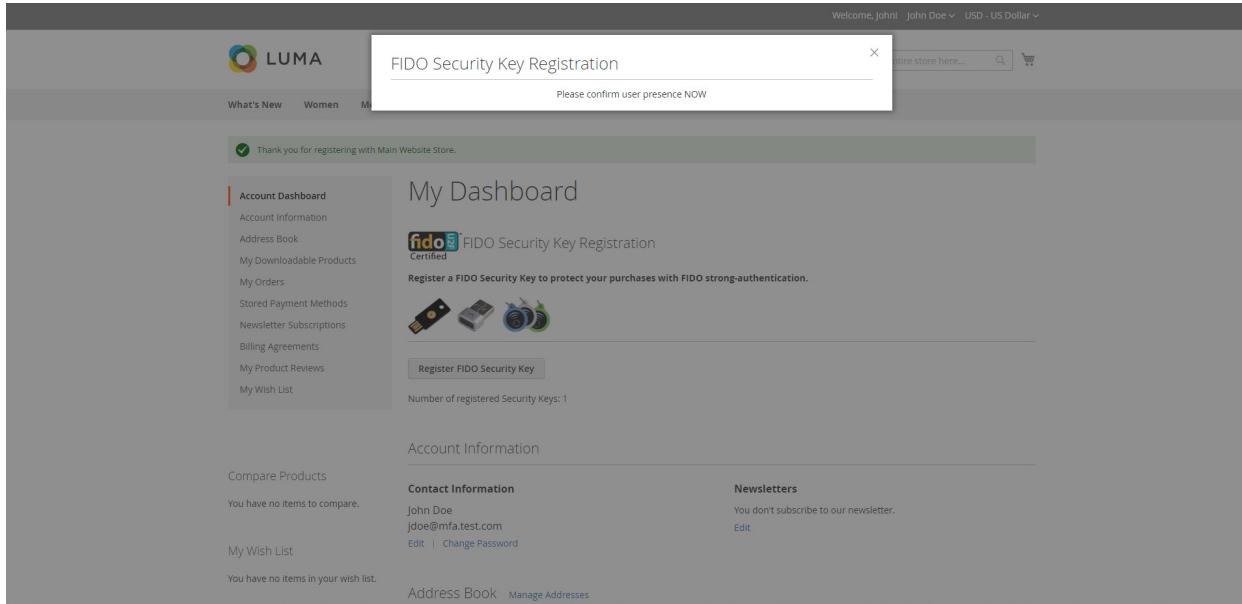
My Wish List

You have no items in your wish list.

[Address Book](#) [Manage Addresses](#)

1466

- 1467 6. The FIDO Authentication Engine will prompt “Please confirm user presence NOW.”



1468

- 1469 Insert the Yubico YubiKey NEO Security Key [15], [16] into an available Universal Serial Bus (USB)
 1470 slot on the computer, and then place a finger on the gold contact pad.
- 1471 7. Successful key registration will result in returning to the **Account Dashboard** page.

 LUMA

What's New Women Men Gear Training Sale

Search entire store here...  

Account Dashboard

- Account Information
- Address Book
- My Downloadable Products
- My Orders
- Stored Payment Methods
- Newsletter Subscriptions
- Billing Agreements
- My Product Reviews
- My Wish List

My Dashboard

 FIDO Security Key Registration

Register a FIDO Security Key to protect your purchases with FIDO strong-authentication.



[Register FIDO Security Key](#)

Number of registered Security Keys: 2

Account Information

Compare Products
You have no items to compare.

My Wish List
You have no items in your wish list.

Contact Information
John Doe
jdoe@mfa.test.com
[Edit](#) | [Change Password](#)

Newsletters
You don't subscribe to our newsletter.
[Edit](#)

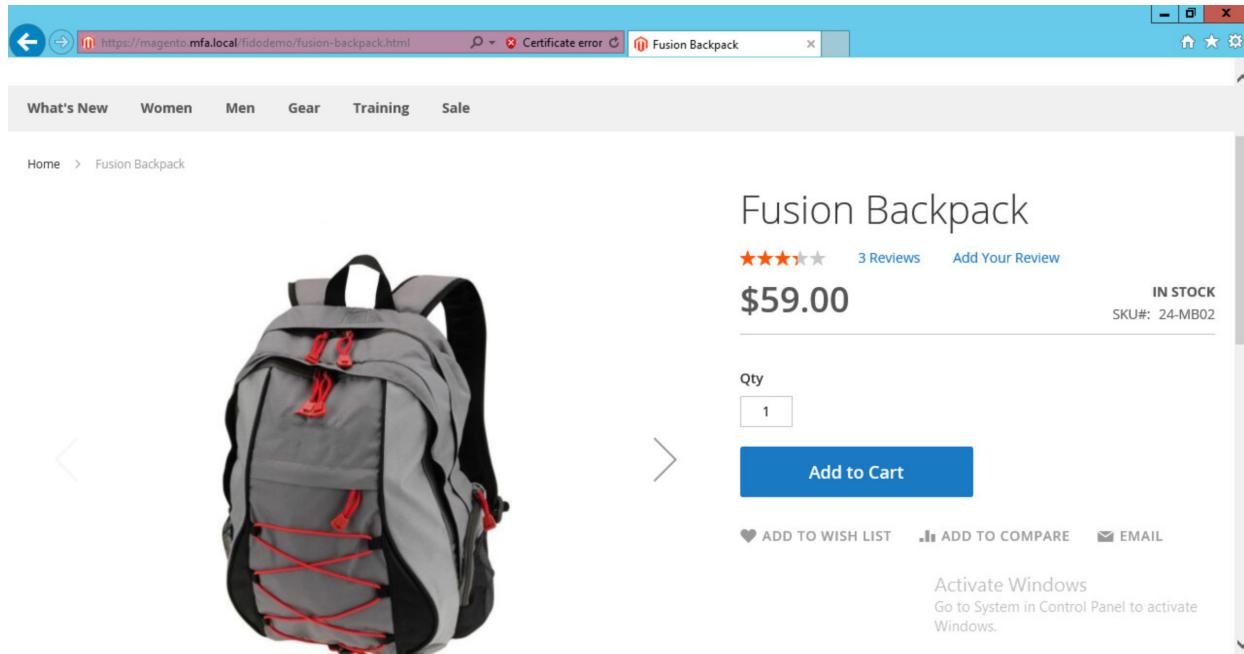
Address Book [Manage Addresses](#)

1472

2.7.3 Testing Customer Checkout

1474 This section provides information for testing that the FIDO server is prompting for a second form of
 1475 authentication for purchases above \$25. This section assumes that an example customer account has
 1476 been created with a registered FIDO Security Key ([Section 2.7.1](#) and [Section 2.7.2](#)).

- 1477 1. Open a web browser and navigate to <https://magento.mfa.local/fidodemo>.
- 1478 2. If not already logged into an example customer account, select **Sign In** from the Magento home
1479 page and log in with the following credentials:
 - 1480 a. **Email:** jdoe@mfa.test.com
 - 1481 b. **Password:** Password!
- 1482 3. You will be taken to the **Account Dashboard** page.
- 1483 4. From there, navigate back to <https://magento.mfa.local/fidodemo>.
- 1484 5. Scroll down the page and select any item over \$25. For our demonstration, we have selected the
1485 Fusion Backpack.



1486

- 1487 6. Click **Add to Cart**.
- 1488 7. Click the shopping-basket icon, and then click **Go to Checkout**.

1489

- 1490 8. Under **Shipping Methods**, select the **Fixed – Flat Rate** radio bubble.

The screenshot shows the Magento checkout process at the 'Shipping' step. The progress bar has two steps: 'Shipping' (marked with a checkmark) and 'Review & Payments' (step 2). The 'Shipping Address' section displays a populated address: John Doe, 123 Freedom Way, Rockville, Maryland 20850, United States, 4105551234. A red box highlights this address input field. To the right is the 'Order Summary' which shows 1 item in cart: a Fusion Backpack at \$59.00, Qty: 1. Below the summary is a note: 'Activate Windows Go to System in Control Panel to activate Windows.' At the bottom right is a blue 'Next' button.

1491

1492 9. Click **Next**.1493 10. On the following page, select **Place Order**.

The screenshot shows the Magento payment method selection screen. The progress bar now has both 'Shipping' and 'Review & Payments' steps marked with checkmarks. The 'Payment Method:' section shows 'Check / Money order' selected. Below it, there's a checkbox for 'My billing and shipping address are the same' which is checked. The address is identical to the one in the previous screenshot. To the right is the 'Order Summary' which shows the cart subtotal (\$59.00), shipping (\$5.00), and order total (\$64.00). It also lists 1 item in cart: a Fusion Backpack at \$59.00, Qty: 1. Below the summary is the same 'Activate Windows' note. At the bottom right is a blue 'Place Order' button.

1494

Apply Discount Code ▾

1495 11. The FIDO Authentication Engine will prompt "Please confirm user presence NOW."

The screenshot shows a Magento checkout process. A modal window titled "FIDO Authentication" is displayed, containing the message "Please confirm user presence NOW" and a small circular progress bar. The background shows the "Payment Method:" section with "Check / Money order" selected, and the "Order Summary" table which includes items like Cart Subtotal (\$52.00), Shipping (\$5.00), and Order Total (\$57.00). A "Place Order" button is visible at the bottom.

1496

- 1497 12. Insert the Yubico YubiKey NEO Security Key into an available USB slot on the computer, and then
 1498 place a finger on the gold contact pad.
- 1499 13. Successfully activating the FIDO token will result in the order confirmation page.

The screenshot shows an order confirmation page for the LUMA theme. It displays the order number (00000006) and a message stating that an order confirmation with tracking info will be emailed. A "Continue Shopping" button is present. At the bottom, there are links for "About us", "Customer Service", "Privacy and Cookie Policy", "Search Terms", and "Advanced Search". A newsletter sign-up form with fields for email and a "Subscribe" button is also shown.

1500

1501

1502 **Appendix A FIDO U2F Security Key Registration**

1503 Fast IDentity Online (FIDO) authentication requires registering one or more *FIDOU2FAuthenticators*, also
1504 known as FIDO Universal Second Factor (U2F) Security Keys, or security keys. Security keys can be used
1505 for authentication on multiple information systems or websites. If the purchaser already has a U2F, then
1506 they can use that same U2F as their multifactor authenticator for the electronic commerce
1507 (e-commerce) example implementations depicted in this guide.

1508 FIDO authentication in these example implementations is accomplished by using the magfido
1509 *FIDOU2FAuthenticator* module created by StrongKey for the Magento Open Source platform. When
1510 deploying the example implementations, there are three parts to the process. While these three parts
1511 all execute in sequence, without the purchaser being aware of each part, it is helpful to explain each
1512 part so that developers understand the workflow.

1513 **A.1 Display Function**

1514 In this part of the process, the Magento layout file *customer_account_index.xml* loads code from the
1515 *fido_register.phtml* file on the server side to perform these two functions:

- 1516 1. Generate HyperText Markup Language (HTML) that displays FIDO registration purchaser-
1517 interface components in the browser, along with summary information of the number of
1518 security keys that a purchaser may have registered. The summary information on registered keys
1519 is shown above the Recent (Magento) Orders section, which normally appears at the top of the
1520 dashboard.
- 1521 2. Execute the FIDO registration process to register a new FIDO Security Key, using JavaScript
1522 embedded in the *fido_register.phtml* file.

1523 If a purchaser has not yet registered a FIDO Security Key within Magento, then the HTML displays a zero
1524 (0) value for the number of registered keys, and a button to register a new security key ([Figure A-1](#)).

1525 **Figure A-1 Browser Display Without Any Security Keys Registered**

Welcome, John! John Doe ▾

LUMA

What's New Women Men Gear Training Sale

Search entire store here...

Account Dashboard

- Account Information
- Address Book
- My Downloadable Products
- My Orders
- Stored Payment Methods
- Newsletter Subscriptions
- Billing Agreements
- My Product Reviews
- My Wish List

My Dashboard

fido2 Certified FIDO Security Key Registration

Register a FIDO Security Key to protect your purchases with FIDO strong-authentication.

Number of registered Security Keys: 0

Compare Products

You have no items to compare.

My Wish List

You have no items in your wish list.

Account Information

Contact Information

John Doe
johndoe@testing.com
[Edit](#) | [Change Password](#)

Newsletters

You don't subscribe to our newsletter.
[Edit](#)

Address Book [Manage Addresses](#)

Default Billing Address

Default Shipping Address

1526

1527 If a purchaser has registered one or more security keys to their account—which the FIDO U2F protocol
 1528 allows—then the *FIDOU2FAuthenticator* module displays the number of security keys registered by the
 1529 purchaser. Otherwise, it displays 0. The HTML display for such a purchaser’s registered keys resembles
 1530 the depiction shown in [Figure A-2](#).

1531 **Figure A-2 Browser Display with Two Security Keys Registered**

The screenshot shows a LUMA-themed Magento storefront. At the top, there's a navigation bar with 'Welcome, John! John Doe ▾', a search bar ('Search entire store here...'), and a shopping cart icon. Below the header is a main menu with links to 'What's New', 'Women', 'Men', 'Gear', 'Training', and 'Sale'. The main content area is titled 'My Dashboard'. On the left, a sidebar lists account-related links: 'Account Dashboard', 'Account Information', 'Address Book', 'My Downloadable Products', 'My Orders', 'Stored Payment Methods', 'Newsletter Subscriptions', 'Billing Agreements', 'My Product Reviews', and 'My Wish List'. The main content area features a 'FIDO Security Key Registration' section with the text 'Register a FIDO Security Key to protect your purchases with FIDO strong-authentication.' and three icons representing different types of FIDO keys. Below this is a button labeled 'Register FIDO Security Key'. Further down, it says 'Number of registered Security Keys: 2'. The page also includes sections for 'Compare Products' (with a message about no items to compare) and 'Account Information' (with contact details for 'John Doe' and an option to 'Edit').

1532

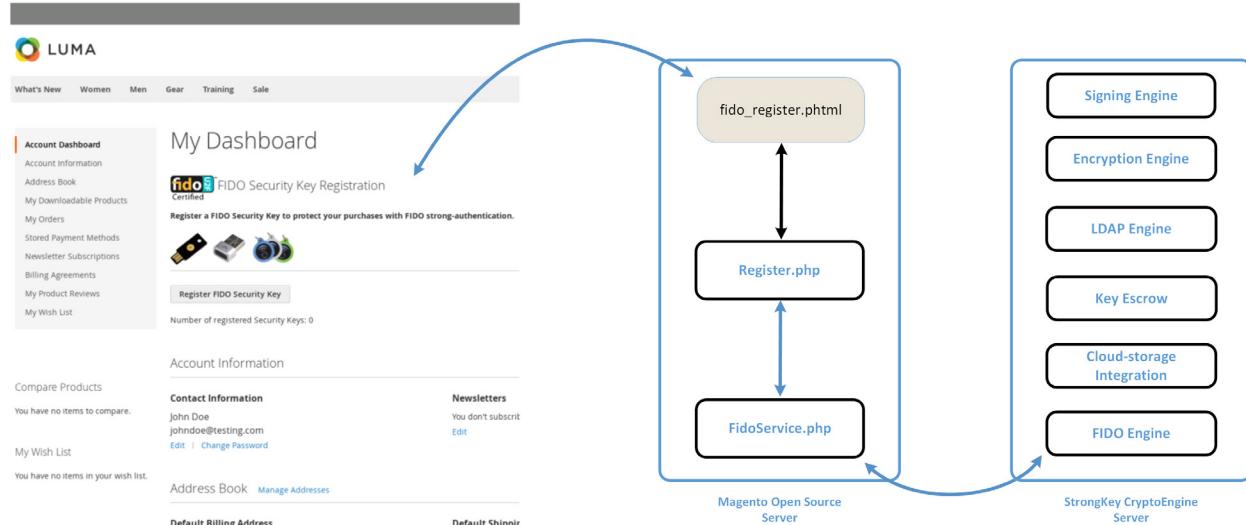
- 1533 To determine the number of FIDO Security Keys registered by a purchaser within their account, the
 1534 server code in *fido_register.phtml* calls the “block” file, *Register.php*. This Hypertext Preprocessor (PHP)
 1535 file, in turn, invokes *FidoService.php* to call a web service (also sometimes known as “consume a web
 1536 service”) on a previously configured FIDO U2F server (implemented in StrongKey CryptoEngine [SKCE])
 1537 known to the Magento instance. The web-service request retrieves security-key-related information for
 1538 the specific purchaser, from the FIDO server.
- 1539 *FidoService.php* parses the retrieved number of registered keys and returns the value to *Register.php*,
 1540 which, in turn, returns the number to *fido_register.phtml* that generates HTML for the browser to
 1541 display.

Note: In this example implementation, *Register.php* is executed only when the purchaser navigates to their purchaser-dashboard page. If a new security key is registered while on that page, then the page is automatically refreshed upon completion of the transaction to display the correct number of registered security keys.

1542

1543 An overview diagram of the first part of the registration process—that displays the current number of
 1544 registered security keys, if any—is shown in [Figure A-3](#).

1545 **Figure A-3 Display Function Part of the FIDO Registration Process**



1546

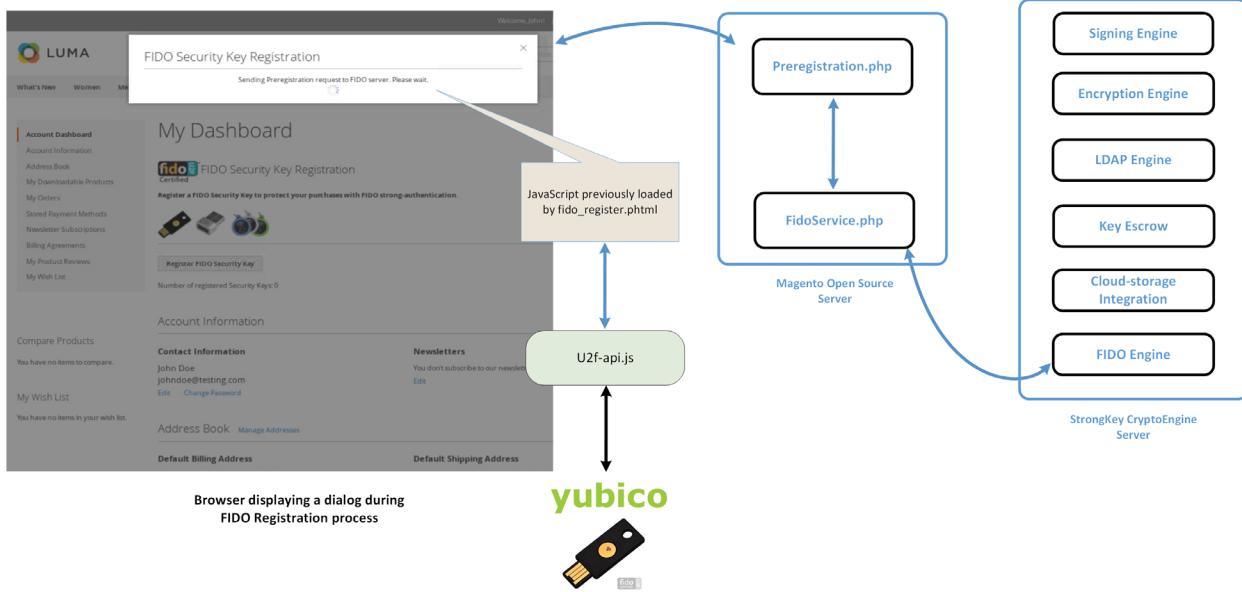
1547 A.2 Preregister Function

1548 The second part of the FIDO registration process acquires a challenge from the FIDO U2F server (SKCE)
 1549 for processing within the purchaser's FIDO Security Key ([Figure A-4](#)).

1550 When the **Register FIDO Security Key** button on the browser is clicked by the purchaser, JavaScript that
 1551 was loaded earlier in the web page (by *fido_register.phtml*) makes an Asynchronous JavaScript and XML
 1552 [Extensible Markup Language] (AJAX) call to *Preregistration.php* on the Magento server, which, in turn,
 1553 invokes *FidoService.php* to call the ***preregister*** web-service operation on the SKCE. SKCE returns a nonce,
 1554 along with a list of previously registered FIDO Security Keys, if any. If this is the first security key being
 1555 registered, then this list is empty.

Note: In the FIDO U2F protocol, currently registered security keys, if any, are returned by the FIDO server to safeguard that security keys do not attempt to generate a duplicate key for purchasers on the same device. This implies that manufacturers of FIDO Security Keys must implement logic to ensure that they check for an existing key pair for a purchaser for the specific website. A FIDO Certified Authenticator will always have this logic implemented because it is part of the protocol-conformance testing to achieve the FIDO Certified label.

1556

1557 **Figure A-4 Preregistration Part of the FIDO Registration Process**

1558

1559 Upon receiving the challenge, the browser and the security key interact with each other by using the
 1560 *u2f-api.js* library to perform FIDO U2F-specified protocol functions. If the security key does not already
 1561 have a cryptographic key pair for this specific website domain, then it requires the purchaser to perform
 1562 an action to prove their presence in front of the computer. Upon the purchaser doing so, it generates a
 1563 new Elliptic Curve Digital Signature Algorithm (ECDSA) key pair.

1564 The “purchaser action” may be something chosen by the manufacturer of the security key, such as these
 1565 actions:

- 1566 ■ touching a metallic component or pressing a button that has a blinking light-emitting diode
- 1567 ■ removing and reinserting a Universal Serial Bus (USB)-based security key
- 1568 ■ bringing a Near Field Communication (NFC)-based security key near the NFC-enabled
 1569 computer/mobile device
- 1570 ■ scanning their finger or iris on a mobile device enabled with biometric capabilities
- 1571 ■ additional manufacturer choices

1572 FIDO protocols do not mandate any specific user/purchaser action for the test of human presence.
 1573 Manufacturers are at liberty to choose whatever complies with the protocol.

1574 A.3 Register Function

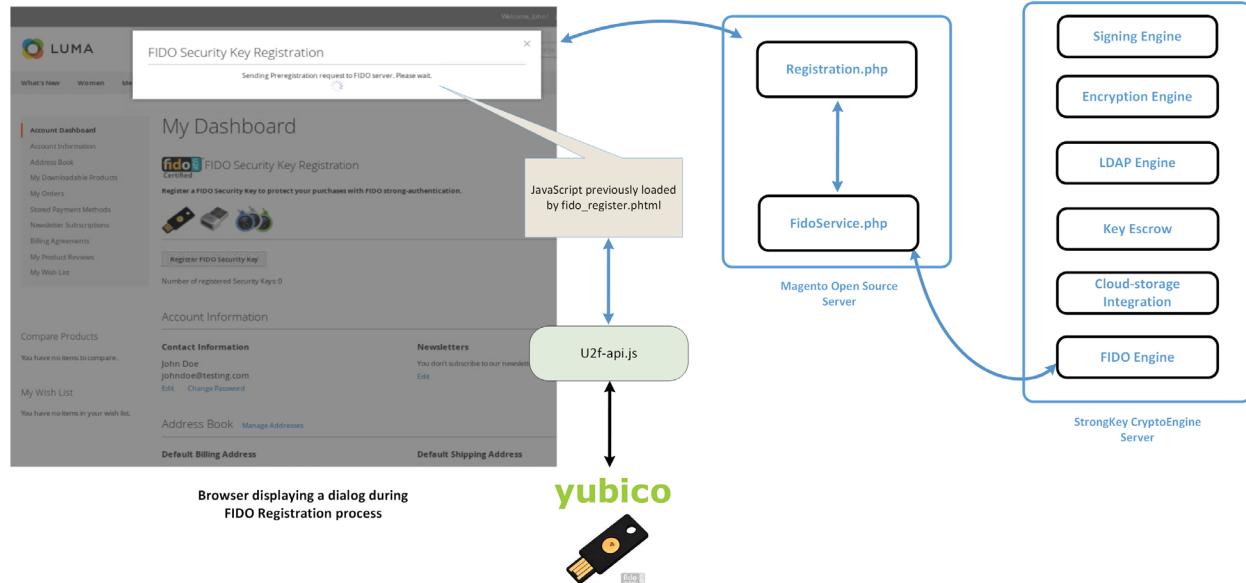
1575 The third, and last, part of the FIDO registration process generates a new key pair for the purchaser for
 1576 the specific website domain on the purchaser's FIDO Security Key, digitally signs the challenge from the
 1577 FIDO U2F server (SKCE), and then submits a package of the response to SKCE for processing.

1578 When the purchaser has "activated" their FIDO Security Key by using the mechanism that the
 1579 manufacturer designed into the process, the security key generates a new ECDSA key pair, uses the
 1580 newly generated private key from the key pair to digitally sign the nonce, and assembles a package of
 1581 information to return to the browser. The browser sends the package to *Registration.php*, which, in
 1582 turn, sends the package to *FidoService.php*, which finally calls the *register* web-service operation on
 1583 SKCE to register the newly generated public key with the FIDO server.

1584 During this process, *fido_register.phtml* displays a modal dialogue to notify purchasers of progress
 1585 and/or error messages, should something go wrong. Any interaction with the modal dialogue, such as
 1586 closing it, does not affect the operation. The operation continues until it succeeds or fails.

1587 This last step of the registration process is shown in [Figure A-5](#).

1588 **Figure A-5 Third and Final Step of the FIDO Registration Process**



1589

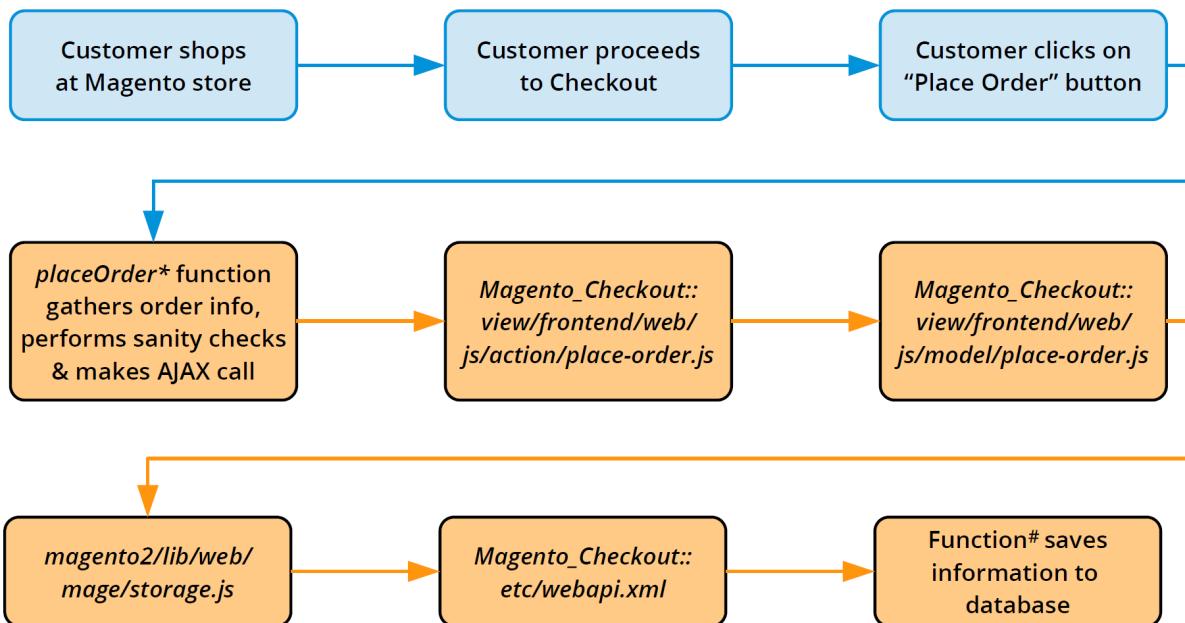
1590 **A.3.1 The Checkout Process**

1591 The *FIDO2FAuthenticator* module must integrate with Magento's default checkout workflow.
 1592 Before describing the FIDO authentication process, a brief background of the default checkout workflow
 1593 is presented below.

- 1594 1. Purchasers browse the e-commerce website to purchase one or more items.
- 1595 2. Purchasers place and remove items in and out of their shopping cart, until they decide to pur-
 1596 chase the items in their shopping cart.
- 1597 3. Purchasers click **Proceed to Checkout**.
- 1598 4. At this point, the checkout process requires the purchaser to fill out billing and shipping infor-
 1599 mation, and then to click **Place Order**.
- 1600 5. This causes the browser to run JavaScript code, which makes an AJAX call to submit the shop-
 1601 ping cart, billing address, and payment information to the Magento server.
- 1602 6. The Magento server processes the information and saves it to its database—or returns an error
 1603 if there is an exception—confirming the conclusion of the transaction.

1604 The checkout workflow is displayed in [Figure A-6](#).

1605 **Figure A-6 Magento Checkout Workflow**



1606

Note: In [Figure A-6](#),

* `placeOrder` is in `Magento_Checkout::view/frontend/web/js/view/payment/default.js`

`savePaymentInformationAndPlaceOrder` is in
`Magento_Checkout::PaymentInformationManagement`

1607

1608 By understanding the above Magento default checkout workflow, you can better understand how the
1609 example implementations' FIDO authentication flow is implemented.

1610 A.3.2 The FIDO Authentication Flow for the Example Implementations

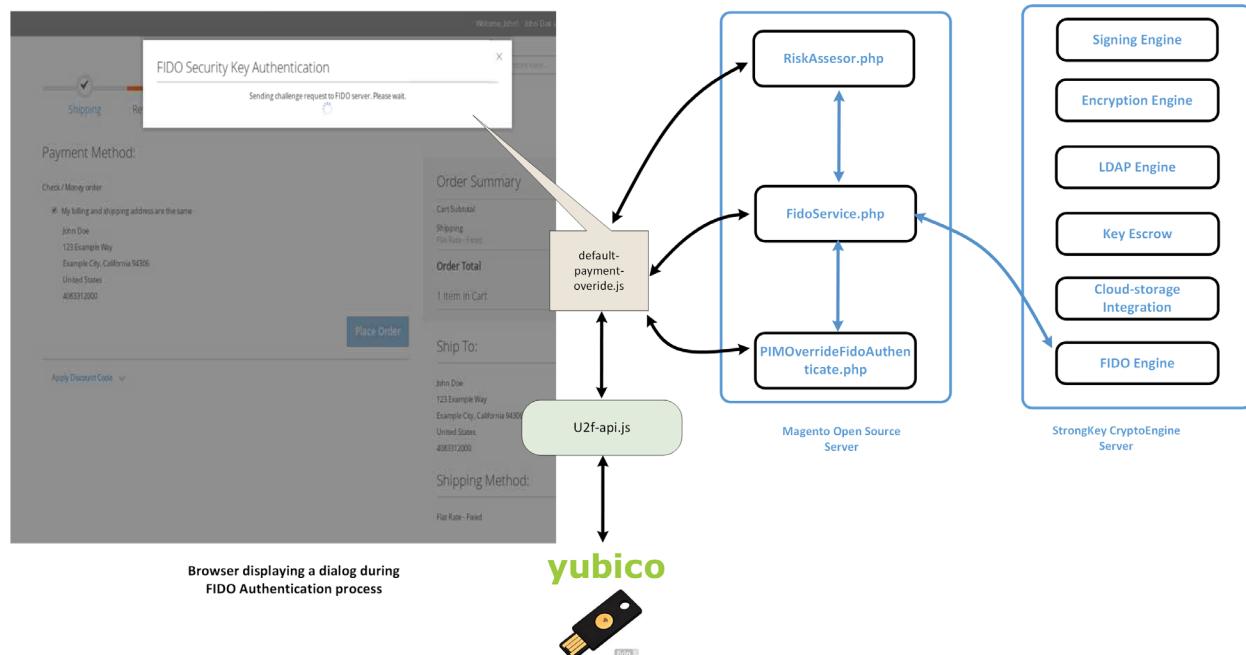
1611 The *FIDOU2FAuthenticator* module, when installed, will inject itself into the workflow described above.
1612 The primary modification that FIDO authentication makes to the checkout process is to override
1613 *Magento_Checkout/view/payment/default.js*'s *placeOrder* function.

- 1614 1. The new *placeOrder* function makes an AJAX call to the *RiskAssessor.php* on the Magento server
1615 to determine whether FIDO authentication is required (based on this example implementation's
1616 rule to check whether the total order is greater than \$25).
- 1617 2. If the total is \$25 or less, then the checkout data is sent to the Magento server to be persisted
1618 directly without any FIDO actions. However, if the order total exceeds \$25, then another AJAX
1619 call is made to *FidoService.php* to request a FIDO challenge from SKCE. This is accomplished by
1620 *FidoService.php* making a *preauthenticate* web-service request to SKCE, the FIDO U2F server.
1621 *FidoService.php* returns the challenge nonce to the calling JavaScript in the customer's browser.
- 1622 3. Upon receiving the challenge, the browser interacts with *u2f-api.js* to prompt the customer to
1623 digitally sign the challenge by using their FIDO Security Key.
- 1624 4. Once the challenge nonce has been signed by using the FIDO Security Key, the digital signature
1625 is appended to checkout data that is normally sent to the Magento server.
- 1626 5. On the server, where the *Magento_Checkout/Model/PaymentInformationManagement save-*
1627 *PaymentInformationAndPlaceOrder* function has been overridden, Magento receives the check-
1628 out data and checks again if FIDO authentication is required. This is to ensure that web-service
1629 requests to the back-end services are not manipulated to bypass FIDO strong authentication.
- 1630 6. If FIDO strong authentication is not required, then Magento goes through the standard checkout
1631 flow and persists the transaction. If FIDO strong authentication is required, then the overridden
1632 code in *PIMOverrideFidoAuthenticate.php* checks for the digital signature bytes appended to the
1633 checkout data.

- 1634 7. If the signature bytes are present, then *PIMOverrideFidoAuthenticate.php* calls the *authenticate*
 1635 web-service operation (by using *FidoService.php*) on SKCE with the signature bytes.
- 1636 8. If the *authenticate* web service returns successfully, then *PIMOverrideFidoAuthenticate.php* con-
 1637 tinues with the checkout process, persists transaction data to the database, and confirms the
 1638 transaction to the customer. A failed response to the *authenticate* web service returns an error
 1639 to the customer, and the checkout fails.
- 1640 In the browser, a modal dialogue provides status messages on the FIDO strong-authentication process
 1641 executing in the background (if FIDO strong authentication is determined to be necessary); otherwise,
 1642 the FIDO dialogue does not display itself. As in the FIDO registration workflow, closing the modal
 1643 dialogue does not stop the FIDO authentication process, and interacting with the browser window in
 1644 any way does not change the behavior.

1645 [Figure A-7](#) provides an overview of the FIDO authentication process at a high level.

1646 **Figure A-7 Overview of the FIDO Authentication Process**



1647

1648 **A.3.3 Information About the magfido Files and Directories**

1649 This section provides additional information regarding files referenced and/or modified by StrongKey to
 1650 implement FIDO U2F MFA for these example implementations. If you are familiar with Magento, then
 1651 you may skip this section; others may find this section to be helpful in understanding what must be done
 1652 to integrate FIDO U2F into their Magento instance in a production environment.

1653 Magento includes several boilerplate/configuration files: *composer.json* and *registration.php* are those
1654 that must be included in every Magento module — because they identify the module to the Magento
1655 system.

1656 The *etc* folder contains configuration files:

- 1657 ■ *module.xml* is a boilerplate file.
- 1658 ■ *di.xml* tells Magento to override the default *PaymentInformationManagement.php* file with
1659 StrongKey's custom version (named *PIMOverrideFidoAuthenticate.php*).
- 1660 ■ *extension_attributes.xml* tells Magento that purchase-transaction data sent to the server may
1661 have signature data appended to it, which can be identified by the attribute name *signature*.
- 1662 ■ *etc/frontend/di.xml* adds an *AdditionalConfigProvider* that supplies the MFA modal dialogue
1663 with the file name *loading.gif*.
- 1664 ■ *routes.xml* tells Magento that this module defines controllers that will handle Uniform Resource
1665 Locator (URL) requests to fidou2fauthenticator.

1666 The *api* folder contains interface files describing valid functions of the models *FidoService* and
1667 *RiskAssessor*. The interface files are named *FidoServiceInterface.php* and *RiskAssessorInterface.php*.

1668 The *block* folder contains server-side logic to generate views displayed by the browser. Specifically, it
1669 contains the file *Register.php* that provides the base URL for AJAX calls in the registration workflow and
1670 returns the number of security keys registered to the online customer.

1671 The *controller* folder contains controllers to handle AJAX calls from the browser. The controllers map to
1672 SKCE web services, such as *preregistration*, *registration*, and *preauthentication*. Because FIDO
1673 authentication is part of the checkout process and is performed in conjunction with payment data, an
1674 explicit controller for FIDO authentication is not defined here, but is included as part of
1675 *PIMOverrideFidoAuthentication*. It also contains the *RiskAssessor.php* controller to call the
1676 *RiskAssessor.php* code in the *model* folder (see below), which performs the actual risk assessment.

1677 The *model* folder contains the following server-side logic files:

- 1678 ■ *AdditionalConfigProvider.php* retrieves the static URL of the *loading.gif* image and adds it to
1679 variables for the browser client to deliver a better user experience.
- 1680 ■ *FidoService.php* makes the actual web-service calls to the FIDO U2F server, SKCE.
- 1681 ■ *RiskAssessor.php* makes the risk decision in this example implementation—to check if the
1682 order's total value is greater than \$25—and returns a *Boolean* value indicating if FIDO
1683 multifactor authentication (MFA) is necessary or not.
- 1684 ■ *PIMOverrideFidoAuthentication.php* implements the server-side logic to check, once again, if
1685 FIDO MFA is necessary, checking if signature bytes are appended to payment data, verifying if

1686 the supplied digital signature is valid (through *FidoService.php*), and persisting the order
1687 transaction.

1688 The *view* folder contains the client-side logic. Because all FIDO-related workflows in this example
1689 implementation are intended for customer interaction only, there is a *frontend* folder inside the *view*
1690 folder (as opposed to an *adminhtml* folder, which would normally define views for administrators).
1691 Within the *frontend* folder, there are four groups of files:

- 1692 ▪ The first group contains files related to the registration workflow:
1693 *layout/customer_account_index.xml* directs Magento to load *templates/fido_register.phtml*
1694 above the Recent Orders section of the Customer dashboard in the browser. *fido_register.phtml*
1695 coordinates the entire FIDO registration workflow.
- 1696 ▪ The second group contains files related to the modal dialogue: *layout/checkout_index_index.xml*
1697 appends JavaScript from *web/js/view/checkout-modal.js* to JavaScript normally loaded on
1698 checkout pages. *checkout-modal.js*, in turn, loads *web/template/checkout-modal.html* with
1699 HTML that is rendered on the checkout page.
- 1700 ▪ The third group of files provides client-side logic to perform FIDO authentication. *requirejs-*
1701 *config.js* is a configuration file to load JavaScript libraries found in *web/js/lib*—including *u2f.js*
1702 and *common.js*, which are part of the standard distribution for FIDO U2F from Google for use
1703 with the Chrome browser—and overrides the default JavaScript in
1704 *Magento_Checkout/js/view/payment/default.js* with *web/js/default-override.js*. The latter file—
1705 *default-override.js*—provides client-side logic, including requesting the challenge nonce, getting
1706 the challenge nonce digitally signed by the FIDO Security Key, returning the digital signature,
1707 and updating the modal dialogue with progress information.
- 1708 ▪ The last group of files found in the *view/frontend* folder contains image files found in
1709 *web/images/*.

1710 A.3.4 Solutions to Common Challenges When Configuring Magento and magfido

1711 The following subsections provide solutions to common challenges when the magfido module is
1712 configured with Magento.

1713 A.3.4.1 Code Was Modified but Change Did Not Take Effect

1714 The most common reason for this issue is that Magento’s cache was not cleared. Clear the browser
1715 cache from the browser’s admin console, or open a terminal, change to the Magento directory
1716 (*/var/www/html/fidodemo*), and run this command:

1717

```
php bin/magento cache:flush
```

1718 *A.3.4.2 Magento Is Unable to Read the WSDL of the FIDO Server*

1719 Possible reasons for Magento being unable to read the FIDO server's Web Services Description Language
1720 (WSDL), and thus being unable to complete the action, are explained below.

- 1721 ■ The Fully Qualified Domain Name (FQDN) of the FIDO server was defined incorrectly. This can be
1722 fixed by modifying the WSDL constant in *StrongAuth_FidoValidator/Model/FidoService.php*.
- 1723 ■ The FIDO server has a self-signed certificate that Hypertext Transfer Protocol Daemon (HTTPD)
1724 does not trust. This can be fixed by adding the self-signed certificate to the trusted certificate
1725 store located in */etc/pki/tls/certs/ca-bundle.crt*.
- 1726 ■ The Security-Enhanced Linux (SELinux) security policy is blocking the outbound port used by
1727 HTTPD to connect to the FIDO server. This can be fixed by disabling SELinux for testing purposes.
1728 In production environments, it is recommended that SELinux rules be modified to permit HTTPD
1729 to connect to the FIDO server.

1730 *A.3.4.3 Error 500 When Attempting to Access the Home Page*

1731 This is not a FIDO-related issue, but can manifest itself as a Magento-HTTPD misconfiguration. While
1732 there are many possible ways that this error can occur, the most common reason is incorrect file
1733 permissions. For testing purposes, running the following command should fix the problem to make the
1734 Magento home page accessible:

```
1735 cd /var/www/html/fidodemo && find var vendor pub/static pub/media app/etc -type f -  
1736 exec chmod 777 {} \; && find var vendor pub/static pub/media app/etc -type d -exec  
1737 chmod 777 {} \; && chmod 777 bin/magento
```

1738 In production environments, consider the security ramifications before adjusting permissions to the
1739 directory structure and files, and before making modifications. Please note that the command shown
1740 above is a concatenation of multiple commands executed as a single command, so either execute them
1741 in a single command (as shown above) or execute them as multiple commands in sequence:

```
1742 cd /var/www/html/fidodemo  
1743 find var vendor pub/static pub/media app/etc -type f -exec chmod 777 {} \;  
1744 find var vendor pub/static pub/media app/etc -type d -exec chmod 777 {} \;  
1745 chmod 777 bin/magento
```

1746

1747 Appendix B List of Acronyms

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CentOS	Community Enterprise Operating System
DNS	Domain Name System
ECDSA	Elliptic Curve Digital Signature Algorithm
e-commerce	Electronic Commerce
FIDO	Fast IDentity Online
FQDN	Fully Qualified Domain Name
GB	Gigabyte(s)
HTML	HyperText Markup Language
HTTPD	Hypertext Transfer Protocol Daemon
HTTPS	Hypertext Transfer Protocol Secure
ID	Identifier
IP	Internet Protocol
IT	Information Technology
JDK	Java Development Kit
JRE	Java Runtime Environment
LAMP	Linux, Apache, MySQL, PHP
LDAP	Lightweight Directory Access Protocol
MFA	Multifactor Authentication
NCCoE	National Cybersecurity Center of Excellence
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
PHP	Hypertext Preprocessor
PIN	Personal Identification Number

QR	Quick Response
RAM	Random Access Memory
SELinux	Security-Enhanced Linux
SKCE	StrongKey CryptoEngine
SP	Special Publication
SPL	Splunk Search Processing Language
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
U2F	Universal Second Factor
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
WSDL	Web Services Description Language
XML	Extensible Markup Language

1749 **Appendix C****Glossary**

Authentication	Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to a system's resources [17]
Authenticator	Something the claimant possesses and controls (typically a cryptographic module or password) that is used to authenticate the claimant's identity [17]
Credential	An object or data structure that authoritatively binds an identity — via an identifier or identifiers — and (optionally) additional attributes to at least one authenticator possessed and controlled by a subscriber While common usage often assumes that the subscriber maintains the credential, these guidelines also use the term to refer to electronic records maintained by the Credential Service Providers that establish binding between the subscriber's authenticator(s) and identity. [17]
Credential Service Provider	A trusted entity that issues or registers subscriber authenticators and issues electronic credentials to subscribers. A Credential Service Provider may be an independent third party or issue credentials for its own use. [17]
Identity	An attribute, or set of attributes, that uniquely describes a subject within a given context [17]
Multifactor	A characteristic of an authentication system or an authenticator that requires more than one distinct authentication factor for successful authentication. MFA can be performed by using a single authenticator that provides more than one factor or by using a combination of authenticators that provide different factors. The three authentication factors are something you know, something you have, and something you are. [17]
Multifactor Authentication (MFA)	An authentication system that requires more than one distinct authentication factor for successful authentication. Multifactor authentication can be performed by using a multifactor authenticator or by using a combination of authenticators that provide different factors. The three authentication factors are something you know, something you have, and something you are. [17]
Personal Identification Number (PIN)	A memorized secret typically consisting of only decimal digits [17]

Private Key	The secret part of an asymmetric key pair that is used to digitally sign or decrypt data [17]
Public Key	The public part of an asymmetric key pair that is used to verify signatures or encrypt data [17]
Public Key Certificate	A digital document issued and digitally signed by the private key of a certificate authority that binds an identifier to a subscriber to a public key. The certificate indicates that the subscriber identified in the certificate has sole control and access to the private key. See also RFC 5280 [17]
Relying Party	An entity that relies upon the subscriber's authenticator(s) and credentials or a verifier's assertion of a claimant's identity, typically to process a transaction or grant access to information or a system [17]
Risk	The level of impact on organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals resulting from the operation of an information system, given the potential effect of a threat and the likelihood of that threat occurring [18]
Session	A persistent interaction between a subscriber and an endpoint, either a relying party or a Credential Service Provider. A session begins with an authentication event and ends with a session termination event. A session is bound by use of a session secret that the subscriber's software (a browser, application, or OS) can present to the relying party or the Credential Service Provider, in lieu of the subscriber's authentication credentials. [17]
Single-Factor	A characteristic of an authentication system or an authenticator that requires only one authentication factor (something you know, something you have, or something you are) for successful authentication [17]
Subscriber	A party who has received a credential or authenticator from a Credential Service Provider [17]
Token	See Authenticator [17]
Transaction	A discrete event between a user and a system that supports a business or programmatic purpose. A government digital system may have multiple categories or types of transactions, which may require separate analysis within the overall digital identity risk assessment. [17]

- 1751 **Appendix D References**
- 1752 [1] FIDO Alliance. (n.d.). *Specifications Overview* [Online].
1753 Available: <https://fidoalliance.org/specifications/overview/>.
- 1754 [2] FIDO Alliance. (n.d.). *FIDO Alliance* [Online]. Available: <https://fidoalliance.org/>.
- 1755 [3] StrongKey. (n.d.). *Home – StrongKey* [Online]. Available: <https://www.strongkey.com/>.
- 1756 [4] Magento, Inc. (n.d.). *eCommerce Platform | Best eCommerce Software for Selling Online*
1757 [Online]. Available: <https://magento.com/>.
- 1758 [5] Magento, Inc. (n.d.). *Magento Open Source* [Online].
1759 Available: <https://magento.com/products/open-source>.
- 1760 [6] A. Noor and A. de Leon. (2018, February 20). *FIDO U2F Integration for Magento 2* [Online].
1761 Available: <https://sourceforge.net/projects/magfido/?source=navbar>.
- 1762 [7] RSA. (n.d.). *RSA / Security Solutions to Address Cyber Threats* [Online].
1763 Available: <https://www.rsa.com/>.
- 1764 [8] RSA Security LLC. (n.d.). *Adaptive Authentication | Fraud Detection – RSA* [Online].
1765 Available: <https://www.rsa.com/en-us/products/fraud-prevention/secure-consumer-access>.
- 1766 [9] TokenOne. (n.d.). *TokenOne | Secure Authentication | Sydney* [Online].
1767 Available: <https://www.tokenone.com>.
- 1768 [10] Splunk Inc. (n.d.). *Splunk* [Online]. Available: <https://www.splunk.com/>.
- 1769 [11] Splunk Inc. (n.d.). *Splunk® Enterprise* [Online].
1770 Available: https://www.splunk.com/en_us/products/splunk-enterprise.html.
- 1771 [12] Splunk Inc. (n.d.). *Splunk® Universal Forwarder: Forwarder Manual* [Online].
1772 Available: <http://docs.splunk.com/Documentation/Forwarder/7.0.2/Forwarder/Abouttheuniversalforwarder>.
- 1774 [13] Splunk Inc. (n.d.). *Splunk DB Connect* [Online].
1775 Available: <https://splunkbase.splunk.com/app/2686/>.
- 1776 [14] Splunk Inc. (n.d.). *Splunk DB Connect Details* [Online].
1777 Available: <https://splunkbase.splunk.com/app/2686/#/details>.
- 1778 [15] Yubico. (n.d.). *Yubico NEO* [Online]. Available: <https://www.yubico.com/products/yubikey-hardware/yubikey-neo/>.

- 1780 [16] Yubico. (n.d.). *Yubico / YubiKey Strong Two Factor Authentication for Business and Individual Use*
1781 [Online]. Available: <https://www.yubico.com/>.
- 1782 [17] National Institute of Standards and Technology (NIST). (2017, June). *SP 800-63-3: Digital Identity*
1783 *Guidelines* [Online]. Available: <https://pages.nist.gov/800-63-3/>.
- 1784 [18] National Institute of Standards and Technology (NIST). (2013, May). *NISTIR 7298 Rev. 2: Glossary*
1785 *of Key Information Security Terms* [Online].
1786 Available: <https://www.nist.gov/publications/glossary-key-information-security-terms-1>.