## **NIST SPECIAL PUBLICATION 1800-38B**

# Migration to Post-Quantum Cryptography Quantum Readiness: Cryptographic Discovery

#### Volume B:

Approach, Architecture, and Security Characteristics of Public Key Application Discovery Tools

#### William Newhouse

Murugiah Souppaya National Institute of Standards and Technology Rockville, Maryland

#### William Barker

Dakota Consulting Silver Spring, Maryland

Chris Brown The MITRE Corporation Mclean, Virginia

Panos Kampanakis Amazon Web Services, Inc. (AWS) Arlington, Virginia

Marc Manzano SandboxAQ Palo Alto, California

## December 2023

PRELIMINARY DRAFT

David McGrew Cisco Systems, Inc. San Jose, California

Anne Dames IBM Yorktown Heights, New York

Vladimir Soukharev InfoSec Global Toronto, Canada

Philip Lafrance ISARA Corporation Ontario, Canada

Anthony Hu wolfSSL Seattle, Washington David Hook Keyfactor Independence, Ohio

Raul Garcia Microsoft Redmond, Washington

Evgeny Gervis SafeLogic, Inc. Palo Alto, California

Eunkyung Kim Changhoon Lee Samsung SDS Co., Ltd. Seoul, Republic of South Korea

This publication is available free of charge from <a href="https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms">https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms</a>



## 1 **DISCLAIMER**

- 2 Certain commercial entities, equipment, products, or materials may be identified by name or company
- 3 logo or other insignia in order to acknowledge their participation in this collaboration or to describe an
- 4 experimental procedure or concept adequately. Such identification is not intended to imply special sta-
- 5 tus or relationship with NIST or recommendation or endorsement by NIST or NCCoE; neither is it in-
- 6 tended to imply that the entities, equipment, products, or materials are necessarily the best available
- 7 for the purpose.
- 8 National Institute of Standards and Technology Special Publication 1800-38B Natl. Inst. Stand. Technol.
- 9 Spec. Publ. 1800-38B, 68 pages, (December 2023), CODEN: NSPUE2

#### 10 FEEDBACK

- 11 This initial draft offers: (1) a functional test plan that exercises the cryptographic discovery tools to
- 12 determine baseline capabilities; (2) a use case scenario to provide context and scope for our
- demonstration; (3) an examination of the threats addressed in this demonstration; (4) a multifaceted
- approach to the discovery process that most organizations can start today; and (5) a high-level
- 15 architecture based on our use case that integrates contributed discovery tools in our lab.
- 16 You can improve this initial public draft by submitting comments. We are always seeking feedback on
- 17 our publications and how they support our readers' needs. We are particularly interested in learning
- 18 from readers if this initial draft is helpful to you and what you want to see covered in future versions of
- 19 this publication.
- 20 The project will soon have a repository in <u>https://github.com/usnistgov</u> to provide the data shared in
- 21 this initial draft and relevant data created by the project in the future.
- 22 Comments on this publication may be submitted to: <u>applied-crypto-pqc@nist.gov</u>.
- 23 Public comment period: December 19, 2023 through February 25, 2024.
- All comments are subject to release under the Freedom of Information Act.

25	National Cybersecurity Center of Excellence
26	National Institute of Standards and Technology
27	100 Bureau Drive
28	Mailstop 2002
29	Gaithersburg, MD 20899
30	Email: <u>nccoe@nist.gov</u>

## 31 NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

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

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

46 <u>https://www.nist.gov.</u>

## 47 NIST CYBERSECURITY PRACTICE GUIDES

- 48 NIST Cybersecurity Practice Guides (Special Publication 1800 series) target specific cybersecurity
- 49 challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the
- 50 adoption of standards-based approaches to cybersecurity. They show members of the information
- 51 security community how to implement example solutions that help them align with relevant standards
- 52 and best practices, and provide users with the materials lists, configuration files, and other information
- 53 they need to implement a similar approach.
- 54 The documents in this series describe example implementations of cybersecurity practices that
- 55 businesses and other organizations may voluntarily adopt. These documents do not describe regulations
- 56 or mandatory practices, nor do they carry statutory authority.

#### 57 **KEYWORDS**

- 58 algorithm; cryptography; encryption; identity management; key establishment and management; post-
- 59 quantum cryptography; public-key cryptography; quantum-resistant; vulnerable cryptography discovery

#### 60 **ACKNOWLEDGMENTS**

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

Name	Organization
Cameron Bytheway	Amazon Web Services, Inc. (AWS)
Torben Hansen	Amazon Web Services, Inc. (AWS)

Name	Organization
Brian Jarvis	Amazon Web Services, Inc. (AWS)
Jake Massimo	Amazon Web Services, Inc. (AWS)
Wesley Rosenblum	Amazon Web Services, Inc. (AWS)
Martin Schaef	Amazon Web Services, Inc. (AWS)
Alex Weibel	Amazon Web Services, Inc. (AWS)
Avani Wildani	Cloudflare, Inc.
Natasha Eastman	Cybersecurity and Infrastructure Security Agency (CISA)
Garfield Jones	Cybersecurity and Infrastructure Security Agency (CISA)
Nancy Pomerleau	Cybersecurity and Infrastructure Security Agency (CISA)
Judith Furlong	Dell Technologies
Corey Bonnell	DigiCert
Avesta Hojjati	DigiCert
Boris Balacheff	HP, Inc.
Tommy Charles	HP, Inc.
Thalia Laing	HP, Inc.
Wai Choi	IBM
Christopher Meyer	IBM
Michael Osborne	IBM
Emily Ratliff	IBM
Kelsey Holler	Information Security Corporation

Name	Organization
Philip George	InfoSec Global
York Lam	InfoSec Global
Ninaa Persad	InfoSec Global
Julien Probst	InfoSec Global
Jackie Parker	ISARA Corporation
Rob Williams	ISARA Corporation
Atsushi Yamada	ISARA Corporation
Hubert Le Van Gong	JPMorgan Chase Bank, N.A.
John M. Chan	JPMorgan Chase Bank, N.A.
Roy Basmacier	Keyfactor
Alexander Scheel	Keyfactor
Ted Shorter	Keyfactor
Bryan Uhri	Keyfactor
Benjamin Rodes	Microsoft
Lily Chen	National Institute of Standards and Technology (NIST)
David Cooper	National Institute of Standards and Technology (NIST)
Daniel Eliot	National Institute of Standards and Technology (NIST)
Dustin Moody	National Institute of Standards and Technology (NIST)
Andy Regenscheid	National Institute of Standards and Technology (NIST)
Mike Jenkins	National Security Agency (NSA)

Name	Organization
Brendan Zember	National Security Agency (NSA)
Sean Morgan	Palo Alto Networks Public Sector, LLC
Graeme Hickey	PQShield
Michael Hutter	PQShield
Kris Kwiatkowski	PQShield
Ben Packman	PQShield
Axel Poschmann	PQShield
Jihoon Cho	Samsung SDS Co., Ltd.
Yoonchan Jhi	Samsung SDS Co., Ltd.
Hyojin Yoon	Samsung SDS Co., Ltd.
Hunhee Yu	Samsung SDS Co., Ltd.
Eric Ahlstrom	SandboxAQ
Jackson Beall	SandboxAQ
Nikolai Chernyy	SandboxAQ
Nick Hamilton	SandboxAQ
Tarun Sibal	SandboxAQ
Michael Toney	SandboxAQ
Daniel Cuthbert	Santander
Jaime Gomez	Santander
Robert Burns	Thales DIS CPL USA, Inc.

Name	Organization
Jane Gilbert	Thales Trusted Cyber Technologies
D'nan Godfrey	Thales Trusted Cyber Technologies
Gina Scinta	Thales Trusted Cyber Technologies
Kaitlyn Laohoo	The MITRE Corporation
Neil McNab	The MITRE Corporation
Jessica Walton	The MITRE Corporation
Volker Krummel	Utimaco
Lee E. Sattler	Verizon

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

Migration to Post Quantum Cryptography Technology Collaborators					
<u>Amazon Web Services, Inc.</u> (AWS)	Information Security Corporation	Samsung SDS Co., Ltd.			
<u>Cisco Systems, Inc.</u>	InfoSec Global	<u>SandboxAQ</u>			
<u>Cloudflare, Inc.</u>	ISARA Corporation	<u>Santander</u>			
Crypto4A Technologies, Inc.	JPMorgan Chase Bank, N.A.	SSH Communications Security Corp			
CryptoNext Security	<u>Keyfactor</u>	Thales DIS CPL USA, Inc.			
Cybersecurity and Infrastruc- ture Security Agency (CISA)	Microsoft	Thales Trusted Cyber Technolo- gies			
Dell Technologies	National Security Agency (NSA)	<u>Utimaco</u>			

Migration to Post Quantum Cryptography Technology Collaborators					
<u>DigiCert</u>	Palo Alto Networks Public Sector, LLC	Verizon			
<u>Entrust</u>	PQShield	VMware, Inc.			
HP, Inc.	QuantumXchange	wolfSSL			
IBM	SafeLogic, Inc.				

## 66 **DOCUMENT CONVENTIONS**

67 The terms "shall" and "shall not" indicate requirements to be followed strictly to conform to the

68 publication and from which no deviation is permitted. The terms "should" and "should not" indicate that

69 among several possibilities, one is recommended as particularly suitable without mentioning or

- 70 excluding others, or that a certain course of action is preferred but not necessarily required, or that (in
- the negative form) a certain possibility or course of action is discouraged but not prohibited. The terms
- 72 "may" and "need not" indicate a course of action permissible within the limits of the publication. The
- terms "can" and "cannot" indicate a possibility and capability, whether material, physical, or causal.

## 74 CALL FOR PATENT CLAIMS

75 This public review includes a call for information on essential patent claims (claims whose use would be

76 required for compliance with the guidance or requirements in this Information Technology Laboratory

77 (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication

or by reference to another publication. This call also includes disclosure, where known, of the existence

of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant

- 80 unexpired U.S. or foreign patents.
- 81 ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in writ-82 ten or electronic form, either:
- a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not
   currently intend holding any essential patent claim(s); or
- b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring
- to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft
- 87 publication either:
- under reasonable terms and conditions that are demonstrably free of any unfair discrimination;
   or
- without compensation and under reasonable terms and conditions that are demonstrably free
   of any unfair discrimination.

#### PRELIMINARY DRAFT

- 92 Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its
- behalf) will include in any documents transferring ownership of patents subject to the assurance, provi-
- sions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that
- 95 the transferee will similarly include appropriate provisions in the event of future transfers with the goal
- 96 of binding each successor-in-interest.
- 97 The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of
- 98 whether such provisions are included in the relevant transfer documents.
- 99 Such statements should be addressed to: applied-crypto-pqc@nist.gov.

## 100 **Contents**

101	1	Sun	nmary	/	1
102		1.1	Challe	nge	2
103		1.2	Outco	mes	2
104		1.3	Prepa	ring for the New Post-Quantum Standards	2
105			1.3.1	Public-Key Cryptographic Technologies	3
106		1.4	Demo	nstration Activity	5
107	2	Ηον	w to U	se This Guide	5
108	3	Арр	oroach	: The Migration to Post-Quantum Cryptography Project's	
109		Cor	ntribut	tion to Quantum Readiness	6
110		3.1	Audie	nce	7
111		3.2	Scope		7
112			3.2.1	Example Discovery Scenario for a Medium-Sized Business	7
113			3.2.2	Project Execution of Example Scenario in Our Lab	9
114			3.2.3	Vulnerable Cryptographic Algorithms	12
115		3.3	Termi	nology	14
116		3.4	Risk A	ssessment	14
117			3.4.1	Threats to Classical Cryptography	15
118			3.4.2	Vulnerabilities	16
119			3.4.3	Survey of Risk Methodologies	18
120	4	Arc	hitect	ure	.21
121		4.1	Archit	ecture Description	21
122			4.1.1	Protecting the Code Development Pipeline	21
123			4.1.2	Operational Systems and Applications	23
124			4.1.3	Transport Protocols and Network Services	24
125			4.1.4	Common Output Elements for Identifying Vulnerable Systems	25
126	5	Тес	hnolo	gies	. 33
127		5.1	Conso	rtium Members	33
128			5.1.1	Cisco	33
129			5.1.2	IBM	33
130			5.1.3	Infosec Global (ISG)	34
131			5.1.4	ISARA Corporation	34

132		5.1.5	Keyfactor	34
133		5.1.6	Microsoft	34
134		5.1.7	SafeLogic	35
135		5.1.8	Samsung SDS	35
136		5.1.9	SandboxAQ	35
137		5.1.10	wolfSSL	35
138	5.2	Produ	cts and Technologies	36
139	6 Fut	ure Pr	oject Considerations	40
140	Append	dix A	List of Acronyms	41
141	Append	dix B	References	44
142	Append	dix C	Discovery Platform Lab Functional Demonstration Plan	48
143	C.1	Use Ca	ase 1	48
144	C.2	Use Ca	ase 2	48
145	C.3	Use Ca	ase 3	49
146	C.4	Use Ca	ase 4	49
147	C.5	Use Ca	ase 5	50
148	C.6	Use Ca	ase 6	51
149	C.7	Use Ca	ase 7	51
150	C.8	Use Ca	ase 8	51
151	Append	dix D	IBM Z16 Remote Discovery Platform Functional	
152			Demonstration Plan	53
153	D.1	Use Ca	ase 1	53
154	D.2	Use Ca	ase 2	53
155	D.3	Use Ca	ase 3	54
156	D.4	Use Ca	ase 4	54
157	D.5	Use Ca	ase 5	55
158	D.6	Use Ca	ase 6	55

# 159 List of Figures

160	Figure 1 NIST cryptographic standards and guidelines
161	Figure 2 Conceptual vulnerable cryptography discovery workflow21

162	Figure 3 Conceptual CI Pipeline	22
163	Figure 4 Sensor data flow	23
164	Figure 5 Passive Network Discovery	24

## 165 List of Tables

166	Table 1 SSDF Practice Groups	.10
167	Table 2 Security Characteristic Mapping to SSDF Tasks	.10
168	Table 3 Scope of Vulnerable Algorithms	.13
169	Table 4 Submitted CWE	.17
170	Table 5 Software Development Components	.22
171	Table 6 Network-based discovery data elements	.25
172	Table 7 tool component property specification	.29
173	Table 8 runs.tool.rules property specification	.30
174	Table 9 runs.results property specification	.31
175	Table 10 runs.tools.rules.message Elements	.32
176	Table 11 Vulnerable Algorithm Identifiers	.32
177	Table 12 Products and Technologies	.36

## 178 **1 Summary**

- 179 In 2021, the National Institute of Standards and Technology (NIST) published CSWP-15, Getting Ready
- 180 for Post-Quantum Cryptography: Exploring Challenges Associated with Adopting and Using Post-
- 181 *Quantum Cryptographic Algorithms* [1], and the National Cybersecurity Center of Excellence (NCCoE)
- 182 initiated the Migration to Post Quantum Cryptography project [2] to develop practices and demonstrate
- 183 capabilities for easing migration from the current set of public-key cryptographic algorithms to
- 184 replacement algorithms that are resistant to quantum-computer-based attacks. The replacement
- algorithms were then identified and selected under a NIST post-quantum standards program initiated in
- 186 2016. Subsequently, on May 4, 2022, the White House issued a <u>National Security Memorandum</u> (NSM)
- 187 on "Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable
- 188 Cryptographic Systems." [3] This NSM reinforced the NCCoE's project priority by directing NIST to
- establish a Migration to Post-Quantum Cryptography Project "as the United States begins the multi-year
- 190 process of migrating vulnerable computer systems to quantum-resistant cryptography."
- 191 The advent of a cryptanalytically relevant quantum computer (CRQC) will render current standard
- 192 public-key cryptographic algorithms ineffective. These public-key algorithms are widely used in
- 193 protecting the integrity and confidentiality of digital information. Replacement of cryptographic
- algorithms is both technically and logistically challenging. It can take years or even decades to complete.
- 195 Working with the public and private sectors to address cybersecurity challenges posed by the transition
- 196 to quantum-resistant cryptography, the NCCoE is undertaking a practical demonstration of technology
- and tools that can assist organizations to develop a migration plan, sometimes called a quantum
- 198 readiness roadmap. Two demonstration activities were selected to enable and inform migration. One
- demonstration is focused on tools that discover where and how public key algorithms are being used;
- 200 the second involves experiments that measure the performance of the emerging quantum-resistant
- algorithms for security functions and protocols that are currently reliant on quantum-vulnerable
- 202 algorithms.
- 203 This volume covers the demonstration activities focused on the use of automated discovery tools. These
- 204 tools identify instances of quantum-vulnerable public-key algorithms that are widely deployed across an
- 205 organization to create a cryptographic algorithm inventory that will help an organization to develop its
- 206 migration roadmap.
- 207 A separate volume will be published on the development and improvement of migration strategies
- 208 enabled by demonstrating the interoperability and performance of post-quantum algorithm
- 209 implementations and sharing our tools and findings with standards developing organizations and
- 210 industry sectors reliant on cryptographic protection.
- 211 The reader should note this is a preliminary draft of the document, and while the content is considered
- to be stable, changes are expected to occur. There are gaps in the content and the overall document is
- 213 still incomplete. NIST welcomes early informal feedback and comments, which will be adjudicated after
- the specified public comment period. Organizations may consider experimenting with guidelines, with
- the understanding that they will identify gaps and challenges.

## 216 **1.1 Challenge**

- 217 A cryptographic inventory should be viewed as an asset that supports an organization's overall
- 218 cybersecurity risk management strategy. A cryptographic inventory is needed to apply cryptographic
- 219 policy across an organization's digital infrastructures; to react quickly to security issues involving
- 220 cryptography; and to inform transformations, such as migrating cryptography services to the cloud or
- 221 deploying post-quantum cryptography.
- 222 This publication shares insights and findings about the use of cryptographic discovery tools to create a
- 223 cryptographic inventory. This information may help organizations to start or improve their inventories.
- 224 This publication is one of the first focused on practices for creating, expanding, or using a cryptographic
- inventory to support migration to quantum-resistant cryptographic technologies. As an initial draft, it
- offers: (1) a functional test plan that exercises the cryptographic discovery tools to determine baseline
- 227 capabilities; (2) a use-case scenario to provide context and scope our demonstration; (3) the threats we
- will address in this demonstration; (4) a multifaceted approach for the discovery process that most
- organizations can start today; and (5) a high-level architecture based on our use case that integrates
- 230 contributed discovery tools in our lab.

### 231 **1.2 Outcomes**

- 232 This preliminary practice guide can help your organization with the following:
- 233 Identifying where and how public-key algorithms are being used in your information systems
- Raising internal awareness and understanding of risk-based cryptographic migration planning
   through the demonstration of tools, practice, and guidance
- Developing a risk-based playbook—involving people, processes, and technologies while
   connecting with existing risk management tools—for performing a migration to post-quantum
   cryptography

## **1.3 Preparing for the New Post-Quantum Standards**

- In recent years, there has been a substantial amount of research on *quantum computers* machines
   that exploit quantum mechanical phenomena to solve mathematical problems that are difficult or
   intractable for conventional computers. If large-scale quantum computers are ever built, they will be
   able to break many of the public-key cryptosystems currently in use. This would seriously compromise
- the confidentiality and integrity of digital communications on the Internet and elsewhere.
- 245 In 2016, NIST began developing standards for post-quantum cryptography (also called quantum-
- resistant cryptography) to enable cryptographic systems that are secure against both quantum and
- 247 classical computers and can interoperate with existing communications protocols and networks. New
- standards that specify key establishment and digital signature schemes that are designed to resist future
- attacks by quantum computers will be published in 2024.
- 250 Previous initiatives to update or replace installed cryptographic technologies have taken many years.
- 251 While the PQC standards are in development, organizations are encouraged to begin cryptographic
- discovery activities to identify the organization's current reliance on quantum-vulnerable cryptography.

## 253 1.3.1 Public-Key Cryptographic Technologies

254 Cryptographic technologies are used throughout government, industry, and academia to authenticate

the source and protect the confidentiality and integrity of information that we communicate and store.

256 Cryptographic technologies include a broad range of protocols, schemes, and infrastructures, but they

rely on a relatively small collection of cryptographic algorithms. They are mathematical functions that

transform data, generally using a variable called a *key* to protect information. The protection of these

- key variables is essential to the continued security of the protected data.
- 260 Public-key (also known as asymmetric) cryptographic algorithms require the originator to use one key
- and the recipient to use a different but related key. One of these asymmetric keys, the private key, must
- 262 be kept secret, but the other key, the public key, can be shared or even made public without degrading
- the security of the cryptographic process. *Symmetric algorithms* require a secret key to be shared by
- sender and receiver. The asymmetric (public-key) algorithms are used for data integrity (e.g., digital
- signature) and protected exchange of shared keys used by symmetric algorithms. The symmetric
- algorithms are more efficient for protection of bulk information, but the secure exchange and
- 267 establishment of shared keys generally requires protection by asymmetric cryptography.
- From time to time, the discovery of a cryptographic weakness, constraints imposed by dependent
   technologies, or advances in the technologies that support cryptanalysis make it necessary to replace a
- 270 cryptographic algorithm. Practical quantum computing is a technological advance that will make the
- 271 standard public-key algorithms now in use inadequate for provision of confidentiality of information,
- including secret keys. Most algorithms on which we depend are used worldwide in components of many
- 273 different communications, processing, and storage systems. Cryptography has become ubiquitous. It is
- 274 embedded in systems and components as different as operating systems, communications products,
- and Internet of Things devices in environments as different as space vehicles, automobiles, enterprise
- 276 data centers, household appliances, and embedded medical devices. Information system owners are
- often unaware of what components have cryptography embedded or how the cryptography is used in
- each case.
- Also, almost all information systems lack cryptographic agility—that is, they are not designed to
- 280 encourage support of rapid adaptations of new cryptographic primitives and algorithms without making
- significant changes to the system's infrastructure. As a result, an organization may not be able to easily
- alter or replace its cryptographic mechanisms when needed. While some components of some systems
- tend to be replaced by improved components on a relatively frequent basis (e.g., cell phones), other
- 284 components are expected to remain in place for a decade or more (e.g., components in electricity
- 285 generation and distribution systems).
- 286 Communications interoperability and records archiving requirements introduce additional constraints
- 287 on system components. As a general rule, cryptographic algorithms cannot be replaced until all
- 288 components of a system are prepared to process the replacement. Updates to protocols, schemes, and
- 289 infrastructures often must be implemented when introducing new cryptographic algorithms.
- 290 Consequently, algorithm replacement can be extremely disruptive and often takes a long time to
- complete. Unfortunately, the implementation of post-quantum public-key standards is likely to be even
- 292 more problematic than the past introduction of new classical cryptographic algorithms. In the absence

of significant implementation planning, it is likely to be decades before the community replaces most of
 the quantum-vulnerable public-key systems currently in use.

295 It would be ideal to have "drop-in" replacements for quantum-vulnerable algorithms (e.g., RSA and 296 Diffie-Hellman) for each of these purposes. Unfortunately, each of the new quantum-resistant 297 algorithms has at least one requirement for secure implementation that makes drop-in replacement 298 unsuitable sometimes. For example, the selected algorithms have large signature sizes, involve excessive 299 processing, require large public and/or private keys, require operations that are asymmetric between 300 sending and receiving parties and require the responder to generate a message based on the initiator's 301 public value, and/or involve other uncertainties with respect to computational results. Depending on the 302 algorithm and the operation using that algorithm, secure implementation may need to address issues 303 such as public-key validation, public-key reuse, decryption failure even when all parameters are 304 correctly implemented, and the need to select new auxiliary functions (e.g., hash functions used in 305 digital signature schemes). Even where secure operation is possible, performance and scalability issues

306 may demand significant modifications to protocols and infrastructures.

307 Different post-quantum algorithms can have significantly different performance characteristics and

- 308 implementation constraints (with respect to key sizes, signature sizes, resource requirements, etc.).
- 309 Consequently, different algorithms can be more suitable than others for specific applications. For
- 310 example, the signature or key size might not be a problem for some applications but can be
- 311 unacceptable for others. Some widely used protocols need to be modified to handle larger signatures or
- key sizes (e.g., using message segmentation). Implementations of new applications will need to
- accommodate the demands of post-quantum cryptography (PQC) and the schemes developed that
- 314 incorporate PQC for digital signatures and key establishment. In fact, PQC requirements may actually
- 315 shape some future application standards. The replacement of algorithms generally requires changing or
- 316 replacing cryptographic libraries, implementation validation tools, hardware that implements or
- 317 accelerates algorithm performance, dependent operating system and application code, communications
- devices and protocols, and user and administrative procedures. Security standards, procedures, and best
- 319 practice documentation are being changed or replaced, and the same will be needed for installation,
- 320 configuration, and administration documentation.
- 321 When a decision is made to replace an algorithm, it is necessary to develop a playbook that takes all of
- 322 the above factors into consideration. Some elements of the playbook are dependent on the
- 323 characteristics of both the algorithms being replaced and the replacement algorithms. Other elements
- needed for developing a detailed migration playbook can be determined before the replacement
- algorithms are selected and documented—for example, discovery and documentation of systems,
- applications, protocols, and other infrastructure and usage elements that use or are dependent on the
- 327 algorithms being replaced.
- 328 The first step in PQC migration planning is to identify where and for what purpose public-key
- 329 cryptography is currently being used. Public-key cryptography has been integrated into existing
- 330 computer and communications hardware, operating systems, application programs, databases,
- 331 communications protocols, key infrastructures, and access control mechanisms. Examples of public-key
- 332 cryptography uses include:
- Digital signatures used to provide source authentication and integrity authentication as well as
   support the non-repudiation of messages, documents, or stored data

- Identity authentication processes used to establish an authenticated communication session or
   authorization to perform a particular action
- Key transport of symmetric keys (e.g., key-wrapping, data encryption, message authentication keys) and other keying material (e.g., initialization vectors)

This volume focuses on meeting the challenge of discovering where and how an enterprise's public-key cryptography is used.

## 341 **1.4 Demonstration Activity**

To support the first step in migrating to post-quantum algorithms — identifying where and for what

- 343 purpose public-key cryptography is being used within an enterprise this project will document a
- 344 demonstration activity which leverages discovery tools and platforms. Identifying assets such as
- hardware and software as part of an inventory is a core function of the Cybersecurity Framework (CSF)
- and a basic pre-condition for any organization to effectively manage cybersecurity risk. This project
- 347 extends an existing inventory capability by identifying *cryptographic* assets and subsequently correlating
- 348 them to hardware, software, and services that have been previously inventoried.
- 349 This project takes a holistic approach for the discovery of cryptographic assets by using a combination of
- active and passive discovery techniques that are described in detail in this document. In summary, we
- focus on vulnerable cryptographic algorithms within an organization's digital systems and codebases,
- 352 and use the discovered cryptographic assets to support the prioritization of their replacement. Lab
- demonstrations aim to leverage existing enterprise tools (code repositories, Governance, Risk, and
- 354 Compliance [GRC] platforms, SIEM solutions, configuration management databases, etc.) and will use
- 355 multiple commercially available and open-source discovery tools to achieve project objectives.
- 356 We have selected an interchange format to demonstrate using the output from the discovery tools for
- 357 post-discovery risk analysis. We invite implementers to use the interchange format as part of an
- 358 organization's approach to leveraging discovery results.

## 359 **2 How to Use This Guide**

360 This NIST Cybersecurity Practice Guide focuses on one identified practice to ease migration from the

- 361 current set of public-key cryptographic algorithms to replacement algorithms that are resistant to
- 362 quantum computer-based attacks. It is an initial, preliminary public draft that shares what has been
- 363 learned to-date regarding the use of automated cryptographic algorithm discovery tools.
- 364 Readers in IT roles such as security architects and system administrators who are responsible for
- 365 monitoring the state of implementation of PQC standards in technology to inform their migration plan
- may also want to refer to a separate document in this series: NIST SP 1800-38C: *Quantum-Resistant*
- 367 Cryptography Technology Interoperability and Performance Report. That report summarizes the
- 368 outcomes from the Performance and Interoperability Workstream testing, in which we identified the
- 369 challenging problems and bottlenecks that integrators will face when transitioning systems to post-
- 370 quantum-ready algorithms.
- This guide and NIST SP 1800-38C were preceded by NIST SP 1800-38A: *Executive Summary, Migration to*
- 372 Post-Quantum Cryptography: Preparation for Considering the Implementation and Adoption of Quantum

- 373 Safe Cryptography. Business decision makers, including chief information security and technology
- officers should refer to this document to understand the drivers for this project and the challenges we
- 375 plan to address.
- 376 This guide uses a monospace typeface for JavaScript Object Notation (JSON) examples.

# 377 3 Approach: The Migration to Post-Quantum Cryptography 378 Project's Contribution to Quantum Readiness

- A successful post-quantum cryptography migration will take time to plan and conduct and can be described by a multistep approach. The approach is consistent with the activities below from the *Quantum-Readiness: Migration to Post-Quantum Cryptography factsheet* [4] created in partnership with the Department of Homeland Security's Cybersecurity & Infrastructure Security Agency (CISA) and the
- 383 National Security Agency (NSA).
- 384 1. Establish a Quantum-Readiness Roadmap
- 385 2. Prepare a Cryptographic Inventory
- 386 3. Discuss Quantum Safe Roadmaps with Technology Vendors
- 387 4. Determine Supply Chain Quantum-Readiness
- 388 The findings of the discovery activity and subsequent interoperability and performance activity can be
- used in refining organizations' quantum-readiness roadmaps as the scope, distribution, characteristics,
   and requirements of their quantum-vulnerable implementations emerge.
- 391 One aspect that the NCCoE Migration to Post-Quantum Cryptography Project approach focuses on is
- automated tools that address step 2, *Prepare a Cryptographic Inventory*. This step should be taken to
- identify vulnerable cryptographic algorithms used by an organization. Automated tools should identify
- the cryptographic algorithms used in hardware and software modules, libraries, and embedded code.
- Automated tools should also identify the cryptographic algorithms currently used by an enterprise to support cryptographic key establishment and management underlying the security of cryptographically
- 397 protected information and access management processes, as well as algorithms used to protect the
- 398 source and content integrity of data at rest, in transit, and in use.
- After the vulnerable public-key cryptography components and associated assets in the enterprise are identified, the next objective of the project is to prioritize those components that need to be considered first in the migration using a risk management methodology informed by the sensitivity and criticality of the information being protected over time.
- 403 In 2021 NIST invited organizations to provide letters of interest describing products and technical
- 404 expertise to support the Migration to Post-Quantum Cryptography project. This notice was the initial
- 405 step for the NCCoE in collaborating with technology companies to address cybersecurity challenges
- 406 identified under this project. Commercial and open-source software and hardware technology providers
- 407 responded with technology and tools that can provide organizations a head start to migrate to post-
- 408 quantum cryptography and signed Cooperative Research and Development Agreement (CRADA) with

- 409 NIST to become project collaborators. The project collaborators first met in June 2022 and established
- 410 two workstreams, each of which focuses on a specific aspect of migration to PQC.
- 411 This initial draft incorporates the contributions of consortium members in terms of their technologies
- 412 and also offers initial strategies on how to leverage discovery to prioritize migration technologies.
- 413 Updates to this document will be made when additional demonstrations are completed. Section 6
- 414 discusses areas we have identified for future work in this workstream.

#### 415 **3.1 Audience**

- 416 This document shares insights for medium to large enterprises using cryptographic technologies
- 417 (products and services including cloud services) that include quantum-vulnerable public-key
- 418 cryptographic algorithms and for companies that supply services and products that employ quantum-
- 419 vulnerable public-key cryptographic algorithms. Within these audiences, we focus on individuals
- 420 supporting those responsible for system provisioning, maintenance, and security. This audience may
- 421 include business decision makers such as authorizing officials and data owners and operators who are
- 422 concerned with cryptographic risk analysis capabilities.
- 423 Additionally, U.S. Federal agencies may find the capabilities and products that we demonstrate useful
- 424 for assigned tasks in the November 2022 Office of Management and Budget (OMB) Memorandum on
- 425 *Migrating to Post-Quantum Cryptography* (M-23-02) [5]. This memorandum includes steps for agencies
- 426 to take to transition to PQC.

### 427 **3.2 Scope**

- 428 This publication includes an example scenario to frame the challenge of cryptographic discovery from
- 429 the perspective of an organization beginning its migration to PQC. The scenario also scopes the desired
- 430 outcomes. In Section 3.2.2, we discuss the approaches we will take to realize the outcomes described in
- 431 Section 3.2.1. In Section 3.2.3, we specify the cryptographic algorithms that are considered vulnerable in
- 432 the context of this demonstration.

## 433 3.2.1 Example Discovery Scenario for a Medium-Sized Business

- 434 The following scenario describes a fictional company and its approach to migration to PQC.
- 435 ZetaMSB Inc. (Zeta) is a medium-sized business that provides IT consultancy services regionally. Zeta has
- 436 approximately 1000 employees with two office locations; however, a quarter of the employees are
- 437 primarily remote workers. Further, each employee is issued an organizationally managed laptop to
- 438 access proprietary Zeta data to support IT consultancy clients. This proprietary knowledge base is stored
- on a mixture of leased bare metal servers that Zeta manages and cloud services. Zeta also has a
- 440 development team that creates software and services which support its clients and Zeta internal441 processes.
- 442 Zeta's CISO has recently been tasked with coming up with a strategy to protect Zeta's critical
- 443 information and its competitive edge far into the future. To that end, the forward-looking CISO has
- started to read about the potential future impacts of quantum computing specifically the threats to
- the integrity and confidentiality of data. The CISO consequently becomes concerned about Zeta's
- 446 current cryptographic posture and how competitors or other bad actors could use cryptanalytically

- relevant quantum computers (CRQCs) to exfiltrate proprietary data while stored on Zeta's servers or intransit to their remote workers.
- 449 Zeta has a fairly mature cybersecurity program that leverages the NIST Cybersecurity Framework (CSF), 450 and as such, periodically identifies future threats as part of risk management. As a result, the CISO 451 decides to develop a plan that will address CRQC threats to their organization by incorporating into their 452 long-term cyber strategy the migration of their systems and services that use vulnerable cryptography to 453 quantum-resistant cryptography, hereafter referred to as post-quantum cryptography (PQC). The CISO 454 adopts a phased approach covering multiple years due to their products having dependencies on third-455 party software that does not yet incorporate PQC. However, they conclude that the initial phase could 456 be started immediately by expanding current hardware and software inventory processes to identify 457 quantum-vulnerable cryptography in use within Zeta.
- 458 In consultation with stakeholders across the organization, the CISO derives a three-pronged approach 459 for identifying the uses of vulnerable cryptography within the organization. First, Zeta's newly formed 460 migration team will identify digital cryptographic assets on operational systems fielded to Zeta 461 employees and on systems and devices that are managed by Zeta's IT department. These include 462 operating systems, communications servers and controllers for wired and wireless internal and external 463 networks, third-party application software, application software developed in-house, security systems 464 such as firewalls, and key generation and management systems. In addition, employees may use cloud 465 services that are not managed by Zeta's IT department, but that provide cryptographic protection for Zeta information. The migration team concludes the best way to identify these services is to capture 466 467 traffic from operational networks and examine it for instances of vulnerable cryptography usage with a focus on the information that traverse the public internet. Finally, to align with the organization's "shift-468 469 left" methodology, the migration team recommends identifying vulnerable cryptography used by their 470 DevOps team in their codebases. This would enable the DevOps team to update their codebases with 471 post-quantum cryptography. After documenting this approach, the CISO directs the migration team to 472 automate as much of the discovery process as possible, leveraging existing enterprise services if feasible. 473 The Zeta migration team proceeds to perform market research on vulnerable cryptography discovery
- tools available in commercial and open-source products. They conclude that multiple products may be
  necessary to achieve their desired outcomes. That presents a challenge to the team, however, because
  there is a discrete reporting format for each discovery platform to provide a common data format for
- 477 collecting the data. As a result, to support automation, each discovery platform's reporting format will
- 478 need to be normalized into a common format in order to integrate into existing security event
- 479 aggregation and reporting tools.
- 480 Resources for the migration task are limited at Zeta because of other planned IT modernization efforts
- 481 occurring in the same timeframe. As a result, the CISO must follow a risk-based approach to prioritize
- 482 which systems will be migrated first after the inventory has been completed. Fortunately, Zeta has
- identified processes and knowledge assets that are critical to the viability of its business by way of past
- 484 CSF activities. These processes and assets will be mapped to operational systems and services identified
- 485 by the discovery platforms and given higher priority in the migration effort.

## 486 3.2.2 Project Execution of Example Scenario in Our Lab

487 To create an implementation in our NCCoE Migration to PQC lab environment for the scenario above,

488 the NCCoE team is collaborating with consortium members to create an appropriate environment

- 489 including vulnerable algorithm discovery tools, representative enterprise infrastructure, and test data.
- 490 The enterprise infrastructure includes virtual and hardware endpoint computing devices, such as
- 491 laptops, with common software installed such as web browsers and business productivity software. This
- 492 will be complemented by servers that leverage cryptographic services, such as HTTPS and Secure Shell
- 493 (SSH). The DevOps environment will be built using common components that are described later in this
- 494 document.
- 495 Finally, test data will include captured network traffic that has protocols using known vulnerable
- 496 cryptography. These captures can support scenarios in which network traffic is "replayed" to simulate an
- 497 operational network, where network-based sensors for discovering vulnerable cryptography can
- 498 perform an analysis in real time. It also supports an alternative discovery scenario where an organization
- has historical captures that are uploaded to the discovery tool for analysis. The captures will function as
- 500 repeatable "known answer tests" where the inputs (vulnerable cryptography) are known and should be
- 501 detected by the discovery tools.
- Additional test data will include a representative code base that leverages classical cryptography that is
   scanned for vulnerable cryptography usage in two scenarios:
- 504 Code that exists in a cloud and on-premises code repository
- 505 A project in active development where code is scanned before each pull or merge request
- 506 The first scenario allows for analysis of existing codebases, while the second detects and flags potential 507 vulnerabilities before developer code is merged into the main code branch. As a result, the flagged code 508 is reviewed and the individual who is responsible for maintaining the repository directs the developer to 509 modify the pull request such that quantum-resistant code (or libraries) are implemented. Once the 510 changes are made to the satisfaction of the maintainer, the pull request is approved, and the DevOps
- 511 process proceeds.
- 512 With the preceding context, we've decomposed the discovery of vulnerable algorithms into the 513 following use cases:
- Vulnerable cryptography used in code, compiled binaries, or dependencies during a continuous
   integration/continuous delivery (CI/CD) development pipeline
- Vulnerable cryptography used in assets on end-user systems and servers, to include applications
   and associated libraries
- Vulnerable cryptography used in network protocols, enabling traceability to specific systems
   using active scanning and historical traffic captures
- 520 The remainder of this section discusses each use case, including the scope and existing best practices 521 that will be leveraged.

## 522 3.2.2.1 Protecting the Code Development Pipeline

- 523 Protecting the code development pipeline in the context of migration to PQC is closely related to the
- 524 outcomes listed for the NCCoE DevSecOps project [6], which describes DevSecOps as helping to "ensure
- 525 that security is addressed as part of all DevOps practices by integrating security practices and
- 526 automatically generating security and compliance artifacts throughout the processes and environments,
- 527 including software development, builds, packaging, distribution, and deployment." The DevSecOps
- 528 project leverages the Secure Software Development Framework (SSDF) [7] to define the tasks that can
- 529 be implemented as part of a DevSecOps approach. Similarly, we align our migration tasks with the
- 530 framework provided by the SSDF.
- 531 The SSDF defines secure software development practices that are organized into four practice groups
- shown in Table 1. This demonstration will focus on the Protect the Software (PS), Produce Well-Secured
- 533 Software (PW), and Respond to Vulnerabilities (RV) groups.
  - Practice Group Description Prepare the Organiza-Organizations should ensure that their people, processes, and technology tion (PO) are prepared to perform secure software development at the organization level. Many organizations will find some PO practices to also be applicable to subsets of their software development, like individual development groups or projects. **Protect the Software** Organizations should protect all components of their software from tam-(PS) pering and unauthorized access. **Produce Well-Secured** Organizations should produce well-secured software with minimal security Software (PW) vulnerabilities in its releases. **Respond to Vulnerabil-**Organizations should identify residual vulnerabilities in their software reities (RV) leases and respond appropriately to address those vulnerabilities and prevent similar ones from occurring in the future.
- 534 Table 1 SSDF Practice Groups

535 Table 2 maps our project security characteristics to tasks in the SSDF.

## 536 Table 2 Security Characteristic Mapping to SSDF Tasks

SSDF Task ID	Task Description	Migration to PQC Characteristic
PS.3.2	Collect, safeguard, maintain, and share provenance data for all components of each software release (e.g., in a software bill of materials [SBOM]).	Creation of a cryptography bill of materials (CBOM)
PW.6.1	Use compiler, interpreter, and build tools that offer fea- tures to improve executable security.	Inspection of source code, de- pendent libraries, and contain- erized software via continuous integration actions

SSDF Task ID	Task Description	Migration to PQC Characteristic
PW.7.2	Perform the code review and/or code analysis based on the organization's secure coding standards, and record and triage all discovered issues and recommended re- mediations in the development team's workflow or is- sue tracking system.	Inspection of source code via continuous integration actions
RV.1.2	Review, analyze, and/or test the software's code to identify or confirm the presence of previously unde-tected vulnerabilities.	Inspection of existing source code in a repository

537 As noted in Table 2, this demonstration will experiment with the creation of a cryptographic bill of 538 materials (CBOM). CBOMs are related to and build upon a software bill of materials (SBOM), which is a 539 digital record containing the details and supply chain relationships of various components used in 540 building software. A CBOM extends an SBOM by defining an object model to describe cryptographic 541 assets and their dependencies. CBOMs, when integrated into reporting and software development 542 practices, have the potential to enable organizations to manage and report usage of cryptography, 543 benefiting asset inventory activities. The CBOM model is still developing and may not address every use 544 case for all organizations, and in future drafts of this publications we will describe our approach to 545 automating the creation of CBOMs or other formats as well as incorporating CBOMs into a broader asset inventory effort. 546

## 547 3.2.2.2 Operational Systems and Applications

548 The list below is a non-exhaustive set of typical cryptographic assets that are in scope for this 549 demonstration based on the capabilities exercised in the lab environment thus far. Note that individual 550 discovery platforms may have extended capabilities. The list is sectioned into two categories: executable 551 and non-executable file objects. Executables are components of runnable software, and non-552 executables are filesystem objects referenced by executables. Non-executables include keystores and 553 other formats that may contain both public and private asymmetric keys. This example implementation 554 detects and reports public/private keypairs which implement one of the vulnerable algorithms listed in 555 Table 3. Note that formats may support password-based encryption of private keys, which may affect the fidelity of the discovery results. 556 557 Executables 558 Application binaries

- Cryptographic libraries
- 560Java archives
- 561 Non-executables
- Keystores: <u>PKCS#12</u>, Java Keystores, <u>Key Data Sets</u>
- Other key formats: <u>OpenPGP Keys</u>, <u>X.509 Certificates</u>, <u>OpenSSH Keys</u>, <u>PKCS#1</u>, <u>PKCS#8</u>

## 564 3.2.2.3 Transport Protocols and Network Services

565 The Canadian National Quantum Readiness Best Practices and Guidelines document [8] suggests a 566 number of technology protocols that organizations should scan for vulnerable cryptography. For this demonstration, based on exercising the discovery platforms in our lab environment, our initial scope is 567 568 three core protocols—Transport Layer Security (TLS), Secure Shell (SSH), and Internet Protocol Security 569 (IPsec). We chose these protocols due to their ubiquitous enterprise usage. TLS is a secure tunneling 570 protocol widely implemented in browsers and web servers. TLS is used to protect multiple application 571 layer protocols, including HTTPS, SMTP (via STARTTLS), IMAPS, POP3S, and Microsoft Remote Desktop 572 Protocol (RDP). Contemporary versions of TLS make use of X.509 certificates and corresponding private 573 keys which rely on quantum-vulnerable signature and key exchange algorithms. The SSH protocol enables a user or an automated process to remotely access the shell of a server system. Like TLS, 574 575 asymmetric cryptographic keys are used to authenticate and establish encrypted SSH connections. 576 Finally, IPsec enables cryptographic-based security for IPv4 and IPv6 and is typically used in virtual 577 private networks (VPNs).

## 578 3.2.3 Vulnerable Cryptographic Algorithms

579 We have based our vulnerable algorithm discovery on the NIST Post-Quantum Cryptography

580 standardization process. Figure 1 represents the NIST cryptographic standards and guidelines, which

581 includes public key (asymmetric) signature and key establishment schemes. Symmetric cryptographic

primitives, such as block ciphers and hash functions, are not as drastically impacted by the advent of a

583 (cryptographically relevant) quantum computer [9] and are not in scope for this project. This scoping

584 provides the basis to develop a discovery policy across disparate platforms.

585 Figure 1 NIST cryptographic standards and guidelines



- 586 In this demonstration, the algorithms listed in Table 3 are considered vulnerable.
- 587 Table 3 Scope of Vulnerable Algorithms

Algorithm	Function	Specification
Elliptic Curve Diffie Hellman (ECDH) Key Exchange	Asymmetric algorithm for digital signa- tures/key exchange	NIST SP 800- 56A/B/C
Menezes Qu Vanstone (MQV) Key Exchange	Asymmetric algorithm for key exchange	NIST SP 800- 56A/B/C
Elliptic Curve Digital Signature Algo- rithm (ECDSA)	Asymmetric algorithms for digital signa- tures/key exchange	FIPS PUB 186-5
Diffie Hellman (DH) Key Exchange	Asymmetric algorithms for digital signatures	IETF RFC 3526
RSA Encryption Algorithm	Asymmetric algorithms for digital signa- tures/key establishment	SP 800-56B Rev. 2
RSA Signature Algorithm	Asymmetric algorithms for digital signa- tures/key exchange	FIPS PUB 186-5

Algorithm		Function	Specification
Digital Signatu	re Algorithm	Asymmetric algorithms for digital signa- tures/key exchange	FIPS PUB 186-5
Edwards-curve gorithm (EdDS	e Digital Signature Al- A)	Asymmetric algorithms for digital signatures	FIPS PUB 186-5

## 588 **3.3 Terminology**

- 589 This document and future volumes will align with the terminology defined in IETF's Terminology for
- 590 Post-Quantum Traditional Hybrid Schemes [10]. The document is intended to be used as a reference and
- to ensure consistency and clarity across different protocols, standards, and organizations. In particular,
- the remainder of this document references the following terms, followed by their definitions.
- 593 **Traditional or Classical Cryptographic Algorithm:** An asymmetric cryptographic algorithm based on 594 integer factorization, finite field discrete logarithms, or elliptic curve discrete logarithms.
- 595 **Post-Quantum Cryptographic Algorithm:** An asymmetric cryptographic algorithm that is believed to be 596 secure against attacks using quantum computers as well as classical computers. The algorithms
- 597 identified as post-quantum algorithms are defined in:
- FIPS 203 (Draft), Module-Lattice-Based Key-Encapsulation Mechanism Standard [11], specifies a cryptographic scheme called "Module Learning with errors Key Encapsulation Mechanism, or
   MLWE-KEM," which is derived from the CRYSTALS-KYBER algorithm.
- FIPS 204 (Draft), Module-Lattice-Based Digital Signature Standard [12], specifies the "Module
   Learning with Errors Digital Signature Algorithm, or ML-DSA," which is based on the CRYSTALS Dilithium submission.
- FIPS 205 (Draft), Stateless Hash-based Digital Signature Standard [13], specifies the "Stateless
   Hash-based Digital Signature Algorithm, or SLH-DSA," which is based on the SPHINCS+
   submission.
- 607 These initial quantum-resistant algorithms and stateless hash-based signature standard will augment
- the public-key cryptographic algorithms already contained in FIPS 186-5, Digital Signature Standard
- 609 (DSS) [14], as well as SP 800-56A Revision 3, Recommendation for Pair-Wise Key-Establishment Schemes
- 610 Using Discrete Logarithm Cryptography [15], and SP 800-56B Revision 2, Recommendation for Pair-Wise
- 611 Key Establishment Using Integer Factorization Cryptography [17]. Additional quantum-resistant
- algorithms will be defined as the standardization effort progresses.
- Finally, several members of our consortium have moved to use the terms "quantum-safe" or "quantumready" as opposed to "post-quantum" to refer to the algorithms and technologies that are protected against cryptographically relevant quantum computer (CRQC) attacks. This document continues to use
- 616 "post-quantum" and will be updated as these terms evolve.

## 617 3.4 Risk Assessment

618 In this section we discuss threats to classical algorithms and the specific attacks that researchers have 619 determined are feasible when a CRQC is operational. Next, we consider the vulnerabilities that a CRQC

- 620 will introduce with a discussion of a proposed Common Weakness Enumeration (CWE) that
- 621 organizations could potentially use during their internal prioritization efforts. Finally, we discuss existing
- 622 research efforts to develop a risk methodology that have influenced this project's approach to
- 623 demonstrating a migration prioritization list.

## 624 3.4.1 Threats to Classical Cryptography

At this time, there are two quantum algorithms that are projected to be used in attacks against data

- 626 protection schemes such as digital encryption and signatures. The first is Shor's algorithm, which
- 627 provides an efficient method for computing the discrete-logarithm problem and the elliptic curve
- 628 discrete-logarithm problem, as well as the problem of factoring large integers, thus breaking current key
- 629 exchange, digital signature, and public key encryption methods that are based on asymmetric
- 630 cryptography. Because of Shor's algorithm, new cryptographic algorithms are needed that are resistant
- to attacks that can be launched from both classical and quantum computers. The second quantum
- algorithm is Grover's algorithm, which can be used to speed up the identification of a secret key in a key
- address space. To thwart this type of attack, strong encryption algorithms are needed with keys whose
- 634 key address space is large enough to be considered not vulnerable.
- Digital encryption and signature schemes are widely used in transport protocols like (D)TLS, IKEv2/IPsec,
- 636 QUIC, and SSH. They all include a key exchange phase where the peers exchange asymmetric keys which
- enable them to establish a shared secret using ECDH. They then proceed to derive a symmetric key
- 638 which is used to symmetrically encrypt data exchanged between the peers. The Authentication phase
- 639 includes providing an asymmetric signature of a transcript of the exchanged data which proves that the
- 640 peer signed this data with its private key. The corresponding public key is usually included with the
- 641 identity of the peer and is authenticated by using PKI or other methods. That way the peers can verify
- 642 they are talking to the peer with the expected identity who holds the expected public key.
- 643 A quantum computer could break the asymmetric schemes for key exchange and signing. Shor's
- algorithm could break ECDH, which means that a quantum-capable threat actor could recover the
- 645 symmetric key used to encrypt data. Specifically, for encryption, although there is no CRQC today,
- someone could be storing data encrypted with TLS or other protocols today in order to retroactively
- 647 decrypt them in the future with a quantum computer, more commonly known as *harvest and decrypt* or
- 648 *store now decrypt later* attacks.
- 649 Consequently, protection of data is needed at rest, in transit, and in use. As a reference we have listed 650 below a variety of techniques an adversary can use to enable the harvesting of encrypted data today, as 651 documented in MITRE's ATT&CK framework [16], a globally-accessible knowledge base of adversary 652 tactics and techniques based on real-world observations. Further, it may not be necessary for the 653 adversary to own their own CRQC, as it is expected that adversaries will have cloud access to CRQCs in 654 the future to conduct their attacks.
- Adversary-in-the-Middle Adversaries may attempt to position themselves between two or more networked devices using an adversary-in-the-middle (AitM) technique to support followon behaviors such as network sniffing or transmitted data manipulation. By abusing features of common networking protocols that can determine the flow of network traffic (e.g., ARP, DNS, LLMNR), adversaries may force a device to communicate through an adversary-controlled system so they can collect information or perform additional actions.

- Automated Collection Once established within a system or network, an adversary may use
   automated techniques for collecting internal data. Methods for performing this technique could
   include use of a command and scripting interpreter to search for and copy information fitting
   set criteria such as file type, location, or name at specific time intervals. In cloud-based
   environments, adversaries may also use cloud APIs, command line interfaces, or extract,
   transform, and load (ETL) services to automatically collect data. This functionality could also be
   built into remote access tools.
- **Data from Cloud Storage** Adversaries may access data from improperly secured cloud storage.
- Data from Local System Adversaries may search local system sources, such as file systems and configuration files or local databases, to find files of interest and sensitive data prior to exfiltration.
- Data from Network Shared Drive Adversaries may search network shares on computers they
   have compromised to find files of interest. Sensitive data can be collected from remote systems
   via shared network drives (host shared directory, network file server, etc.) that are accessible
   from the current system prior to exfiltration. Interactive command shells may be in use, and
   common functionality within cmd may be used to gather information.
- For a more in-depth discussion on threats to existing security architectures which employ vulnerable
  cryptography, we recommend readers review the content produced by other organizations such as the
  Accredited Standards Committee's *Quantum Computing Risks to the Financial Services Industry* [18] and
  the Quantum-Readiness Working Group's (QRWG) of the Canadian Forum for Digital Infrastructure
  Resilience (CFDIR) *Best Practices and Guidelines* [8].

## 682 3.4.2 Vulnerabilities

- 683 During some workstream discussions, the project team identified the need to report quantum
- 684 vulnerabilities to existing upstream risk management systems in a standardized format. We chose
- 685 Common Weakness Enumeration (CWE) [19], a community-developed list of software and hardware
- 686 weakness types, due to its widespread use in industry. It serves as a common language and a baseline
- 687 for weakness identification, mitigation, and prevention efforts.
- 688 CWEs in the cryptographic domain are currently categorized as a failure in a protection mechanism 689 (CWE-693). These include existing CWEs (CWE-327: Use of a Broken or Risky Cryptographic Algorithm
- 690 and CWE-326: Inadequate Encryption Strength, for example) that address weak algorithms.
- The CWE concept is targeted around existing vulnerabilities. Future vulnerabilities such as those coming from a quantum threat do not appear to easily fit into this concept. Weaknesses only manifest if certain conditions are met, leading to the potential of many false positives. Nevertheless, a major benefit in using the CWE approach would be to signal future issues in a way that allows remediation development to be planned for in advance as a feature, rather than later as a remediation action. This could be handled in different ways within the CVE scheme:
- 697 through the use of additional attributes,
- 698 through defining a separate block of codes for future weaknesses, or
- 699 through logic in evaluating additions to existing blocks.

- 700 During project workstream meetings, a proposed CWE was created (see Table 4) to capture the
- 701 particular weakness of algorithms that are quantum-vulnerable but are otherwise safe. Due to the
- nature of this weakness, not all proposal elements were applicable. It was proposed to be a Child of
- 703 CWE-327 (CWE-327: Use of a Broken or Risky Cryptographic Algorithm) with this description:
- Cryptographic algorithms are used to protect the confidentiality and authenticity stored in or transmitted through an untrusted medium. Quantum-vulnerable algorithms will be exploitable if or when a cryptographically relevant quantum computer (CRQC) is built, in which case sensitive information could be exposed, data could be undetectably modified, and the identities of users and devices could be spoofed, among other impacts.
- Data that is encrypted by a quantum-vulnerable encryption or key establishment algorithm and has a long data protection period is especially at risk, because an adversary can store encrypted data today and decrypt it later with a CRQC. Such an attack is called store-now-decrypt-later, and it creates significant risk for long-term data confidentiality. Additionally, an adversary with a CRQC could forge signatures for quantum-vulnerable algorithms, creating a risk for signatures with a long data protection period, especially those used to sign software images in devices where the root of trust cannot be upgraded.
- 716 Table 4 Submitted CWE

Elements to Include	Value
Name	Use of a quantum-vulnerable algorithm
<u>Summary</u>	The use of a quantum-vulnerable algorithm is a risk that may result in the exposure of sensitive information if and when a cryptographically relevant quantum computer becomes available.
Extended Description	The use of a quantum-vulnerable algorithm creates a risk if a cryptographically relevant quantum computer (CRQC) becomes available. A CRQC could threaten all public key algorithms based in the integer factorization and (elliptic curve) discrete logarithm problem and compromise whatever data has been protected. In particular, an adversary storing encrypted data today could theoretically decrypt them later with a CRQC. Such an attack is called harvest-now-decrypt-later, and it creates significant risk for long-term data confidentiality. Additionally, an adversary with a CRQC could forge signatures for quantum-vulnerable algorithms, creating risk for signatures with a long data protection period.
Modes of In- troduction	Architecture and Design
Potential Mitigations	Use standardized quantum-safe asymmetric algorithms or hybrid schemes that incorporate both post-quantum and traditional asymmetric algorithms.
Common Conse- quences	Confidentiality Technical Impact: Read Application Data The confidentiality of sensitive data may be compromised by the use of a quantum- vulnerable cryptographic algorithm.

#### PRELIMINARY DRAFT

Elements to Include	Value
	Technical Impact: Modify Application Data The integrity of sensitive data may be compromised by the use of a quantum-vulnera- ble cryptographic algorithm.
	Accountability Non-Repudiation Technical Impact: Hide Activities
	If the cryptographic algorithm is used to ensure the identity of the source of the data (such as digital signatures), then the use of a quantum-vulnerable algorithm will compromise this scheme and the source of the data cannot be proven.
Applicable Platforms	ALL
Demonstra- tive Exam- ples	N/A
Observed Ex- amples	N/A
<u>Relation-</u> <u>ships</u>	N/A
<u>References</u>	The references provided were to NSA Commercial National Security Algorithm Suite 2.0 ( <u>https://media.defense.gov/2022/Sep/07/2003071834/-1/-</u> <u>1/0/CSA_CNSA_2.0_ALGORITHMSPDF</u> ) and NIST Post-Quantum Cryptography ( <u>https://csrc.nist.gov/projects/post-quantum-cryptography</u> ).

717 At the time of writing, the proposal has been submitted to the CWE team for feedback.

## 718 3.4.3 Survey of Risk Methodologies

- 719 This project aims to leverage a reusable, generalized approach that could facilitate the migration process
- 720 for many organizations. The following subsections describe several risk methodologies specifically to
- address the threat of a CRQC at the time of writing. The methodologies are similar in that they all
- describe a cryptographic asset discovery activity from which a prioritization list can be derived.
- However, some of the methodologies describe sector-specific guidance to perform a risk assessment.

## 724 *3.4.3.1 Mosca's Theorem*

- 725 Mosca's Theorem is commonly referenced as the starting point for organizations initiating a migration to
- post-quantum algorithms. Once the public-key cryptography components and associated assets in the
- 727 enterprise are identified, the next element of the scope of the project is to prioritize those components
- that need to be considered first in the migration using a risk management methodology informed by
- 729 Mosca's Theorem and other recommended practices. The Global Risk Institute describes [20] one such
- 730 approach below:

- 731 Phase 1 Identify and document information assets, and their current cryptographic protection.
- Phase 2 Research the state of emerging quantum computers and quantum-safe cryptography.
   Estimate the timelines for availability of these technologies. Influence the development and validation of quantum-safe cryptography.
- **Phase 3** Identify threat actors and estimate their time to access quantum technology "z".
- Phase 4 Identify the lifetime of your asset's "x", and the time required to transform the organization's technical infrastructure to a quantum-safe state "y".
- Phase 5 Determine quantum risk by calculating whether business assets will become vulnerable before the organization can move to protect them. (x + y > z?)
- 740 Mosca's Theorem has been applied in the report Preparing for Post-Quantum Critical Infrastructure by
- the Rand Corporation [21]. The report performed high-level assessments of quantum vulnerabilities in
- the 55 national critical functions (NCFs). The report found six NCFs are high priority for assistance in
- migration, 15 are medium priority, and 34 are low priority.

### 744 3.4.3.2 Crypto Agility Risk Assessment Framework (CARAF)

The Crypto Agility Risk Assessment Framework (CARAF) [22] provides an extension to Mosca's Theorem

- by focusing on developing information systems that encourage support of rapid adaptations of new
- 747 cryptographic primitives and algorithms without making significant changes to the system's
- infrastructure, otherwise known as cryptographic agility [1]. There are five phases to the frameworksummarized below:
- 750 Phase 1 identify threats
- 751 Phase 2 inventory of assets
- 752 Phase 3 risk estimation
- 753 Phase 4 secure assets through risk mitigation
- 754 Phase 5 organizational roadmap

Phases 1-3 are similar to Mosca's algorithm in that the output of the risk analysis is a probability based
on a timeline, given a threat such as the development of a quantum-capable system. Phases 4 and 5 aim
to "facilitate informed decision making to accept, mitigate, or reject the risk from lack of crypto agility as
well as plan to address the risk when appropriate."

## 759 3.4.3.3 Financial Sector (FS) – ISAC Risk Model

- 760 The FS-ISAC Post-Quantum Cryptography Working Group has developed an Infrastructure Inventory
- 761 Technical Paper [23] which suggests that financial organizations develop a Cryptographic Agility Index
- 762 (CAI). The CAI is described as a holistic view that reflects several specific points around prioritization,
- 763 controls, business capabilities, vendors, mitigation, and implementation plans.
- 764 A notable differentiator in the CAI methodology is whether a cryptographic asset is organizationally
- 765 developed or from a third-party vendor. In other words, an organization should incorporate the risk of
- the continued use of third-party software in the overall risk calculation. The CAI also suggests sector-
- 767 specific services, such as the SWIFT banking system, are carefully considered in the migration risk
- 768 calculation process.

#### 769 *3.4.3.4 Telecom Sector*

- 770 In September 2023, the GSM Association published a white paper titled Guidelines for Quantum Risk
- 771 Management for Telco [24] to put forward a methodology that supports telecommunication service
- providers and the extended telecommunication supply chain by using telecom-relevant use cases as an
- example. In this white paper, the authors analyze two existing quantum risk assessment methodologies
- 774 Mosca's "x,y,z" quantum risk model described in Section 3.4.3.1 and CARAF described in Section
- 3.4.3.2. Drawbacks to each approach specific to the telecom sector are identified, and recommendations
- are provided to modify the frameworks to better align with telecom technical and regulatory
- 777 constraints.

## 778 3.4.3.5 Cybersecurity and Infrastructure Security Agency (U.S. DHS/CISA)

- The DHS roadmap for the transition to post-quantum algorithms [25] advises organizations to consider
   the following factors when evaluating systems during the risk assessment process:
- 781 1. Is the system a high-value asset based on organizational requirements?
- What is the system protecting (e.g., key stores, passwords, root keys, signing keys, personally
   identifiable information, sensitive personally identifiable information)?
- 784 3. What other systems does the system communicate with?
- 785 4. To what extent does the system share information with federal entities?
- 786 5. To what extent does the system share information with other entities outside of your organiza-787 tion?
- 788 6. Does the system support a critical infrastructure sector?
- 789 7. How long does the data need to be protected?
- 790 These factors are geared towards U.S. Federal Government system owners; however, they can also be 791 applied to the private sector, especially organizations that support operational critical infrastructure.
- 792 3.4.3.6 Cyber Physical Systems (CPS)
- A report titled Quantum Computing Threat Modelling on a Generic CPS Setup [26] proposes a risk
   methodology for CPSs based on a Process of Attack Simulation and Threat Analysis (PASTA) threat modeling exercise complemented with attack trees and the STRIDE model for identifying threats. The
   report concluded that a threat-based model is a valid approach when assessing risk in the context of
   CPSs.

## 798 **4** Architecture

799 The proposed project architecture is designed to align with the conceptual workflow depicted in Figure 800 2, based on lab discovery technology research and the use case described in Section 3.2.1. In Section 4.1, we present a systems-level view of an architecture that implements vulnerable algorithm discovery 801 802 in the three areas within the scope of the project – the code development pipeline, operational network services and protocols, and operational systems and applications. Finally, we describe our approach to 803 804 normalize the output from the discovery platforms into the common format. In future versions of this document, we will detail the architecture components that will ingest the normalized discovery platform 805 806 output and produce a prioritization list.



807 Figure 2 Conceptual vulnerable cryptography discovery workflow

## 808 4.1 Architecture Description

- 809 This section describes the "to be" architecture for each vulnerable algorithm discovery use case. This is
- subject to change as we continue to experiment with the platforms and receive feedback from the
- 811 larger community of interest. This section may also help inform decision makers in your organization as
- use cases are developed to transition systems to PQC. In future revisions of this document, we will map
- 813 these capabilities to existing cybersecurity best practices and describe a final architecture that will
- 814 identify the usage of vulnerable algorithms within an organization and prioritize their replacement.

## 4.1.1 Protecting the Code Development Pipeline

- 816 Typical CI/CD pipeline components are described in NIST's Implementation of DevSecOps for a
- 817 Microservices-based Application with Service Mesh [27] and are reproduced in Table 5. In this example
- 818 implementation, we focus on pipeline software, SDLC software, and repository components. This
- 819 document describes the CI/CD processes at a high level, and the reader is encouraged to review the
- 820 code development resources below for an in-depth review.

#### 821 Table 5 Software Development Components

Component	Description	
Pipeline Software (CI)	Pulls code from a code repository, invokes the build software, invokes test tools, and stores tested artifacts to image registry	
Pipeline Software (CD)	Pulls out artifacts, packages, and deploys the package based on computing, network, and storage resource descriptions	
SDLC Software	Build tools (e.g., IDEs)	
	Testing tools (e.g., SAST, DAST, SCA)	
Repositories	Source code repositories (e.g., GitHub)	
	Container image repositories or registries	
Observability or Moni-	Logging and log aggregation tools	
toring Tools	Tools that generate metrics	
	Tracing tools (sequence of application calls)	
	Visualization tools (combine data from above to generate dashboard/alerts)	

- Using the tools described in Table 5, we created a code development pipeline that integrates vulnerable
- algorithm discovery at the build stage. In the first scenario, an analyst can extract data from an existing
- 824 codebase to create a queryable database. Once the database is created, the analyst creates and
- 825 executes a query that detects the usage of vulnerable algorithms in the existing codebase. The query is
- 826 executed from a command-line tool or an IDE plugin. The output is generated in SARIF format, described
- 827 in Section 4.1.4.2, where it is consumed by the enterprise visualization platform.
- 828 We also demonstrate two automated versions of this process by integrating the query into an existing
- 829 enterprise CI platform. In the first, an on-premises CI system analyzes the codebase using the command-
- 830 line version of the tool as part of the build process. In the second, we use the built-in automation tools
- from a cloud-based code repository that are executed on each pull request [28]. The full code pipeline
- 832 with vulnerable cryptography discovery activities described above are highlighted below as *static*
- 833 *application security testing* in Figure 3.
- 834 Figure 3 Conceptual CI Pipeline



## 835 4.1.2 Operational Systems and Applications

Filesystem scanning sensors are used to detect quantum-vulnerable algorithms in software. An 836 837 organization may choose to deploy scanners to end-user devices or servers through existing automated 838 enterprise deployment tools. Scans are triggered either manually or through an automated mechanism, 839 and the resulting output is transmitted to te back-end analysis engine or, in the manual case, uploaded 840 to the back-end by the operator. This architecture demonstrates scanning of x64 Linux and Windows 841 hosts, but other platforms may be supported by individual discovery platforms. Further, some discovery scanning solutions may offer integrations with endpoint detection and response (EDR) platforms, which 842 843 combine real-time continuous monitoring and collection of endpoint data with rules-based automated 844 response and analysis capabilities. Such integrations offer the benefit of leveraging already-existing 845 cybersecurity processes and dashboarding capabilities. The scan is also performed on the binaries to 846 discover algorithms that there might not be a source code for, as, for example, in third-party

- 847 applications.
- 848 Figure 4 describes an architecture where the discovery platform sensor output is transmitted to the
- 849 analysis engine.

850 Figure 4 Sensor data flow



## 851 4.1.3 Transport Protocols and Network Services

852 In the scenario described above, we identified two network discovery capabilities – real-time scanning 853 and discovery, and "passive" scanning and discovery from historical packet captures. In Figure 5, we 854 consider traffic capture from the cloud, on-premises, and untrusted networks. The cloud segment 855 contains systems, both bare metal and virtual, that host internal corporate services. Network packets 856 are mirrored to the cloud-based network traffic capture appliance; however, the implementation is 857 dependent on the cloud provider. The network traffic is routed to the PQC vulnerable discovery platform 858 via a secure tunnel to the on-premises network. The on-premises segment contains systems that also 859 host internal services and additionally supports endpoint devices (e.g., laptops) that are operated by end 860 users and managed by the organization. Here, network traffic is mirrored from a physical switch and 861 routed to the network traffic capture appliance. Finally, network traffic from corporately managed endpoint systems that are operated by remote users cannot be directly forwarded to the PQC 862 863 vulnerable discovery platform in real-time. In this scenario, network traffic is captured as a file and 864 forwarded asynchronously to the discovery platform via a secure tunnel (e.g., VPN). Similarly, file-based 865 historical network traffic captures are uploaded directly to the discovery platform for asynchronous

- 866 analysis.
- 867 Figure 5 Passive Network Discovery



## 868 4.1.4 Common Output Elements for Identifying Vulnerable Systems

869 While discovery platforms can provide their own reporting capabilities that enable an administrator to 870 quickly view the results of the discovery process, this demonstration integrates the reports from these 871 platforms into the context of an enterprise-wide dashboard capability. Such an approach consolidates

872 reports from disparate security products that an enterprise may already have fielded, such as EDR

- 873 platforms. Further, consolidation of vulnerable algorithm discovery reports with other cybersecurity
- 874 product reports enables enterprise-level risk management, rather than performing an analysis in
- 875 isolation.
- 876 The reports produced by the discovery platforms in this demonstration are unique in that they do not
- use a common format for representing the discovery results. In a contrived example, a network
- discovery platform may identify a host system as host.example.com:443, whereas another may
- 879 omit the port number (host.example.com). Therefore, we identified the need for a common format
- to represent normalized discovery reports, ideally by leveraging existing normalization efforts to the
- 881 extent possible. The remainder of this section defines a preliminary version of a normalization scheme
- across all discovery platforms that would allow an organization to make a risk decision. It does not
- define a schema, but instead defines descriptive data elements and how they can be obtained from
- passive network observations, active network scans, endpoint monitoring, or configuration information,
- and how those elements can be compared.
- 886 In alignment with our project goal to support automation wherever feasible, the first step in our
- 887 normalization effort was to identify structured machine-readable report formats, such as Extensible
- 888 Markup Language (XML) or JavaScript Object Notation (JSON), that were supported by each discovery
- platform. From there, we identified the application programming interface (API) calls necessary to
- access the resulting report, if the discovery platform offered this capability. APIs are typically serviced
- 891 via synchronous HTTP-based request/response paradigm.

## 892 4.1.4.1 Network Discovery Analysis

893 This section defines the data elements that are used to identify vulnerable algorithms used in protocols

- 894 (as mentioned in Section 4.1.3) identified in Section 3.2.2. First, Table 6 below lists each data element895 and its corresponding representation format.
- 896 Table 6 Network-based discovery data elements

Data Element	Representation
IP (v4 or v6) Address	String
Destination Port	Number
Hostname	String
Application Layer Protocol	String, Service Name or TLS ALPN ID
Application Software	String, Common Platform Enumeration (CPE) 2.3
Operating System	String, Common Platform Enumeration (CPE) 2.3
Device Vendor	String, Common Platform Enumeration (CPE) 2.3

- 897 Next, we describe how each field's value is populated and note other constraints that may affect the
- translation of data presented in the discovery platform to our profile. Generally, *String* and *Number* data
- types are as defined in RFC 8259, The JavaScript Object Notation (JSON) Data Interchange Format [29].
- 900 Service Names are as per RFC 6335 [30], Section 5.1, and are registered by the Internet Assigned
- 901 Numbers Authority (IANA); the registry can be accessed at <u>https://www.iana.org/assignments/service-</u>
- 902 <u>names-port-numbers/service-names-port-numbers.xhtml</u>. TLS Application Layer Protocol Negotiation
- 903 (ALPN) Identifiers (ID) are defined in RFC 7301 [31] and registered at
- 904 <u>https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml#alpn-</u>
- 905 protocol-ids. CPE strings are as defined in the Common Platform Enumeration: Naming Specification,
- 906 Version 2.3, NIST IR 7695 [32].

#### 907 IP Address Data Element Description

- 908 In an IP packet containing a TLS Client Hello, the IP source address is that of the client. In an IP packet
- 909 containing a TLS server response, the IP source address is that of the server. A packet contains a single IP 910 address that corresponds to the client or the server
- address that corresponds to the client or the server.
- 911 In a PKIX certificate, a subjectAltName iPAddress field holds the address of the host associated
- 912 with the certificate. Certificates most often are associated with servers. A certificate can contain
- 913 multiple IP addresses.
- 914 A host observation of an address can contain multiple IP addresses due to multihoming.

#### 915 Destination Port Data Element Description

- In an IP packet containing a TLS Client Hello, the destination port is associated with the service. In an IP
   packet containing a TLS server response, the source port is associated with that service.
- 918 The destination port is essential in identifying a server. TLS is used to protect several protocols, and
- some protocols may appear on more than one port. For instance, the destination port 443 is registered
- 920 for the HTTPS protocol, but HTTPS is also used by other registered protocols, including amt-soap-https,
- 921 appserv-https, commtact-https, 26icros-https, llsurfup-https, oob-ws-https, oracleas-https, pcsync-
- 922 https, pgpkey-https, plysrv-https, sun-sr-https, sun-user-https, tungsten-https, wap-push-https, wbem-
- 923 exp-https, and wbem-https. Additionally, TLS is registered for use by several non-HTTPS protocols.

#### 924 Hostname Data Element Description

- 925 A DNS hostname is a string of labels, each containing up to 63 octets, separated by dots (the '.'
- 926 Character), with a maximum total length of 255 octets [33]. A label is a string that starts and ends with
- 927 an alphanumeric character and contains only alphanumeric characters and hyphens.
- 928 In a TLS Client Hello, the hostname of the server often appears in the Server Name extension, with the
- 929 NameType of HostName. RFC 6066, Section 3 [34] requires this field to contain DNS hostnames and
- 930 not string representations of addresses, though some clients send addresses in this field. The standard
- 931 requires that no more than a single hostname appears in this extension.
- 932 In a TLS Server Certificate, the hostname of the server often appears in the SubjectName or
- 933 SubjectAltName. Multiple hostnames can appear in the certificate.

#### PRELIMINARY DRAFT

#### 934 Application Layer Protocol Data Element Description

- 935 The application layer protocol used in TLS is negotiated between the client and server. In a TLS Client
- 936 Hello, the ALPN field contains a list of ALPN IDs. The server response contains a single ALPN ID. Some
- 937 ALPN IDs are intentionally overloaded to hinder network monitoring; in particular, DNS over HTTPS
- 938 (DoH) uses the ALPN ID of HTTPS. Thus, active scanning obtains more accurate measurements of the
- application layer protocols supported by a server, as compared to passive monitoring.

#### 940 Application Software Data Element Description

- 941 Application software can be identified by a CPE string with a "part" of "a". On a host, identification can
- 942 be performed directly. On a network, it can be performed through banner scraping or behavioral
- 943 fingerprinting.

#### 944 **Operating System Data Element Description**

- An operating system can be identified by a CPE string with a "part" of "o". On a host, identification can
- 946 be performed directly. On a network, it can be performed through banner scraping or behavioral
- 947 fingerprinting.

#### 948 Device Vendor Data Element Description

- A hardware device can be identified by a CPE string with a "part" of "h". On a host, identification can be
- 950 performed directly. On a network, it can be performed through IEEE Organizational Unit Identifier (OUI)
- 951 reporting or behavioral fingerprinting.
- 952 While the CBOM object model is still in its nascent stages, it could potentially provide the necessary
- 953 structure and extensibility to capture the common data elements in a machine-readable format.
- 954 Readers of this document are encouraged to reference the CycloneDX Bill of Materials specification, as it
- provided the basis for the CBOM structure. A CBOM defines a crypto-asset, which is a representation of
- a BOM component type. A crypto-asset is defined by its cryptoProperties object that describes
- 957 predefined assetTypes. In the case of network discovery, we choose the protocol assetType which is
- 958 described by the tlsCipherSuites property. The tlsCipherSuites property defines the TLS cipher suites
- supported by the TLS protocol instantiation reported by the discovery platform. Finally, we leverage the
- 960 properties array, a name-value store defined in the core SBOM specification, to communicate the
- 961 remainder of the common elements. An example follows below of a complete CBOM:

```
962
       {
963
           "bomFormat": "CycloneDX",
964
           "components": [
965
             {
966
               "bom-ref": "oid:1.3.18.0.2.32.104",
967
               "cryptoProperties": {
                 "assetType": "protocol",
968
969
                 "protocolProperties": {
970
                   "tlsCipherSuites": [
971
                     "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519)"
972
                   ]
```

```
973
                  },
 974
                  "scanner": "product name here"
 975
                },
 976
                "name": "tlsv12",
 977
                "properties": [
 978
                  {
 979
                    "name": "internal:ipaddress",
 980
                    "value": "10.10.10.10"
 981
                  },
 982
                  {
                    "name": "internal:Destination Port",
 983
 984
                    "value": "443"
 985
                  },
 986
                  {
 987
                    "name": "internal:Hostname",
 988
                    "value": "some.website.com"
 989
                  },
 990
                  {
 991
                    "name": "internal: Application Layer Protocol",
 992
                    "value": "HTTP/2 over TLS"
 993
                  },
 994
                  {
 995
                    "name": "internal:Application",
 996
                    "value": "cpe:2.3:a:google:chrome:30.0.1599.40:*:*:*:*:*:*:*
 997
                  },
 998
                  {
 999
                    "name": "internal:Operating System",
1000
                    "value": "cpe:2.3:0:28icrosoft:windows_11:22h2:*:*:*:*:*:x64:*"
1001
                  },
1002
                  {
1003
                    "name": "internal:Device Vendor",
                    "value": "D0-43-1E"
1004
1005
                  }
1006
                1,
1007
                "type": "crypto-asset"
1008
              }
1009
            ],
1010
            "specVersion": "1.5",
            "version": 0
1011
1012
          }
```

#### 1013 4.1.4.2 Static Analysis

In Sections 3.2.2.1 and 3.2.2.2, we identified a need to run vulnerable algorithm discovery tools to help
 scan and detect vulnerable code in source code, files, and databases. This demonstration normalizes the
 results in SARIF (Static Analysis Results Interchange Format) [35] as it defines a standard format for the

- 1017 output of static analysis tools and can be used to streamline how static analysis tools share their results.
- 1018 In this section, we specify our profile of the run object, which describes a single run of an analysis tool
- 1019 and contains the output of that run. Within the run object, a tool component describes the specific
- tool (in the context of this project, a discovery engine) which has completed a scan. Table 7 lists the
- 1021 properties of the tool component.
- 1022 Table 7 tool component property specification

Property	Status	Notes
name	mandatory	Consortium member-provided name
version	optional	SARIF includes many tool driver properties that can help to identify the tool version information that was used to create the SARIF log. It is recommended to use at least one of these properties to help iden- tify the tool version, and therefore the rules version, at a later date.
downloadUri	optional	A URI from which this version of the tool can be downloaded.
informationUri	optional	A URI from which information about this version of the tool can be found.
Rules	optional	See " <u>rules property</u> " section below.

1023 An example output of the SARIF *tool* property is presented below.

1024	"tool": {
1025	"driver": {
1026	"name": "Discovery Platform Name",
1027	"organization": "Company Name",
1028	<pre>"product": "Product Name",</pre>
1029	"fullName": "Product Name 1.0.0.0",
1030	"version": "1.0.0.0",
1031	"semanticVersion": "1.0.0",
1032	<pre>"downloadUri": "https://download.uri.contoso.com/v.1.0",</pre>
1033	<pre>"informationUri": "https://information.uri.contoso.com/v.1.0",</pre>
1034	"rules": []

#### 1035 Runs.tool.rules Property Description

1036 The rules property encapsulates an array of zero or more reportingDescriptor objects, which

- 1037 contain the information describing a reporting item generated by the tool, which is either a) the result of
- 1038 the tool's analysis or b) a notification encountered by the tool. Although there are many ways to define
- a SARIF log document, one recommended approach is to enumerate all the rules a tool has executed as
- 1040 a whole, even if there are no findings, then record all unique findings under the results property. This
- 1041 strategy will assist to identify what rules were executed but found nothing to report.

#### 1042 Table 8 runs.tool.rules property specification

Property	Status	Notes
id	mandatory	It is recommended that the id is a readable and stable identifier; it may be an opaque identifier.
Name	optional	A localizable (optionally) string that represents an identifier that is understandable for the end user.
fullDescription	mandatory	A localizable object that describes in detail the reporting item. Use the text sub-property for a description string.
messageStrings	optional	A localizable object that defines the strings that will be generated when the tool execution happens. It is recommended to define the messageStrings/Default with a string so it can be used with the re- sults property. For this project, we use strings with placeholders to unify as much as possible the elements of the quantum-vulnerable cryptography elements found. For details see the " <u>Recommended</u> <u>Message Placeholder Elements</u> " section.
helpUri	optional	An absolute URI of the primary documentation for the reporting item.

1043 An example output of the SARIF *run.tool.rules* property is presented below.

1044	"rules":
1045	{
1046	<pre>"id": "sqlqvc/mssql/qvs001",</pre>
1047	"name": "certificate detection",
1048	"fullDescription": {
1049	"text": "Full description."
1050	},
1051	<pre>"messageStrings": {</pre>
1052	"Default": {
1053	<pre>"text": "Vulnerable Algorithm: [{0}]. Certificate ID `[{1}]` with subject `{2}`, key</pre>
1054	length {3}, expiration date '{4}' that is using quantum-vulnerable cryptography was detected."
1055	}
1056	},
1057	<pre>"helpUri": "https://www.example.com/rules/sqlqvc/mssql/ qvs001"</pre>
1058	}, ]

- 1059 Runs.results Property Description
- 1060 The results property is an array of result objects, which each represent a single result detected by the
- 1061 tool during a run. Following the idea of specifying all rules that were executed under the
- 1062 runs.tool.rules property, the properties in Table 9 are used to represent the actual findings in the
- 1063 tool execution.

## 1064 Table 9 runs.results property specification

Property	Status	Notes
ruleId	mandatory	Although the SARIF specification denotes that this property ex- istence depends on many circumstances, we recommend us- ing it. This property would match a value from the id property from an object defined under runs.tool.rules.
message	optional	Assuming the corresponding object of the runs.tool.rules array has a message, this object will define which one of the mes- sageStrings within the message property for the rule is used, and the values for the placeholders.
Message.id	mandatory	The Id value corresponding to the messageString element that will be reported.
Message.arguments	mandatory	An array of values that correspond to the string placeholders that match the message.id.
locations	mandatory	It should include a value that corresponds to the physical (e.g., file, URI) or logical (e.g., fully qualified function name) location of the element where the issue was found. The actual implementation of the location may change for each tool, but it is typically described as a URI (file://, etc.), and may include a region (e.g., starting row/column, end row/column).

1065 An example output of the SARIF *runs.results* property is presented below.

1066	"results": [
1067	{
1068	<pre>"ruleId": "sqlqvc/mssql/certificates-detection",</pre>
1069	"ruleIndex": 0,
1070	"message": {
1071	"id": "Default",
1072	"arguments": [
1073	"rsa",
1074	"cert_pvk_mk",
1075	"Private Key protected by master Key",
1076	"3072",
1077	"2/17/2024 5:15:37 AM"
1078	]
1079	},
1080	"locations": [
1081	{
1082	"physicalLocation": {
1083	<pre>"artifactLocation": {</pre>
1084	"uri": "sys.certificates",
1085	<pre>"uriBaseId": "REPO_ROOT",</pre>

1086 "index": 0 1087 } 1088 } 1089 }

1090

#### 1091 Recommended Message Property Placeholder Elements

- 1092 This section describes the recommended elements that should be included as part of the
- 1093 runs.tool.rules.message placeholders to describe the elements found by the tool.

1094 The custom elements listed in Table 10 allow upstream tools and aggregating tools to find and present

1095 these elements with ease within the SARIF log. For standardization purposes, we recommend publicly

- 1096 documenting the elements included in your tool and rules version. Table 11 adds recommended values
- to use when communicating the discovered vulnerable algorithm.

#### 1098 Table 10 runs.tools.rules.message Elements

Index	Element Description	Notes
0	Name for the quan- tum-vulnerable algo- rithm detected	The textual representation of the algorithm object identifier being de- tected as at risk. For example: id-ecPublicKey or rsaEncryp- tion (as listed in Table 11)
1	A unique identifier for the particular ele- ment	A unique identifier for the detected element that can be used to de- tect its usage and dependencies elsewhere. For example, in the case of a certificate, it could be a fingerprint of the certificate, a key opaque identifier, or a URI where a key is stored in a vault. If no unique identifier is available, any hint to find the element at risk in a system would be an acceptable alternative.
2+	Other	Starting on index 2, it may be useful to include data that may provide additional information on how the finding may have been used (e.g., what data it may be signing or protecting), or other elements that may assist (X.509 properties, other pieces of metadata, etc.).

#### 1099 Table 11 Vulnerable Algorithm Identifiers

Algorithm Friendly Name	Algorithm Textual Representation	Algorithm Object Identifier	Reference
RSA Keys	rsaEncryption	1.2.840.113549.1.1.1	RFC 3279 [36]
DSA Signature Keys	id-dsa	1.2.840.10040.4.1	RFC 3279
Diffie-Hellman Key Ex- change Keys	dhpublicnumber	1.2.840.10046.2.1	RFC 3279
ECDSA and ECDH Keys	id-ecPublicKey	1.2.840.10045.2.1	RFC 3279
RSASSA-PSS Public Keys	id-RSASSA-PSS	1.2.840.113549.1.1.10	RFC 4055 [37]

Algorithm Friendly Name	Algorithm Textual Representation	Algorithm Object Identifier	Reference
Elliptic Curve Cryptog- raphy Public Key Algo- rithm Identifiers	id-ecPublicKey	1.2.840.10045.2.1	RFC 5480 [38]
Restricted Algorithm Iden- tifiers and Parameters	id-ecDH	1.3.132.1.12	RFC 5480
Restricted Algorithm Iden- tifiers and Parameters	id-ecMQV	1.3.132.1.13	RFC 5480
Curve25519 and Curve448 Algorithm Identifiers	id-Ed25519	1.3.101.112	RFC 8410 [39]
Curve25519 and Curve448 Algorithm Identifiers	id-Ed448	1.3.101.113	RFC 8410
Curve25519 and Curve448 Algorithm Identifiers	id-X25519	1.3.101.110	RFC 8410
Curve25519 and Curve448 Algorithm Identifiers	id-X448	1.3.101.111	RFC 8410

## 1100 **5 Technologies**

## 1101 5.1 Consortium Members

#### 1102 5.1.1 Cisco

1103 Cisco Systems, or Cisco, delivers collaboration, enterprise, and industrial networking and security 1104 solutions. The company's cybersecurity team, Cisco Secure, is one of the largest cloud and network 1105 security providers in the world. Cisco's Talos Intelligence Group, the largest commercial threat 1106 intelligence team in the world, is comprised of world-class threat researchers, analysts, and engineers, 1107 and supported by unrivaled telemetry and sophisticated systems. The group feeds rapid and actionable 1108 threat intelligence to Cisco customers, products, and services to help identify new threats quickly and 1109 defend against them. Cisco solutions are built to work together and integrate into your environment, using the "network as a sensor" and "network as an enforcer" approach to both make your team more 1110 1111 efficient and keep your enterprise secure.

## 1112 5.1.2 IBM

- 1113 IBM researchers, with their academic and industry partners, developed three of the four post-quantum
- 1114 cryptographic algorithms to be standardized by NIST. IBM z16 enterprise server leverages hybrid key
- agreement schemes and dual signing schemes to protect its infrastructure, and relevant to the project it
- 1116 provides a hardware security module and software libraries which allow its clients to experiment with
- 1117 FIPS 203 (CRYSTALS Kyber) and FIPS 204 (CRYSTALS Dilithium), two of the primary post-quantum
- 1118 algorithms slated to be standardized.

- 1119 IBM's technology contribution consists of remote access to a z/OS image of an IBM z16 Mainframe
- 1120 environment, a workstation server, and the software, hardware, and tools needed to demonstrate the
- 1121 crypto discovery capabilities in support of the discovery workstream. This system is hosted at an IBM
- 1122 facility with remote access from the NCCoE Laboratory over a VPN, allowing collaborators to experiment
- 1123 with and understand IBM technologies that discover cryptographic usage within applications (with or
- 1124 without source code) and network connections.

## 1125 5.1.3 Infosec Global (ISG)

ISG is a fast-growing cybersecurity company providing innovative solutions in the field of cryptographic
agility management, cryptographic discovery, and post-quantum cryptography. ISG has a global
footprint with offices in Canada, Switzerland, and the U.S. The ISG team combines the best cryptography
experts (including the 'father' of SSL) with seasoned business leaders experienced in building and
growing new businesses globally.

## 1131 5.1.4 ISARA Corporation

- 1132 ISARA is a security solutions company specializing in cryptographic risk management, quantum-safe
- 1133 cryptography, and cryptographic agility for today's information technology ecosystems. Co-founded in
- 1134 2015 by former BlackBerry security executives, ISARA's cutting-edge technologies are enabling next-
- 1135 generation security for enterprises and governments. With ISARA, you can inventory and manage your
- 1136 cryptographic risks, future-proof your mission-critical systems, and achieve your quantum-safe and zero-
- trust goals. With an emphasis on interoperability, ISARA proudly collaborates on internationalstandards-setting efforts.
- -

## 1139 5.1.5 Keyfactor

- 1140 Keyfactor brings digital trust to the hyper-connected world with identity, encryption, and authentication
- 1141 for every machine, workload, human, and connected thing. By modernizing PKI, discovering and
- automating every digital certificate, and protecting critical software and product supply chains with
- secure digital signing and cryptography, Keyfactor helps organizations establish digital trust then
- 1144 maintain it.
- 1145 Keyfactor is committed to making quantum-ready PKI, signing, and cryptography solutions accessible for
- all, founding and actively supporting widely adopted open-source projects, including EJBCA, SignServer,
- and the FIPS 140-validated Bouncy Castle Cryptography APIs. As a member of the X9 Committee,
- 1148 Keyfactor is an active participant in standards-setting and has incorporated PQC algorithms into the
- 1149 Bouncy Castle APIs and its commercial PKI and signing solutions. With quantum-ready solutions and
- 1150 expertise, Keyfactor is working with customers to stay resilient in the post-quantum world.

## 1151 5.1.6 Microsoft

- 1152 Microsoft is committed to providing secure and trustworthy products and services to its customers. As
- 1153 such, Microsoft has been investing in PQC research, development, experimentation, and collaboration
- since 2014, playing a role in the emergence of PQC and public standards. In particular, Microsoft
- submitted four algorithms in NIST's standardization effort. Microsoft is proud to participate in the Open
- 1156 Quantum Safe project, where they help develop the liboqs library used in this project and by many PQC

industry vendors. Microsoft established the Quantum Safe Program, aiming to accelerate and advanceall quantum-safe efforts across the company from both technical and business perspectives.

## 1159 5.1.7 SafeLogic

1160 SafeLogic is a premier provider of cryptographic solutions that enable enduring privacy and trust in the 1161 ever-changing digital world. Founded in 2012, SafeLogic supplies organizations with strong, FIPS 140 1162 validated cryptography via its FIPS Validation-as-a-Service. SafeLogic's CryptoComply FIPS 140 validated 1163 cryptographic software modules support a broad range of platforms, programming languages, operating 1164 systems, and open-source libraries. SafeLogic expedites delivery of FIPS 140 CMVP certificates for its 1165 CryptoComply customers via RapidCert managed service. It then keeps those certificates active over 1166 time via MaintainCert, the company's white glove managed service that uniquely provides both 1167 software support and certificate maintenance.

- 1168 SafeLogic is keenly interested in building and supporting next-generation cryptographic solutions that
- 1169 will remain secure and compliant with the advent of quantum computers and will be easy for
- 1170 organizations to adopt. As part of that endeavor, SafeLogic is committed to working with NIST and
- 1171 fellow collaborators to capture lessons learned and develop best practices for adopting post-quantum
- 1172 cryptography.

## 1173 5.1.8 Samsung SDS

- 1174 Samsung SDS provides cloud and digital logistics services. Samsung SDS builds optimized cloud
- 1175 environments with Samsung Cloud Platform and provides all-in-one management service, as well as
- 1176 SaaS solutions proven successful in many use cases. One of the core capabilities for delivering their
- 1177 service is cybersecurity, and cryptographic technology plays a fundamental role to enhance security. To
- 1178 this end, Samsung SDS is engaged in various cryptographic research and development activities,
- 1179 including the design, implementation, and architecting of cryptographic techniques, including post-
- 1180 quantum cryptography.

## 1181 5.1.9 SandboxAQ

1182 SandboxAQ recently launched its Security Suite, a product for modern cryptography management which 1183 enables customers to pilot this change process towards a world where cryptography is observable,

- 1184 controllable, compliant, and secure. The SandboxAQ Security Suite is an end-to-end, crypto-agility
- 1185 platform that provides a full inventory of existing cryptography use, including vulnerability and
- 1186 compliance analysis, as well as a path to centrally managed, robust and agile cryptography. The result is
- 1187 protection from today's attacks, as well as security against the future threat of a large-scale quantum
- 1188 computer.

## 1189 5.1.10 wolfSSL

- 1190 wolfSSL focuses on providing lightweight and embedded security solutions with an emphasis on speed,
- size, portability, features, and standards compliance. With its SSL/TLS products and crypto library,
- 1192 wolfSSL is supporting high-security designs in automotive, avionics, and other industries. In avionics,
- 1193 wolfSSL supports Radio Technical Commission for Aeronautics Software Considerations in Airborne
- 1194 Systems and Equipment Certification. In automotive, wolfSSL supports MISRA-C capabilities. For

- government consumers, wolfSSL has a valid <u>FIPS 140-2</u> certificate. wolfSSL supports industry standards
   up to the current TLS 1.3 and DTLS 1.3, offers a simple API and an OpenSSL compatibility layer, is backed
- 1197 by the wolfCrypt cryptography library, and provides 24x7 support and much more. wolfSSL's products
- 1197 by the woncrypt cryptography library, and provides 24x7 support and much more, wonsse's products
- are open source, giving customers the ability to examine them.

## 1199 **5.2 Products and Technologies**

- 1200 The organizations listed in Section 5.1 have contributed the products and technologies listed in Table 12.
- 1201 For each product or technology, the table specifies the type of component, its name, and the function
- 1202 the technology will serve in the demonstration.
- 1203 Table 12 Products and Technologies

Component Type	Product or Tech- nology Name	Function
Vulnerable Al- gorithm Dis- covery Plat- form System	IBM Z16 Main- frame	Computing platform supporting tools, applications, and crypto hardware.
Vulnerable Al- gorithm Dis- covery Plat- form System	IBM Workstation Server	Computing platform supporting tools.
Quantum- Ready Crypto Implementa- tion	IBM Crypto Ex- press 8S	HSM providing support for cryptographic algorithm services.
Vulnerable Al- gorithm Dis- covery Plat- form	IBM z/OS Inte- grated Crypto- graphic Service Facility (ICSF)	Captures crypto-related information by writing System Manage- ment Facility (SMF) activity log records aggregating usage of cryptographic engines, services, and algorithms. The information is captured dynamically while the application is running. This fa- cility can be used both to provide an inventory and to audit and observe during the complete migration process.
Vulnerable Al- gorithm Dis- covery Plat- form	IBM Application Discovery and Delivery Intelli- gence (ADDI) v.6.1	Serves as a static analysis tool that analyzes COBOL application source files capturing all ICSF crypto services, the parameters as- sociated with the services, and valuable metadata. Produces a crypto discovery report.
Vulnerable Al- gorithm Dis- covery Plat- form	IBM Crypto Ana- lytics Tool (CAT)	Provides snapshots of the z/OS environment by extracting secu- rity and cryptographic information based on configurable poli- cies. Provides details on keys managed by ICSF and RACF. Helps identify insecure keys, algorithms, and enabled services.

Component Type	Product or Tech- nology Name	Function
Vulnerable Al- gorithm Dis- covery Plat- form	IBM z/OS Encryp- tion Readiness Technology (zERT)	Collects and reports the cryptographic security attributes of Ipv4 and Ipv6 connections that are protected using the TLS/SSL, SSH, and Ipsec cryptographic network security protocols. This tool helps provide the context for linking keys, certificates, and the applications using them. It identifies security protocols, crypto al- gorithms, key lengths, etc., all important information to have during the crypto discovery process.
Vulnerable Al- gorithm Dis- covery Plat- form	(SandboxAQ) Se- curity Suite – Dis- covery Modules	Provides cryptographic observability capabilities. Analyzes IT in- frastructure and creates a cryptographic inventory that allows stakeholders to monitor who/what, where, when, and how cryp- tography is used across an organization, including PQC.
Vulnerable Al- gorithm Dis- covery Plat- form	Samsung SDS Crypto Agility Platform for En- terprise (S-CAPE)	A platform that enables the discovery of PQ-vulnerable algo- rithms across the enterprise's DevSecOps pipeline. By consolidat- ing the data from various sensors, it provides visibility and esti- mated risk scores for the identified PQ vulnerabilities.
Cryptographic Inventory	InfoSec AgileSec Analytics Enter- prise Server	AgileSec Analytics is an enterprise-grade security solution de- signed to enable companies in building a comprehensive and centralized inventory of all cryptographic assets, including cryp- tographic keys, keystores, X.509 certificates, cryptographic librar- ies, cryptographic algorithms, and cryptographic protocols de- ployed across their digital footprint.
Cryptographic Risk Assess- ment	InfoSec AgileSec Analytics Dash- board	AgileSec Analytics Dashboard is a core component of AgileSec Analytics enabling companies to review the cryptographic inven- tory and proactively identify cryptographic weaknesses, compli- ance gaps, or quantum-vulnerable objects based on a customiza- ble cryptographic policy.
Cryptographic Data Collec- tion	InfoSec AgileSec Sensors	AgileSec Analytics Sensors are core components of AgileSec Ana- lytics and are used to scan different technologies and systems deployed within a digital footprint. The sensors can be used to scan hosts (filesystem, binary data, running processes, certificate store), network interfaces, CI/CD pipelines, application reposito- ries, key management systems, PKI systems, HSM systems, and other technologies.
Cryptographic Vulnerability Remediation	AgileSec Analyt- ics Vulnerability Response Con- nector	AgileSec Analytics Vulnerability Response Connector enables companies to seamlessly process the remediation of crypto- graphic vulnerabilities detected across their digital footprint with their existing solutions (e.g., ServiceNow).

Component Type	Product or Tech- nology Name	Function
Cryptographic Agility	AgileSec Agility SDK	AgileSec Agility SDK enables companies building applications to natively integrate cryptographic agility within their sensitive busi- ness applications. With AgileSec Agility SDK, the cryptographic operations are abstracted from the developers and managed through policy, enabling seamless migration from classical to post-quantum, national, or any other future cryptographic stand- ards.
Vulnerable Al- gorithm Dis- covery Plat- form	ISARA Advance <sup>®</sup> Cryptographic Discovery and Risk Assessment Tool	A platform for analyzing and inventorying the use of cryptog- raphy on enterprise networks. Includes features for inventorying devices and servers, and analyzing cryptographic risk to help you prioritize assets for cryptographic migrations. Results can be viewed in the platform's easy-to-use dashboard or exported to other systems.
Vulnerable Al- gorithm Dis- covery Plat- form	(Samsung SDS) SECUI BLUEMAX NGF VE	A virtualized firewall configured to detect the presence of post- quantum vulnerable traffic.
Vulnerable Al- gorithm Dis- covery Plat- form	Microsoft CodeQL	Extension of Visual Studio Code and GitHub action that detects post-quantum vulnerable code.
Vulnerable Al- gorithm Dis- covery Plat- form	Cisco Mercury	Post-quantum vulnerable network packet metadata capture and analysis.
TLS Crypto- graphic Li- brary Protocol Implementa- tion	wolfSSL	A software library that implements TLS and DTLS 1.3, supporting classical and quantum-safe symmetric and asymmetric ciphers to be standardized by NIST.
Cryptographic Library Imple- mentation	Keyfactor (Bouncy Castle)	(In partnership with the Legion of the Bouncy Castle Inc.) The Bouncy Castle libraries now include support for both classical and quantum-safe algorithms (NIST upcoming standards included), together with support for protocols such as TLS/DTLS, CMS, Time-Stamp Protocol, OpenPGP, and a variety of protocols around X.509 certificate generation and management.
Certificate Discovery and Management	Keyfactor Com- mand	Command is a certificate discovery and lifecycle automation so- lution that provides centralized visibility, governance, and lifecy- cle automation for digital certificates at scale, helping organiza- tions to identify and replace weak and non-compliant certifi- cates.

Component Type	Product or Tech- nology Name	Function
Quantum- Ready PKI Platform	Keyfactor EJBCA	EJBCA is a modern, crypto-agile PKI platform that can be easily deployed in the cloud, on-premises, or in a hybrid mode. EJBCA supports issuance of certificates using both classical and quan- tum-safe algorithms.
Quantum- Ready Digital Signing	Keyfactor Sign- Server	SignServer is a flexible, crypto-agile signing solution that enables application, manufacturing, and product teams to digitally sign code and artifacts with quantum-safe algorithms.

## 1204 6 Future Project Considerations

1205 This volume presents a desired end state of an architecture which demonstrates tools that will discover 1206 vulnerable algorithms in an existing enterprise operational environment. We hope that publishing this 1207 document prior to integration activities will allow for refinements that will be submitted during the 1208 public comment period. As of the writing of this document, the project has installed, configured, and 1209 deployed technologies and tools from multiple consortium members that address discovery in our lab 1210 environment. These technologies are at the leading edge of the migration process, and as such, the 1211 project has taken a deliberate approach in learning the capabilities each platform provides by 1212 developing the demonstration plan described in Appendix C and 0. The demonstration plan serves as a 1213 rough guide to develop the practical and readily adoptable use cases described within this guide. In 1214 future iterations we will finalize a data normalization scheme and add components that will enable 1215 automation and prioritization using a risk-managed approach.

- 1216 Cryptographic discovery tools may produce inventories that indicate an organization is using more
- 1217 quantum-vulnerable cryptography than was expected, highlighting to the organization that they need to
- use a risk-based approach to mitigating the risks of the use of quantum-vulnerable cryptography. Future
- demonstrations within this project may address one or more of the risk methodologies identified in
- 1220 Section 3.4.3 to support the risk-based decisions an organization needs to make to prioritize its actions
- 1221 to migrate to post-quantum cryptography.
- 1222 In the next revision of the draft document, we may broaden the scope of the project to include other
- discovery cases. One area that we are exploring is data-at-rest protection. When thinking of data-at-rest,
- 1224 the focus is primarily on the encryption algorithm, but attention must also be paid to the integrity of
- data-at-rest. An example is ensuring the integrity of audit records. In this case, hashes and/or digital
- 1226 signatures are often used to ensure that records have not been tampered with. Digital signatures will be
- susceptible to attack and broken by Shor's algorithms. Example scenarios of data-at-rest protection
- 1228 could include database protection schemes, Secure Multipurpose Internet Mail Extensions (S/MIME),
- 1229 code signing, and record integrity.
- 1230 Finally, the project has also submitted the CWE described in Section 3.4.2 to the Federally Funded
- 1231 Research and Development Center (The MITRE Corporation) that maintains the CWE program for
- 1232 review. In the next version of this document, we hope that the submission spurs conversation in this
- space, and that a consensus is achieved on one or more CWEs that will benefit organizations and
- industry alike.

PRELIMINARY DRAFT

# 1235 Appendix A List of Acronyms

ADDI	IBM Application Discovery and Delivery Intelligence
AitM	Adversary-in-the-Middle
ALPN	TLS Application Layer Protocol Negotiation
ΑΡΙ	Application Programming Interface
ARP	Address Resolution Protocol
CAI	Cryptographic Agility Index
CARAF	Crypto Agility Risk Assessment Framework
САТ	IBM Crypto Analytics Tool
СВОМ	Cryptography Bill of Materials
CFDIR	Canadian Forum for Digital Infrastructure Resilience
CI/CD	Continuous Integration/Continuous Delivery
CISA	Cybersecurity & Infrastructure Security Agency
CISO	Chief Information Security Officer
СРЕ	Common Platform Enumeration
CPS	Cyber Physical System
CRADA	Cooperative Research and Development Agreement
CRQC	Cryptanalytically Relevant Quantum Computer
CSF	Cybersecurity Framework
CWE	Common Weakness Enumeration
DAST	Dynamic Application Security Testing
DH	Diffie Hellman
DHS	U.S. Department of Homeland Security
DNS	Domain Name System
DoH	DNS over HTTPS
DSS	Digital Signature Standard
DTLS	Datagram Transport Layer Security
ECDH	Elliptic Curve Diffie Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm

EdDSA	Edwards-curve Digital Signature Algorithm
EDR	Endpoint Detection and Response
ETL	Extract, Transform, and Load
FIPS	Federal Information Processing Standard
FS-ISAC	Financial Services Information Sharing and Analysis Center
HSM	Hardware Security Module
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICSF	IBM z/OS Integrated Cryptographic Service Facility
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IKEv2	Internet Key Exchange Version 2
IMAPS	Internet Message Access Protocol Secure
IPsec	Internet Protocol Security
JSON	JavaScript Object Notation
LLMNR	Link-Local Multicast Name Resolution
ML-DSA	Module Learning with Errors Digital Signature Algorithm
MLWE-KEM	Module Learning with errors Key Encapsulation Mechanism
MQV	Menezes Qu Vanstone
NCCoE	National Cybersecurity Center of Excellence
NCF	National Critical Function
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NSM	National Security Memorandum
ОМВ	Office of Management and Budget
PASTA	Process of Attack Simulation and Threat Analysis
РКСЅ	Public-Key Cryptography Standard
РКІ	Public Key Infrastructure
РКІХ	Public Key Infrastructure (X.509)

POP3S	Post Office Protocol 3 Secure
PQ	Post-Quantum
PQC	Post-Quantum Cryptography
QRWG	Quantum-Readiness Working Group
RACF	(IBM) Resource Access Control Facility
RDP	(Microsoft) Remote Desktop Protocol
RFC	Request for Comment
S-CAPE	Samsung SDS Crypto Agility Platform for Enterprise
SARIF	Static Analysis Results Interchange Format
SAST	Static Application Security Testing
SBOM	Software Bill of Materials
SCA	Software Composition Analysis
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SIEM	Security Information and Event Management
SLH-DSA	Stateless Hash-based Digital Signature Algorithm
SMF	System Management Facility
SMTP	Simple Mail Transfer Protocol
SP	Special Publication
SSDF	Secure Software Development Framework
SSH	Secure Shell
TLS	Transport Layer Security
VPN	Virtual Private Network
XML	Extensible Markup Language
zERT	IBM z/OS Encryption Readiness Technology

## 1236 Appendix B References

- Barker WC, Polk WT, Souppaya MP (2021) Getting Ready for Post-Quantum Cryptography:
   Exploring Challenges Associated with Adopting and Using Post-Quantum Cryptographic
   Algorithms. (National Institute of Standards and Technology, Gaithersburg, MD), NIST
   Cybersecurity White Paper (CSWP) NIST CSWP 15. https://doi.org/10.6028/NIST.CSWP.15
- 1241[2]Barker W, Souppaya M, Newhouse W (2021) Migration to Post-Quantum Cryptography Project1242Description. (National Institute of Standards and Technology, Gaithersburg, MD). Available at1243https://csrc.nist.gov/pubs/pd/2021/08/04/migration-to-postquantum-cryptography/final
- 1244[3]National Security Memorandum 10 (NSM-10) (2022) National Security Memorandum on1245Promoting United States Leadership in Quantum Computing While Mitigating Risks to1246Vulnerable Cryptographic Systems. (The White House, Washington, DC). Available at1247https://www.whitehouse.gov/briefing-room/statements-releases/2022/05/04/national-1248security-memorandum-on-promoting-united-states-leadership-in-quantum-computing-while-1249mitigating-risks-to-vulnerable-cryptographic-systems
- 1250[4]Cybersecurity & Infrastructure Security Agency, National Security Agency, and National Institute1251of Standards and Technology (2023) Quantum-Readiness: Migration to Post-Quantum1252Cryptography. (CISA, Arlington, Virginia), August 21, 2023. Available at1253https://www.cisa.gov/resources-tools/resources/quantum-readiness-migration-post-quantum-1254cryptography
- 1255[5]Office of Management and Budget (2022) Migrating to Post-Quantum Cryptography. (The White1256House, Washington, DC), OMB Memorandum M-23-02, November 18, 2022. Available at1257https://www.whitehouse.gov/wp-content/uploads/2022/11/M-23-02-M-Memo-on-Migrating-1258to-Post-Quantum-Cryptography.pdf
- Souppaya M, Ogata M, Watrobski P, Scarfone K (2022) Software Supply Chain and DevOps
   Security Practices: Implementing a Risk-Based Approach to DevSecOps Project Description.
   (National Institute of Standards and Technology, Gaithersburg, MD). Available at
   <u>https://www.nccoe.nist.gov/sites/default/files/2022-11/dev-sec-ops-project-description-</u>
   <u>final.pdf</u>
- Souppaya MP, Scarfone KA, Dodson DF (2022) Secure Software Development Framework (SSDF)
   Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. (National
   Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-218.
   <u>https://doi.org/10.6028/NIST.SP.800-218</u>
- 1268[8]Quantum-Readiness Working Group of the Canadian Forum for Digital Infrastructure Resilience1269(2023) Canadian National Quantum-Readiness Best Practices and Guidelines, Version 03.1270Available at <a href="https://ised-isde.canada.ca/site/spectrum-management-">https://ised-isde.canada.ca/site/spectrum-management-</a>1271telecommunications/sites/default/files/attachments/2023/cfdir-quantum-readiness-best-
- 1272 practices-v03.pdf

- 1273[9]Alagic G, Apon D, Cooper DA, Dang QH, Dang T, Kelsey JM, Lichtinger J, Liu Y-K, Miller CA, Moody1274D, Peralta R, Perlner RA, Robinson A, Smith-Tone D (2022) Status Report on the Third Round of1275the NIST Post-Quantum Cryptography Standardization Process. (National Institute of Standards1276and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) NIST IR 8413-upd1,1277Includes updates as of September 26, 2022. <a href="https://doi.org/10.6028/NIST.IR.8413-upd1">https://doi.org/10.6028/NIST.IR.8413-upd1</a>
- 1278 [10] Florence D (2023) Terminology for Post-Quantum Traditional Hybrid Schemes. (Internet
   1279 Engineering Task Force (IETF)), Internet-Draft draft-ietf-pquip-pqt-hybrid-terminology. Available
   1280 at <u>https://datatracker.ietf.org/doc/draft-ietf-pquip-pqt-hybrid-terminology/00/</u>
- [11] National Institute of Standards and Technology (2023) Module-Lattice-based Key-Encapsulation
   Mechanism Standard. (U.S. Department of Commerce, Washington, DC), Federal Information
   Processing Standards Publication (FIPS) 203 (Draft). <u>https://doi.org/10.6028/NIST.FIPS.203.ipd</u>
- 1284[12]National Institute of Standards and Technology (2023) Module-Lattice-Based Digital Signature1285Standard. (U.S. Department of Commerce, Washington, DC), Federal Information Processing1286Standards Publication (FIPS) 204 (Draft). <a href="https://doi.org/10.6028/NIST.FIPS.204.ipd">https://doi.org/10.6028/NIST.FIPS.204.ipd</a>
- 1287 [13] National Institute of Standards and Technology (2023) Stateless Hash-Based Digital Signature
   1288 Standard. (U.S. Department of Commerce, Washington, DC), Federal Information Processing
   1289 Standards Publication (FIPS) 205 (Draft). <u>https://doi.org/10.6028/NIST.FIPS.205.ipd</u>
- [14] National Institute of Standards and Technology (2023) Digital Signature Standard (DSS). (U.S.
   Department of Commerce, Washington, DC), Federal Information Processing Standards
   Publication (FIPS) 186-5. https://doi.org/10.6028/NIST.FIPS.186-5
- [15] Barker EB, Chen L, Roginsky AL, Vassilev A, Davis R (2018) Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-56A, Rev. 3.
   https://doi.org/10.6028/NIST.SP.800-56Ar3
- 1297 [16] MITRE (2023) ATT&CK. Available at https://attack.mitre.org/tactics/TA0009/
- [17] Barker EB, Chen L, Roginsky AL, Vassilev A, Davis R, Simon S (2019) Recommendation for Pair Wise Key-Establishment Using Integer Factorization Cryptography. (National Institute of
   Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-56B, Rev. 2.
   https://doi.org/10.6028/NIST.SP.800-56Br2
- 1302[18]Accredited Standards Committee X9 (2022) Quantum Computing Risks to the Financial Services1303Industry. X9 Informative Report X9 IR F01-2022. Available at <a href="https://x9.org/download-qc-ir/">https://x9.org/download-qc-ir/</a>
- 1304 [19] MITRE (2023) Common Weakness Enumeration (CWE). Available at <u>https://cwe.mitre.org/</u>
- [20] Mosca M, Mulholland J. A Methodology for Quantum Risk Assessment. (Global Risk Institute.)
   Available at <u>https://globalriskinstitute.org/mp-files/a-methodology-for-quantum-risk-</u>
   assessment-pdf.pdf/

1308 1309 1310 1311	[21]	Vermeer MJD, Parker E, Kochhar A (2022) Preparing for Post-Quantum Critical Infrastructure: Assessments of Quantum Computing Vulnerabilities of National Critical Functions. (Homeland Security Operational Analysis Center operated by the Rand Corporation.) Available at <u>https://www.rand.org/pubs/research_reports/RRA1367-6.html</u>
1312 1313 1314	[22]	Ma C, Colon L, Dera J, Rashidi B, Garg V (2021) CARAF: Crypto Agility Risk Assessment Framework. Journal of Cybersecurity, Volume 7, Issue 1, pp. 1–11. <u>https://doi.org/10.1093/cybsec/tyab013</u>
1315 1316 1317	[23]	Post-Quantum Cryptography Working Group (2023) Risk Model Technical Paper. (FS-ISAC Post- Quantum Cryptography Working Group.) Available at <u>https://www.fsisac.com/hubfs/Knowledge/PQC/RiskModel.pdf?hsLang=en</u>
1318 1319 1320	[24]	GSM Association (2023) Guidelines for Quantum Risk Management for Telco, Version 1.0. (GSM Association.) Available at <u>https://www.gsma.com/aboutus/workinggroups/wp-</u> content/uploads/2023/09/Guidelines-for-Quantum-Risk-Management-for-Telco-v1.0.pdf
1321 1322 1323 1324 1325	[25]	Cybersecurity & Infrastructure Security Agency, National Security Agency, and National Institute of Standards and Technology (2023) Quantum-Readiness: Migration to Post-Quantum Cryptography. (CISA, Arlington, Virginia), August 21, 2023. Available at <u>https://www.cisa.gov/resources-tools/resources/quantum-readiness-migration-post-quantum- cryptography</u>
1326 1327	[26]	Lee CC, Tan TG, Sharma V, Zhou J (2021) Quantum Computing Threat Modelling on a Generic CPS Setup. Available at <u>https://link.springer.com/chapter/10.1007/978-3-030-81645-2_11</u>
1328 1329 1330	[27]	Chandramouli R (2022) Implementation of DevSecOps for a Microservices-based Application with Service Mesh. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-204C. <u>https://doi.org/10.6028/NIST.SP.800-204C</u>
1331 1332 1333 1334	[28]	Github (2023) Configuring default setup for code scanning. Available at <a href="https://docs.github.com/en/code-security/code-scanning/automatically-scanning-your-code-for-vulnerabilities-and-errors/configuring-code-scanning-for-a-repository#configuring-code-scanning-automatically">https://docs.github.com/en/code-security/code-scanning/automatically-scanning-your-code-for-vulnerabilities-and-errors/configuring-code-scanning-for-a-repository#configuring-code-scanning-automatically</a>
1335 1336 1337	[29]	Bray T (2017) The JavaScript Object Notation (JSON) Data Interchange Format. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8259. https://doi.org/10.17487/RFC8259
1338 1339 1340 1341	[30]	Cotton M, Eggert L, Touch J, Westerlund M, Cheshire S (2011) Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 6335. <u>https://doi.org/10.17487/RFC6335</u>
1342 1343 1344	[31]	Friedl S, Popov A, Langley A, Stephan E (2014) Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 7301. <u>https://doi.org/10.17487/RFC7301</u>

- [32] Cheikes BA, Waltermire DA, Scarfone KA (2011) Common Platform Enumeration: Naming
   Specification Version 2.3. (National Institute of Standards and Technology, Gaithersburg, MD),
   NIST Interagency or Internal Report (IR) 7695. <u>https://doi.org/10.6028/NIST.IR.7695</u>
- 1348[33]Braden R (1989) Requirements for Internet Hosts Application and Support. (Internet1349Engineering Task Force (IETF)), IETF Request for Comments (RFC) 1123.1350<a href="https://doi.org/10.17487/RFC1123">https://doi.org/10.17487/RFC1123</a>
- [34] Eastlake D (2011) Transport Layer Security (TLS) Extensions: Extension Definitions. (Internet
   Engineering Task Force (IETF)), IETF Request for Comments (RFC) 6066.
   https://doi.org/10.17487/RFC6066
- 1354[35]OASIS Static Analysis Results Interchange Format Technical Committee (2023) Static Analysis1355Results Interchange Format (SARIF) Version 2.1.0 Plus Errata 01. Available at <a href="https://docs.oasis-open.org/sarif/v2.1.0/sarif-v2.1.0.pdf">https://docs.oasis-open.org/sarif/v2.1.0/sarif-v2.1.0.pdf</a>
- 1357[36]Bassham L, Polk W, Housley R (2002) Algorithms and Identifiers for the Internet X.509 Public Key1358Infrastructure Certificate and Certificate Revocation List (CRL) Profile. (Internet Engineering Task1359Force (IETF)), IETF Request for Comments (RFC) 3279. <a href="https://doi.org/10.17487/RFC3279">https://doi.org/10.17487/RFC3279</a>
- [37] Schaad J, Kaliski B, Housley R (2005) Additional Algorithms and Identifiers for RSA Cryptography
   for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List
   (CRL) Profile. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 4055.
   https://doi.org/10.17487/RFC4055
- 1364[38]Turner S, Brown D, Yiu K, Housley R, Polk T (2009) Elliptic Curve Cryptography Subject Public Key1365Information. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 5480.1366<a href="https://doi.org/10.17487/RFC5480">https://doi.org/10.17487/RFC5480</a>
- 1367[39]Josefsson S, Schaad J (2018) Algorithm Identifiers for Ed25519, Ed448, X25519, and X448 for Use1368in the Internet X.509 Public Key Infrastructure. (Internet Engineering Task Force (IETF)), IETF1369Request for Comments (RFC) 8410. <a href="https://doi.org/10.17487/RFC8410">https://doi.org/10.17487/RFC8410</a>

# Appendix C Discovery Platform Lab Functional Demonstration Plan

## 1372 C.1 Use Case 1

- 1373 Scenario
- 1374 Validate the discovery of quantum-vulnerable algorithms in TLS protocol version 1.2.

#### 1375 Steps

- 13761. Configure discovery tool to scan layer 4 (transport) traffic. This process will depend on the indi-1377vidual discovery tool deployment methodology.
- 1378 2. Modify test data packet capture IP addresses as needed for target network.
- Replay TLS 1.2 test data onto network segment with deployed discovery tool using a utility such
   as tcpreplay.

#### 1381 Test Data

1382 <u>Encrypted Web traffic dataset</u> from network monitoring and host-based monitoring across a large
 1383 campus network

#### 1384 Expected Results

- 1385 Discovery tool detects the presence of vulnerable key exchange/agreement and authentication
- algorithms, with results captured in an output file and/or dashboard.

## 1387 C.2 Use Case 2

- 1388 Scenario
- 1389 Validate the discovery of quantum vulnerable algorithms in SSH protocol version 2.

#### 1390 Steps

- 13911. Configure discovery tool to scan layer 4 (transport) traffic. This process will depend on the indi-1392vidual discovery tool deployment methodology.
- 1393 2. Modify test data packet capture IP addresses as needed for target network.
- 13943. Replay SSH 2.0 test data onto network segment with deployed discovery tool using a utility such1395as tcpreplay.
- 1396 Test Data
- 1397 <u>AZSecure Data</u> provided by AZSecure Data and the University of Arizona Artificial Intelligence Lab.

#### 1398 Expected Results

- 1399 Discovery tool detects the presence of vulnerable key exchange algorithms. Refer to RFC 9142, Key
- 1400 Exchange (KEX) Method Updates and Recommendations for Secure Shell (SSH) for existing key exchange 1401 method names with results captured in an output file and/or dashboard.

### 1402 C.3 Use Case 3

- 1403 Scenario
- 1404 Validate the discovery of quantum-vulnerable algorithms in non-executable files in Windows-based1405 operational systems.
- 1406 Steps
- 1407 1. Create a fully updated Windows 10/11 virtual machine.
- 1408 2. Install the discovery tool sensor component if applicable.
- 14093. Confirm proper communication between the sensor component and back-end systems provided1410 by the discovery tool.
- 1411 4. Move dataset files to the local disk of the target virtual machine.
- 1412 5. Trigger the discovery sensor to perform a scan

#### 1413 Test Data

- 1414 Using algorithms listed in Table 3, the following datasets are project-specific artifacts generated by the1415 project team:
- 1416 PKCS#12 keystores created via OpenSSL
- Java keystores created via keytool utility
- 1418 PKCS#1 keystores created via OpenSSL
- 1419 PKCS#8 keystores created via OpenSSL
- OpenSSH keystores created via ssh-keygen
- OpenPGP keystores created via gpg

#### 1422 Expected Results

1423 Discovery tool sensor detects files from dataset, with results captured in an output file and/or 1424 dashboard.

#### 1425 C.4 Use Case 4

- 1426 Scenario
- 1427 Validate the discovery of quantum-vulnerable algorithms in non-executable files in a Linux-based
- 1428 operational system.

#### PRELIMINARY DRAFT

#### 1429 Steps

- 1430 1. Create a fully updated Ubuntu 22.04 virtual machine.
- 1431 2. Install the discovery tool sensor component if applicable.
- Confirm proper communication between the sensor component and back-end systems provided
   by the discovery tool.
- 1434 4. Move dataset files to the local disk of the target virtual machine.
- 1435 5. Trigger the discovery sensor to perform a scan.

#### 1436 Test Data

1437 See Use Case 3.

#### 1438 Expected Results

1439 Discovery tool sensor detects files from dataset, with results captured in an output file and/or1440 dashboard.

## 1441 C.5 Use Case 5

#### 1442 Scenario

- 1443 Validate the discovery of quantum-vulnerable algorithms in executable files on Windows-based
- 1444 operational systems that are representative of a typical enterprise deployment.
- 1445 Steps
- 1446 1. Install, configure, and deploy discovery tool endpoint agent.
- 1447 2. Note discovered artifacts, given discovery platform capabilities.

#### 1448 Test Data

- 1449 Windows 11-based client with the following software packages:
- 1450 Google Chrome
- 1451 Java Runtime Environment
- 1452 <u>VLC Media Player</u>
- 1453OpenVPN Connect
- 1454 <u>GRR Rapid Response</u>
- 1455 <u>Slack</u>
- 1456 Expected Results
- 1457 The discovered artifacts are collected and optionally transmitted to a back-end system for review.

## 1458 **C.6 Use Case 6**

- 1459 Scenario
- 1460 Validate the discovery of quantum-vulnerable algorithms in executable files on Linux-based operational1461 systems that are representative of a typical enterprise deployment.
- 1462 Steps
- 1463 1. Install, configure, and deploy discovery tool endpoint agent.
- 1464 2. Note discovered artifacts, given discovery platform capabilities.
- 1465 Test Data
- 1466 Ubuntu 22.04 Server Edition image with all base services enabled.
- 1467 Expected Results
- 1468 The discovered artifacts are collected and optionally transmitted to a back-end system for review.

### 1469 C.7 Use Case 7

- 1470 Scenario
- 1471 Validate the discovery of quantum-vulnerable algorithms in code that leverages cryptography in a CI/CD
- 1472 pipeline (IDE plugin).

#### 1473 Steps

- 1474 1. Create a development project in IDE using source code from dataset.
- 1475 2. Trigger the discovery tool within the IDE.
- 1476 Test Data

1478

1481

- Source Code
  - RSA Encryption/Description API usage
- 1479 o ECDSA Sign/Verify API usage
- 1480 Library/Process
  - KeyPairGenerator, KeyFactory, Signature, SignatureException Java class
- 1482Expected Results
- Discovery tool identifies files and lines of code that use the methods described in the test dataset anddisplays results within the IDE.

### 1485 C.8 Use Case 8

1486 Scenario

1487 Validate the discovery of quantum-vulnerable algorithms in code that leverages cryptography in a CI/CD1488 pipeline (Repository).

#### PRELIMINARY DRAFT

1489	Steps		
1490	1.	Clone repository containing source code from dataset.	
1491	2.	Make a change in the codebase (e.g., add a print statement such as "hello world").	
1492	3.	Commit the change.	
1493	4.	Create a pull request in the configured remote repository.	
1494	5.	Code scan is triggered automatically.	
1495 Test Data			
1496	•	Source Code	
1497		<ul> <li>RSA Encryption/Description API usage</li> </ul>	
1498		<ul> <li>ECDSA Sign/Verify API usage</li> </ul>	
1499	•	Library/Process	
1500		<ul> <li>KeyPairGenerator, KeyFactory, Signature, SignatureException Java class</li> </ul>	
1501	1 Expected Results		
1502	Discovery tool identifies files and lines of code that use the methods described in the test dataset and		

1503 displays results within the repository console.

# Appendix D IBM Z16 Remote Discovery Platform Functional Demonstration Plan

## 1506 **D.1 Use Case 1**

- 1507 Scenario
- 1508 Validate the discovery of post-quantum vulnerable algorithms in executable modules in the z/OS-based1509 operational system.

#### 1510 Steps

- 1511 1. Enable the ICSF crypto usage tracking feature of z/OS.
- 1512 2. Run the job associated with the application to be evaluated.
- 1513 3. Retrieve the SMF 82 and/or 113 records associated with that job from the dataset.
- 1514 4. Run the supplied job to parse desired information from the SMF dataset.
- 1515 5. Generate the desired reports from the collected data.

#### 1516 Test Data

- SMF 82 log records ICSF and/or
- 1518 SMF 113 log records Processor Activity Instrumentation

#### 1519 Expected Results

- 1520 Discovery feature detects use of ICSF and CPACF crypto, including engines, algorithms, and key lengths
- used, with results captured in SMF data sets.

## 1522 D.2 Use Case 2

- 1523 Scenario
- 1524 Validate the discovery of post-quantum vulnerable algorithms in code that leverages ICSF cryptography1525 in COBOL application source.

#### 1526 Steps

- 1527 1. Download code to Application Discovery and Delivery Intelligence system.
- 1528 2. Start ADDI.
- 1529 3. Enable the JSON configuration file.
- 1530 4. Select the source code to analyze.
- 1531 5. Perform the ADDI build step.
- 1532 6. Review the report.
- 1533 Test Data
- 1534 COBOL source code containing:
- 1535 ICSF crypto calls (CSNBENC, CSNDDSG, CSNDPKE, CSNDEDH, etc.)
- Rule array keywords

#### 1537 Expected Results

- 1538 Discovery tool identifies and reports cryptographic calls, important parameters, call line numbers, and
- 1539 metadata, and writes it to a file for later display or export to a CSV file.

## 1540 **D.3 Use Case 3**

- 1541 Scenario
- 1542 Validate the discovery of post-quantum vulnerable algorithms in code that leverages a crypto API in
- 1543 COBOL application source.

#### 1544 Steps

- 1545 1. Download code to Application Discovery and Delivery Intelligence system.
- 1546 2. Update the JSON file to identify the crypto calls you would like to search for.
- 1547 3. Start ADDI.
- 1548 4. Enable the customconfiguration file.
- 1549 5. Select the source code to analyze.
- 1550 6. Perform the ADDI build step.
- 1551 7. Review the report.

#### 1552 Test Data

- 1553 COBOL source code containing:
- Crypto calls (e.g., c\_sign, c\_verify) as defined in JSON configuration file
- 1555 Parameters
- 1556 Expected Results
- 1557 Discovery tool identifies and reports cryptographic calls, important parameters, call line numbers, and
- 1558 metadata, and writes it to a file for later display or export to a CSV file.

## 1559 **D.4 Use Case 4**

#### 1560 Scenario

- 1561 Validate the discovery of post-quantum vulnerable algorithms in code that leverages cryptography in a
- 1562 CI/CD pipeline (IDE plugin).

#### 1563 Steps

- 1564 1. In an IDE, populate the data.
- 1565 2. Check-in and commit in git.
- 1566 3. Execute pipeline, which triggers ADDI's build.
- 1567 4. Perform the discovery.
- 1568 Test Data
- 1569 Source code with crypto calls such as:
- 1570 RSA Encryption/Decryption API usage

- RSA Sign/Verify API usage
- ECDSA Sign/Verify API usage
- EdDSA Sign/Verify API usage

#### 1574 Expected Results

- 1575 Discovery tool identifies files and lines of code that use the methods described in the test dataset and
- 1576 displays results within the IDE.

### 1577 **D.5 Use Case 5**

#### 1578 Scenario

1579 Validate the discovery of post-quantum vulnerable algorithms in network traffic.

#### 1580 Steps

- 1581 1. Capture a snapshot of network traffic by collecting SMF 119 records.
- 1582 2. Make SMF data available to z/OS Encryption Readiness Technology.
- 1583 3. Log on to z/OS Management Facility.
- 1584 4. Start the z/Network Analyzer.
- 1585 5. Select the run and manage query option.
- 1586 6. Create a query that looks for ECC/DH/DSA and other quantum-vulnerable algorithm usage.
- 1587 7. Run query to generate a report.

#### 1588 Test Data

- 1589 SMF 119 Data:
- Cipher suites
- 1591 TLS version
- Certificate information expiration, serial numbers, etc.
- 1593 IP address
- Protocol session identifiers
- Protection information key lengths, algorithm, etc.
- User ID, ports, jobnames

#### 1597 Expected Results

1598 Discovery tool can display on a dashboard and/or produce a report that identifies connections that 1599 are using vulnerable cryptography.

## 1600 **D.6 Use Case 6**

- 1601 Scenario
- 1602 Discover if quantum-vulnerable keys are managed by ICSF or RACF.

#### 1603 Steps

1604 1. Start the Crypto Analytics Tool.

- 16052. Take a snapshot of the certificates and keys and their related protection/authorization infor-1606mation stored in the RACF data base and the keys in the ICSF key data sets (CKDS, PKDS, and1607TKDS).
- 1608 3. Define a policy to identify the characteristics of the keys and certificates to be evaluated.
- 4. Apply the policy, which will create a report that identifies keys and certificates matching the cri-teria specified in the policy.
- 1611 5. Review the results.

#### 1612 Test Data

1613 ICSF key data sets and RACF information

#### 1614 Expected Results

- 1615 The discovery tool:
- 1616 Identifies keys that are considered quantum-vulnerable
- 1617 Identifies keys that match the desired search criteria
- 1618 Identifies characteristics of certificates managed by RACF
- Identifies callable services that are available for use in the HSM that should be disabled
- Displays information on the dashboard. Reports can also be generated.