Carnegie Mellon University
Software Engineering Institute

A State-Based Model for Multi-Party Coordinated Vulnerability Disclosure (MPCVD)

Allen Householder
Jonathan Spring

July 2021

SPECIAL REPORT
CMU/SEI-2021-SR-021
DOI: 10.1184/R1/16416771

CERT Division

http://www.sei.cmu.edu

This material is based upon work funded and supported by the Department of Homeland Security under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

This report was prepared for the SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100

DM21-0607

Table of Contents

List of Figures

List of Tables

Acknowledgments

We offer our sincere gratitude to the United States Senators, Members of Congress, and their staffs, whose insightful questions surrounding the coordinated disclosure of the Meltdown and Spectre vulnerabilities prompted us to recognize the need for a better way to reason about and describe what "good" vulnerability disclosure outcomes look like.

We also thank the anonymous reviewers at the ACM Journal *Digital Threats: Research and Practice* for their helpful feedback on an earlier version of this report.

CERT/CC staff were instrumental in reviewing and providing feedback on the model described in this report as it developed. We are grateful for the opportunity to have them as colleagues.

Abstract

Coordinated Vulnerability Disclosure (CVD) stands as a consensus response to the persistent fact of vulnerable software, yet few performance indicators have been proposed to measure its efficacy at the broadest scales. In this report, we seek to fill that gap. We begin by deriving a model of all possible CVD histories from first principles, organizing those histories into a partial ordering based on a set of desired criteria. We then compute a baseline expectation for the frequency of each desired criteria and propose a new set of performance indicators to measure the efficacy of CVD practices based on the differentiation of skill and luck in observation data. As a proof of concept, we apply these indicators to a variety of longitudinal observations of CVD practice and find evidence of significant skill to be prevalent. We conclude with reflections on how this model and its accompanying performance indicators could be used by various stakeholders (vendors, system owners, coordinators, and governments) to interpret the quality of their CVD practices.

1 Introduction

Software vulnerabilities remain pervasive. To date, there is little evidence that we are anywhere close to equilibrium between the introduction and elimination of vulnerabilities in deployed systems. The practice of Coordinated Vulnerability Disclosure (CVD) emerged as part of a growing consensus to develop normative behaviors in response to the persistent fact of vulnerable software. Yet while the basic principles of CVD have been established [17, 31, 30, 14], there has been limited work to measure the efficacy of CVD programs, especially at the scale of industry benchmarks. ISO 29147 [31] sets out the goals of vulnerability disclosure:

> a) ensuring that identified vulnerabilities are addressed;
> b) minimizing the risk from vulnerabilities;
> c) providing users with sufficient information to evaluate risks from vulnerabilities to their systems;
> d) setting expectations to promote positive communication and coordination among involved parties.

Meanwhile, the use of third party libraries and shared code components across vendors and their products creates a need to coordinate across those parties whenever a vulnerability is found in a shared component. While it can be difficult for stakeholders to ascertain the prevalence of components across products—and efforts such as the National Telecommunications and Information Administration (NTIA)'s Software Bill of Materials (SBOM) [55] are working to address the informational aspects of that problem—our concern here is the coordination of multiple parties in responding to the vulnerability.

Multi-Party Coordinated Vulnerability Disclosure (MPCVD) is a more complex form of CVD, involving the necessity to coordinate numerous stakeholders in the process of recognizing and fixing vulnerable products. Initial guidance from the Forum of Incident Response and Security Teams (FIRST) acknowledges the additional complexity that can arise in MPCVD cases [42]. The need for MPCVD arises from the complexities of the software supply chain. Its importance was illustrated by the Senate hearings about the Meltdown and Spectre vulnerabilities [39]. Nevertheless, the goals of CVD apply to MPCVD, as the latter is a generalization of the former.

The difficulty of MPCVD derives from the diversity of its stakeholders: different software vendors have different development budgets, schedules, tempos, and analysis capabilities to help them isolate, understand, and fix vulnerabilities. Additionally, they face diverse customer support expectations and obligations, plus an increasing variety of regulatory regimes governing some stakeholders but not others. For these reasons and many others, practitioners of MPCVD highlight *fairness* as a core difficulty in coordinating disclosures across vendors [30].

With the goal of minimizing the societal harm that results from the existence of a vulnerability in multiple products spread across multiple vendors, our motivating question is, "What does *fair* mean in MPCVD?" Optimizing MPCVD directly is not currently possible, as we lack a utility function to map from the events that occur in a given case to the impact that case has on the world. While this document does not fully address that problem, it sets out a number of steps toward a solution. We seek a way to sort MPCVD cases into better outcomes or worse outcomes. Ideally, the sorting criteria should based on unambiguous principles that are agreed upon and intelligible by all interested parties. Further, we seek a way to measure relevant features across MPCVD cases. Feature observability is a key factor: our measurement needs to be simple and repeatable without overly relying on proprietary or easily hidden information.

While a definition of *fairness* in MPCVD is a responsibility for the broader community, we focus on evaluating the skill of the coordinator. We expect this contributes to fairness based on the EthicsfIRST principles of ethics for incident response teams promoted by FIRST [22].[1] To that end, our research questions are:

RQ1 : Construct a model of CVD states amenable to analysis and also future generalization to MPCVD.

RQ2 : What is a reasonable baseline expectation for ordering of events in the model of CVD?

RQ3 : Given this baseline and model, does CVD as observed "in the wild" demonstrate skillful behavior?

This paper primarily focuses on the simpler case of CVD, with some initial thoughts towards extending it to MPCVD. This focus provides an opportunity for incremental analysis of the success of the model; more detailed MPCVD modeling can follow in future work.

1.1 Approach

The CERT® Coordination Center (CERT/CC) has a goal to improve the MPCVD process. Improvement involves automation. The creation of VINCE[2] is a significant step toward this goal, as it has helped us to recognize gaps in our own processes surrounding the MPCVD services we provide. As part of the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU), we also recognize that automation is made better when we can formalize process descriptions. In this report, we construct a toy model of the CVD process with the interest of a better understanding of how it might be formalized.

Our intent with this report is not to provide a complete solution to automate either CVD or MPCVD. Rather, this report is an attempt to systematize the basics in a way that can be extended by future work toward the specification of protocols that facilitate the automation of coordination tasks within MPCVD.

The model presented here provides a foundation on which we might build an MPCVD protocol. While we stop well short of a full protocol spec, we feel that this report contributes to improved understanding of the problems that such a protocol would need to address. And although an actual protocol would need to support a far more complicated process (i.e., the coordination and resolution of actual MPCVD cases), our contention is that we should be able to derive and learn quite a few of the basics from this toy model. A protocol that works on the toy model might not work in the real world. But any proposed real-world protocol should probably work on the toy model. The model is intended to be a sort of minimum acceptance test for any future protocol—if a proposed MPCVD process doesn't improve outcomes even in the toy model, one might wonder what it *is* doing.

1.2 Organization of This Document

We begin by deriving a model of all possible CVD case states and histories from first principles in §2 and §3, organizing those histories into a partial ordering based on a set of desired criteria in §4. We then compute a baseline expectation for the frequency of each desired criteria and propose a new set of performance indicators to measure the efficacy of CVD practices based on the differentiation of skill and luck in observation data in §5. As a proof of concept, we apply these indicators to a variety of longitudinal observations of CVD practice and find evidence of significant skill to be prevalent. In §6, we explore some of the implications and uses of such a model in any CVD case before extending it to MPCVD. The remainder

[1] Specifically, skill in our model will align with fulfilling the duty of coordinated vulnerability disclosure, duty of confidentiality, duty to inform, duty to team ability, and duty of evidence-based reasoning.

[2] CERT/CC Vulnerability Information and Coordination Environment (VINCE). `https://www.kb.cert.org/vince/`

of that section offers reflections on how this model and its accompanying performance indicators could be used by various stakeholders (vendors, system owners, coordinators, and governments) to interpret the quality of their CVD and MPCVD practices We continue with a review of related work in §7, future work in §8, and give our conclusions in §9.

An appendix summarizing each state in the model in conjunction with the discussion in §6 is also provided.

2 A State-based model for CVD

Our goal is to create a toy model of the MPCVD process that can shed light on the more complicated real thing. We begin by building up a state map of what FIRST refers to as bilateral CVD [42], which we will later expand into the MPCVD space. We start by defining a set of events of interest. We then use these to construct model states and the transitions between them.

2.1 Events in a Vulnerability Lifecycle

The goal of this section is to establish a model of events that affect the outcomes of vulnerability disclosure. Our model builds on previous models of the vulnerability lifecycle, specifically those of Arbaugh et al. [3], Frei et al. [24], and Bilge and et al. [10]. A more thorough literature review of vulnerability lifecycle models can be found in [37]. We are primarily interested in events that are usually observable to the stakeholders of a CVD case. Stakeholders include software vendors, vulnerability finder/reporters, coordinators, and deployers [30]. A summary of this model comparison is shown in Table 2.1.

Since we are modeling only the disclosure process, we assume the vulnerability both exists and is known to at least someone. Therefore we ignore the *birth* (*creation*, *introduced*) and *discovery* states as they are implied at the beginning of all possible vulnerability disclosure histories. We also omit the *anti-virus signatures released* event from [10] since we are not attempting to model vulnerability management operations in detail.

The first event we are interested in modeling is *Vendor Awareness*. This event corresponds to *Disclosure* in [3] and *vulnerability discovered by vendor* in [10] (this event is not modeled in [24]). In the interest of model simplicity, we are not concerned with *how* the vendor came to find out about the vulnerability's existence—whether it was found via internal testing, reported by a security researcher, or noticed as the result of incident analysis.

The second event we include is *Public Awareness* of the vulnerability. This event corresponds to *Publication* in [3], *time of public disclosure* in [24], and *vulnerability disclosed publicly* in [10]. The public might find out about a vulnerability through the vendor's announcement of a fix, a news report about a security breach, a conference presentation by a researcher, by comparing released software versions as in [57, 56], or a variety of other means. As above, we are primarily concerned with the occurrence of the event itself rather than the details of *how* the public awareness event arises.

The third event we address is *Fix Readiness*, by which we refer to the vendor's creation and possession of a fix that *could* be deployed to a vulnerable system, *if* the system owner knew of its existence. Here we differ somewhat from [3, 24, 10] in that their models address the *release* of the fix rather than its *readiness* for release.

The reason for this distinction will be made clear, but first we must mention that *Fix Deployed* is simply that: the fix exists, and it has been deployed.

We chose to include the *Fix Ready*, *Fix Deployed*, and *Public Awareness* events so that our model could better accommodate two common modes of modern software deployment:

- *shrinkwrap* - The traditional distribution mode in which the vendor and deployer are distinct entities and deployers must be made aware of the fix before it can be deployed. In this case, which corresponds to the previously mentioned *fix release* event, both fix readiness and public awareness are necessary for the fix to be deployed.

Table 2.1: Vulnerability Lifecycle Events: Comparing Models. Symbols for our model are defined in §2.4.

Arbaugh et al. [3]	Frei et al. [24]	Bilge et al. [10]	Our Model
Birth	creation (t_{creat})	introduced (t_c)	(implied)
Discovery	discovery (t_{disco})	n/a	(implied)
Disclosure	n/a	discovered by vendor (t_d)	Vendor Awareness ($\mathbf{V}$)
n/a	patch availability (t_{patch})	n/a	Fix Ready ($\mathbf{F}$)
Fix Release	n/a	patch released (t_p)	Fix Ready and Public Awareness
Publication	public disclosure (t_{discl})	disclosed publicly (t_0)	Public Awareness ($\mathbf{P}$)
n/a	patch installation (t_{insta})	patch deployment completed (t_a)	Fix Deployed ($\mathbf{D}$)
Exploit Automation	exploit availability (t_{explo})	Exploit released in wild (t_e)	n/a
Exploit Automation	n/a	n/a	Exploit Public ($\mathbf{X}$)
Exploit Automation	n/a	n/a	Attacks Observed ($\mathbf{A}$)
n/a	n/a	anti-virus signatures released (t_s)	n/a

- *SaaS* - A more recent delivery mode in which the vendor also plays the role of deployer. In this distribution mode, fix readiness can lead directly to fix deployed with no dependency on public awareness.

We note that so-called *silent fixes* by vendors can sometimes result in a fix being deployed without public awareness even if the vendor is not the deployer. Thus, it is possible (but unlikely) for *fix deployed* to occur before *public awareness* even in the *shrinkwrap* case above. It is also possible, and somewhat more likely, for *public awareness* to occur before *fix deployed* in the *SaaS* case as well.

We diverge from [3, 24, 10] again in our treatment of exploits and attacks. Because attacks and exploit publication are often discretely observable events, the broader concept of *exploit automation* in [3] is insufficiently precise for our use. Both [24, 10] focus on the availability of exploits rather than attacks, but the observability of their chosen events is hampered by attackers' incentives to maintain stealth. Frei et al. [24] uses *exploit availability*, whereas Bilge et al. [10] calls it *exploit released in wild*. Both refer to the state in which an exploit is known to exist. This can arise for at least two distinct reasons, which we wish to differentiate:

- *exploit public*—the method of exploitation for a vulnerability was made public in sufficient detail to be reproduced by others. Posting proof of concept (PoC) code to a widely available site or including the exploit in a commonly available exploit tool meets this criteria; privately held exploits do not.
- *attacks observed*—the vulnerability was observed to be exploited in attacks. This case requires evidence that the vulnerability was exploited; we can then presume the existence of an exploit regardless of its availability to the public. Analysis of malware from an incident might meet *attacks observed* but not *exploit public*, depending on how closely the attacker holds the malware. Use of a public exploit in an attack meets both *exploit public* and *attacks observed*.

Therefore, while we appreciate the existence of a hidden *exploit exists* event as causal predecessor of both *exploit public* and *attacks observed*, we assert no causal relationship between exploit public and attacks observed in our model. We make this choice in the interest of observability. The *exploit exists* event is difficult to consistently observe independently. Its occurrence is nearly always inferred from the observation of either *exploit public* or *attacks observed*.

Further discussion of related work can be found in §7.

2.2 Notation

Before we discuss CVD states (§2.3), transitions (§2.4), or possible histories (§3) in the vulnerability life cycle, we need to formally define our terms. In all of these definitions, we take standard Zermelo-Fraenkel set theory. The concept of sequences extends set theory to include a concept of ordered sets. From them, we adopt the following notation:

- $\{\dots\}$ An unordered set which makes no assertions about sequence
- $(\dots)$ An ordered set in which the items occur in that sequence
- The normal proper subset ($\subset$), equality ($=$), and subset ($\subseteq$) relations between sets
- The precedes ($\prec$) relation on members of an ordered set: $\sigma_i \prec \sigma_j$ if and only if $\sigma_i, \sigma_j \in s$ and $i < j$ where s is a sequence as defined in (3.1)

2.3 Deterministic Finite State Automata

Transitions during CVD resemble a Deterministic Finite Automaton (DFA) in that the transitions available to the current state are dependent on the state itself. Although DFAs are often used to determine whether the final or end state is acceptable, for analyzing CVD we are more interested in the order of the transitions. The usual DFA notation will still be effective for this modeling goal.

A DFA is defined as a 5-tuple $(\mathcal{Q}, \Sigma, \delta, q_0, F)$ [35].

- $\mathcal{Q}$ is a finite set of states
- Σ is a finite set of input symbols
- δ is a transition function $\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q}$
- $q_0 \in \mathcal{Q}$ is an initial state
- $\mathcal{F} \subseteq \mathcal{Q}$ is a set of final (or accepting) states

In our model, the state of the world is a specification of the current status of all the events in the vulnerability lifecycle model described in §2.1. We represent each of these statuses in vulnerability coordination by a letter for that part of the state of the world. For example, v means no vendor awareness and V means vendor is aware. The complete set of status labels is given in Table 2.2.

A state q represents the status of each of the six events. The possible states are all the combinations of the six event statuses. For state labels, lowercase letters designate events that have not occurred and uppercase letters designate events that have occurred in a particular state. For example, the state $VFdpXa$ represents vendor is aware, fix is ready, fix not deployed, no public awareness, exploit is public, and no attacks. The order in which the events occurred does not matter when defining the state. However, we will observe a notation convention keeping the letter names in the same case-insensitive order (v, f, d, p, x, a).

Table 2.2: Event status labels

Status	Meaning
v	Vendor is not aware of vulnerability
V	Vendor is aware of vulnerability
f	Fix is not ready
F	Fix is ready
d	Fix is not deployed
D	Fix is deployed
p	Public is not aware of vulnerability
P	Public is aware of vulnerability
x	No exploit has been made public
X	Exploit has been made public
a	No attacks have been observed
A	Attacks have been observed

All vulnerabilities start in the base state q_0 in which no events have occurred.

$$q_0 = vfdpxa \tag{2.1}$$

The lone final state in which all events have occurred is $VFDPXA$.

$$\mathcal{F} = \{VFDPXA\} \tag{2.2}$$

Note that this is a place where our model of vulnerability lifecycle histories diverges from what we expect to observe in vulnerability cases in the real world. There is ample evidence that most vulnerabilities never have exploits published or attacks observed [29, 34]. Therefore, practically speaking we might expect vulnerabilities to wind up in one of

$$\mathcal{F}' = \{VFDPxa, VFDPxA, VFDPXa, VFDPXA\}$$

at the time a case is closed. However, because we are modeling the observation of events as the transitions of a DFA, we allow for the possibility that an observed history remains incomplete at the time of case closure—in other words, it remains possible for exploits to be published or attacks to be observed long after a CVD case has been closed.

Intermediate states can be any combination of statuses, with the caveats elaborated in §2.4. In other words, valid states must contain one of the following strings: vfd, Vfd, VFd, or VFD.

As a result, there are thirty-two possible states, which we define as the set of all states $\mathcal{Q}$ in (2.3).

$$\begin{aligned}\mathcal{Q} \stackrel{\text{def}}{=} \{&vfdpxa, vfdPxa, vfdpXa, vfdPXa,\\ &vfdpxA, vfdPxA, vfdpXA, vfdPXA,\\ &Vfdpxa, VfdPxa, VfdpXa, VfdPXa,\\ &VfdpxA, VfdPxA, VfdpXA, VfdPXA,\\ &VFdpxa, VFdPxa, VFdpXa, VFdPXa,\\ &VFdpxA, VFdPxA, VFdpXA, VFdPXA,\\ &VFDpxa, VFDPxa, VFDpXa, VFDPXa,\\ &VFDpxA, VFDPxA, VFDpXA, VFDPXA\}\end{aligned} \tag{2.3}$$

When referring to a subset of states from $\mathcal{Q}$, any unlisted status remains unconstrained. For example,

$$\mathcal{Q}_{VFdP} = VFdP = \{VFdPxa, VFdPxA, VFdPXa, VFdPXA\}$$

In most cases, we will use the non-subscripted notation (e.g., $VFdP$). We will use the subscripted notation when it is needed to avoid confusion—for example, to disambiguate the status v from the set of states $\mathcal{Q}_v \subset \mathcal{Q}$ in which the status v holds.

2.4 State Transitions

In this section, we elaborate on the input symbols and transition function for our DFA.

2.4.1 Input Symbols

The input symbols to our DFA correspond to observations of the events outlined in Table 2.1. For our model, an input symbol σ is "read" when a participant observes a change in status (the vendor is notified, an exploit has been published, etc.). For the sake of simplicity, we begin with the assumption that observations are globally known—that is, a status change observed by any CVD participant is known to all. In the real world, the CVD process itself is well poised to ensure eventual consistency with this assumption through the communication of perceived case state across coordinating parties. We define the set of input symbols for our DFA as:

$$\Sigma \stackrel{\text{def}}{=} \{\mathbf{V}, \mathbf{F}, \mathbf{D}, \mathbf{P}, \mathbf{X}, \mathbf{A}\} \tag{2.4}$$

2.4.2 Transition Function

A DFA transitions between states according to its transition function δ based on which symbol is input to δ. On our way to defining the transition function δ for the complete model, we build up the allowed state transitions according to subsets of the complete set of six event statuses. State transitions correspond to the occurrence of a single event $\sigma \in \Sigma$. Because states correspond to the status of events that have or have not occurred, and all state transitions are non-reversible, the result will be an acyclic directed graph of states beginning at $q_0 = vfdpxa$ and ending at $\mathcal{F} = \{VFDPXA\}$ with allowed transitions as the edges.

In our model, many inputs would represent errors because no transition is possible given the model constraints we are about to describe. Yet a DFA requires a transition from every state for every possible input value. In the transition function tables to follow, we represent these transitions as "-". In an implementation these would result in an error condition and rejection of the input string. The state diagrams likewise omit the error state, although it is implied for any transition not explicitly depicted.

It is common to use subscripts on the transition function δ corresponding to the partial function δ_σ on states for a specific symbol $\sigma \in \Sigma$. However, because the symbols in Σ are so closely tied to both the state names and the transitions between them, we will avoid the subscripts and just use the symbols themselves (bold capital letters) as proxies for the subscripted transition function. For notation purposes, transitions between sets of states will use an arrow with the specific event $\sigma \in \Sigma$ as its label ($\xrightarrow{\sigma}$). For example,

$$vfdpxa \xrightarrow{\mathbf{V}} Vfdpxa$$

is equivalent to

$$\delta_V(vfdpxa) = Vfdpxa \text{ and also } \delta(vfdpxa, \mathbf{V}) = Vfdpxa$$

and indicates the transition from the base state caused by notifying the vendor ($\mathbf{V}$).

Table 2.3: Transition function δ_{VFD} for the vendor fix path

State (q)	**V**	**F**	**D**
vfd	Vfd	-	-
Vfd	-	VFd	-
VFd	-	-	VFD
VFD	-	-	-

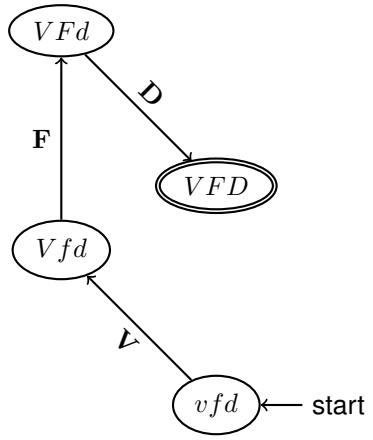


*Figure 2.1: Submap of vendor fix path state transitions (**V**, **F**, **D**)*

Vendor Fix Path The primary aspect of our model is the vendor fix flow. The relevant events and corresponding transitions in this dimension are Vendor Awareness (**V**), Fix Available (**F**), and Fix Deployed (**D**). A vendor cannot produce a fix and make it ready if the vendor is not aware of the problem, and the fix cannot be deployed if it is not ready. Therefore, we take the precedence relation $\mathbf{V} \prec \mathbf{F} \prec \mathbf{D}$ as a strong constraint on possible sequences.

The DFA specification for this submodel is given in (2.5). The resulting state subsets and transitions are as shown in Table 2.3 and Figure 2.1. The double circle in Figure 2.1 and subsequent state diagrams indicates the final state $\mathcal{F}$ for that submap.

$$\begin{aligned} \mathcal{Q}_{VFD} &= \{vfd, Vfd, VFd, VFD\} \\ \Sigma_{VFD} &= \{\mathbf{V}, \mathbf{F}, \mathbf{D}\} \\ \delta_{VFD} &= \text{see Table 2.3} \\ q^0_{VFD} &= vfd \\ \mathcal{F}_{VFD} &= \{VFD\} \end{aligned} \tag{2.5}$$

Private vs Public Awareness A second aspect of vulnerability disclosure is whether or not a vulnerability is known to the public. The public may become aware of a vulnerability for a number of reasons, including:

- the vendor publishes a fix
- the researcher publishes a report about the vulnerability
- exploit code is made available to the public
- attackers are found to be exploiting the vulnerability and this information is made public

Table 2.4: Transition function δ_{VFDP} for the vendor fix path incorporating public awareness

State	**V**	**F**	**D**	**P**
$vfdp$	$Vfdp$	-	-	$vfdP$
$Vfdp$	-	$VFdp$	-	$VfdP$
$VFdp$	-	-	$VFDp$	$VFdP$
$VFDp$	-	-	-	$VFDP$
$vfdP$	$VfdP$	-	-	-
$VfdP$	-	$VFdP$	-	-
$VFdP$	-	-	$VFDP$	-
$VFDP$	-	-	-	-

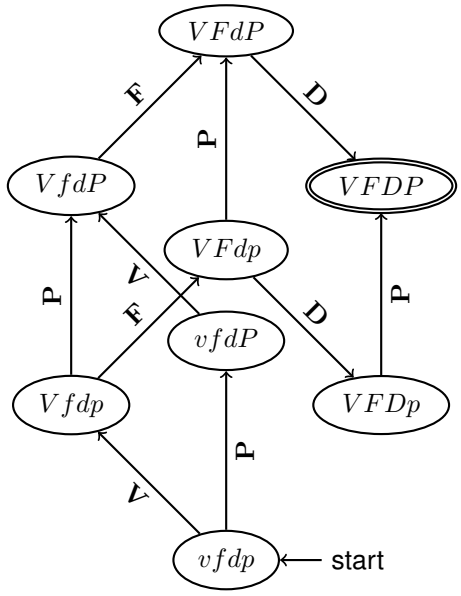


*Figure 2.2: Submap of vendor fix path state transitions with public awareness (**V**, **F**, **D**, **P**)*

Table 2.5: Transition function δ_{PXA} for the non-fix path transitions

State	**P**	**X**	**A**
pxa	Pxa	pXa	pxA
pxA	PxA	pXA	-
pXa	PXa	-	-
pXA	PXA	-	-
Pxa	-	PXa	PxA
PxA	-	PXA	-
PXa	-	-	PXA
PXA	-	-	-

Our model assumes that vendors immediately become aware of what the public is aware of. Therefore, all states in vP are unstable, and must lead to the corresponding state in VP in the next step.

CVD attempts to move vulnerabilities through states belonging to p until the process reaches a state in $VFdp$ at least. Vendors that can control deployment will likely prefer the transition from $VFDp \xrightarrow{\mathbf{P}} VFDP$. On the other hand, vulnerabilities requiring system owner action to deploy fixes will be forced to transition through $VFdp \xrightarrow{\mathbf{P}} VFdP \xrightarrow{\mathbf{D}} VFDP$ instead since public awareness is required in order to prompt such action. This implies that states in $VFDp$ are unreachable to vendors whose distribution model requires system owner action to deploy fixes.

The DFA specification for this submodel is given in (2.6). Table 2.4 shows the transition function δ_{VFDP}, while Figure 2.2 depicts the transitions among these states.

$$\begin{aligned} \mathcal{Q}_{VFDP} &= \{vfdp, vfdP, Vfdp, VfdP, \\ &\qquad VFdp, VFdP, VFDp, VFDP\} \\ \Sigma_{VFDP} &= \{\mathbf{V}, \mathbf{F}, \mathbf{D}, \mathbf{P}\} \\ \delta_{VFDP} &= \text{see Table 2.4} \\ q^0_{VFDP} &= vfdp \\ \mathcal{F}_{VFDP} &= \{VFDP\} \end{aligned} \tag{2.6}$$

Public awareness, exploits, and attacks Before fully integrating all thirty two states, we pause here to develop a three dimensional sub-model that highlights the interaction of public awareness, exploit publication, and attacks. Unlike the causal relationship representing the vendor process in Figure 2.1, these three transitions can occur independently. We therefore treat them as their own dimensions, as shown in Figure 2.3.

Because we defined **X** to correspond to the public availability of an exploit, as a simplification we impose the constraint that exploit publication leads directly to public awareness when the public was previously unaware of the vulnerability. For practical purposes, this constraint means that all states in pX are unstable and must lead to the corresponding state in PX in the subsequent step. As a result, transitions from pXa to pXA are disallowed, as reflected in Figure 2.3. The transition function δ_{PXA} is given in Table 2.5. Further discussion of this transition can be found in §6.5.1.

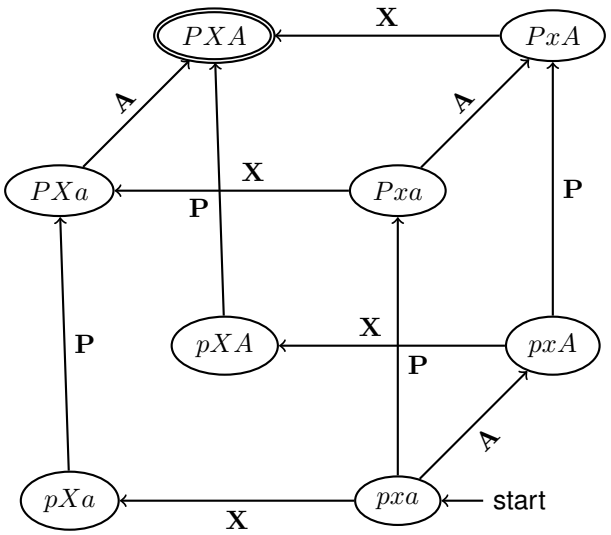


*Figure 2.3: Submap of non-fix path state transitions (**P**, **X**, **A**)*

$$
\begin{aligned}
\mathcal{Q}_{PXA} &= \{pxa, pxA, pXa, pXA, \\
&\qquad Pxa, PxA, PXa, PXA\} \\
\Sigma_{PXA} &= \{\mathbf{P}, \mathbf{X}, \mathbf{A}\} \\
\delta_{PXA} &= \text{see Table 2.5} \\
q^0_{PXA} &= pxa \\
\mathcal{F}_{PXA} &= \{PXA\}
\end{aligned}
\tag{2.7}
$$

In this model, attacks observed need not immediately cause public awareness, although that can happen. Our reasoning is that the connection between attacks and exploited vulnerabilities is often made later during incident analysis. While the attack itself may have been observed much earlier, the knowledge of which vulnerability it targeted may be delayed until after other events have occurred. In other words, although pA does not require an immediate transition to PA the way $pX \xrightarrow{\mathbf{P}} PX$ does, it does seem plausible that the likelihood of **P** occurring increases when attacks are occurring. Logically, this is a result of there being more ways for the public to discover the vulnerability when attacks are happening than when they are not. For states in pa, the public depends on the normal vulnerability discovery and reporting process. States in pA include that possibility and add the potential for discovery as a result of security incident analysis.

Five Dimensions of CVD By composing these sub-parts, we arrive at our complete state transition model, which we construct by combining the vendor fix path $vfd \rightarrow Vfd \rightarrow VFd \rightarrow VFD$ defined by (2.5) and its extension in (2.6) with the PXA cube defined by (2.7). The complete map is shown in Figure 2.4. We also can now define the transition function δ for the entire model, as shown in Table 2.6. A summary of the complete DFA specification is given in (2.8).

Table 2.6: Transition function δ for the full model

State q	Σ **V**	**F**	**D**	**P**	**X**	**A**
$vfdpxa$	$Vfdpxa$	-	-	$vfdPxa$	$vfdpXa$	$vfdpxA$
$vfdpxA$	$VfdpxA$	-	-	$vfdPxA$	$vfdpXA$	-
$vfdpXa$	-	-	-	$vfdPXa$	-	-
$vfdpXA$	-	-	-	$vfdPXA$	-	-
$vfdPxa$	$VfdPxa$	-	-	-	-	-
$vfdPxA$	$VfdPxA$	-	-	-	-	-
$vfdPXa$	$VfdPXa$	-	-	-	-	-
$vfdPXA$	$VfdPXA$	-	-	-	-	-
$Vfdpxa$	-	$VFdpxa$	-	$VfdPxa$	$VfdpXa$	$VfdpxA$
$VfdpxA$	-	$VFdpxA$	-	$VfdPxA$	$VfdpXA$	-
$VfdpXa$	-	-	-	$VfdPXa$	-	-
$VfdpXA$	-	-	-	$VfdPXA$	-	-
$VfdPxa$	-	$VFdPxa$	-	-	$VfdPXa$	$VfdPxA$
$VfdPxA$	-	$VFdPxA$	-	-	$VfdPXA$	-
$VfdPXa$	-	$VFdPXa$	-	-	-	$VfdPXA$
$VfdPXA$	-	$VFdPXA$	-	-	-	-
$VFdpxa$	-	-	$VFDpxa$	$VFdPxa$	$VFdpXa$	$VFdpxA$
$VFdpxA$	-	-	$VFDpxA$	$VFdPxA$	$VFdpXA$	-
$VFdpXa$	-	-	-	$VFdPXa$	-	-
$VFdpXA$	-	-	-	$VFdPXA$	-	-
$VFdPxa$	-	-	$VFDPxa$	-	$VFdPXa$	$VFdPxA$
$VFdPxA$	-	-	$VFDPxA$	-	$VFdPXA$	-
$VFdPXa$	-	-	$VFDPXa$	-	-	$VFdPXA$
$VFdPXA$	-	-	$VFDPXA$	-	-	-
$VFDpxa$	-	-	-	$VFDPxa$	$VFDpXa$	$VFDpxA$
$VFDpxA$	-	-	-	$VFDPxA$	$VFDpXA$	-
$VFDpXa$	-	-	-	$VFDPXa$	-	-
$VFDpXA$	-	-	-	$VFDPXA$	-	-
$VFDPxa$	-	-	-	-	$VFDPXa$	$VFDPxA$
$VFDPxA$	-	-	-	-	$VFDPXA$	-
$VFDPXa$	-	-	-	-	-	$VFDPXA$
$VFDPXA$	-	-	-	-	-	-

$$
\begin{aligned}
\mathcal{Q} &= \text{see (2.3)} \\
\Sigma &= \text{see (2.4)} \\
\delta &= \text{see Table 2.6} \\
q_0 &= \text{see (2.1)} \\
\mathcal{F} &= \text{see (2.2)}
\end{aligned}
\tag{2.8}
$$

In this combined model, each point along the vendor fix flow in Figure 2.1 corresponds to an instance of the public/exploit/attack cube from Figure 2.3. Figure 2.4 shows each of these as distinct cubes embedded in the larger model.

The *ignorant vendor* cube (vfd) Found at the lower right of Figure 2.4, the vfd cube is the least stable of the four because many of its internal transitions are disallowed, owing to the instability of both pX and vP. The effect is a higher likelihood of exiting this

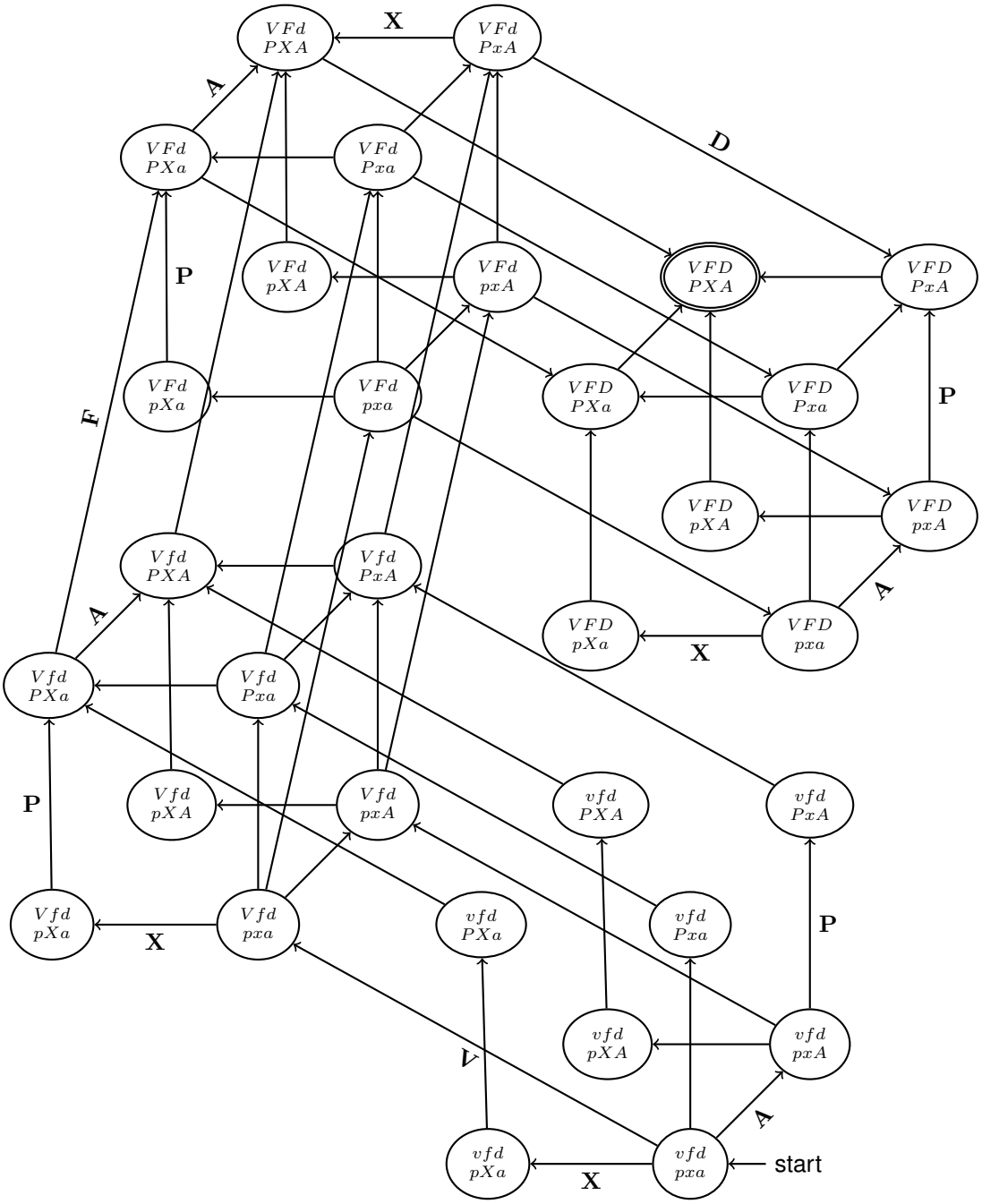


*Figure 2.4: Complete map of the 32 possible states in our model of vulnerability disclosure and their allowed transitions (***V***, **F**, **D**, **P**, **X**, **A**)*

Table 2.7: Semantic encoding of states in $\mathcal{S}$

Bit Position	0	1
0	$\mathcal{Q}_v \cup \mathcal{Q}_D$	$\mathcal{Q}_{Vd}$
1	$\mathcal{Q}_f$	$\mathcal{Q}_F$
2	$\mathcal{Q}_p$	$\mathcal{Q}_P$
3	$\mathcal{Q}_x$	$\mathcal{Q}_X$
4	$\mathcal{Q}_a$	$\mathcal{Q}_A$

cube than the others. The practical interpretation is that vendors are likely to become aware of vulnerabilities that exist in their products barring significant effort on the part of adversaries to prevent exiting the vfd states.

The *vendor aware* cube (Vfd) In this cube, the vendor is aware of the vulnerability, but the fix is not yet ready. Vulnerabilities remain in Vfd until the vendor produces a fix.

The *fix available* cube (VFd) States in this cube share the fact that a fix is available but not yet deployed. Many publicly-disclosed vulnerabilities spend a sizable amount of time in this cube as they await system owner or deployer action to deploy the fix.

The *fix deployed* cube (VFD) This cube is a sink: once it is reached, there are no exits. Attacks attempted in this cube are expected to fail. The broader the scope of one's concern in terms of number of systems, the less certain one can be of having reached this cube. It is rather easy to tell when a single installed instance of vulnerable software has been patched. It is less easy to tell when the last of thousands or even millions of vulnerable software instances across an enterprise has been fixed.

The states and transitions of the model can be represented as a partial Hamming cube in in 5 dimensions. In this representation, each state maps onto a binary value between 00000 and 11111, corresponding to the 32 vertices of the 5-dimensional Hamming Cube. The semantics of each bit position from left to right are given in Table 2.7. Correspondingly, each transition represents a single bit flip in the state encoding. Some edges (transitions) are disallowed by the causal requirements described in this section and formalized in the next section (see (3.3), (3.4), and (3.5)). This observation serves as the basis of the visualization given in Figure 2.4.

3 Sequences of Events and Possible Histories in CVD

In §2, we began by identifying a set of events of interest in CVD cases. Then we constructed a state model describing how the occurrence of these events can interact with each other. In this section, we look at paths through the resulting state model.

A sequence s is an ordered set of some number of events $\sigma_i \in \Sigma$ for $1 \leq i \leq n$ and the length of s is $|s| \stackrel{\text{def}}{=} n$. In other words, a sequence s is an input string to the DFA defined in §2.

$$s \stackrel{\text{def}}{=} (\sigma_1, \sigma_2, \ldots \sigma_n) \tag{3.1}$$

A vulnerability disclosure history h is a sequence s containing one and only one of each of the symbols in Σ; by definition $|h| = |\Sigma| = 6$. Note this is a slight abuse of notation; $|\ |$ represents both sequence length and the cardinality of a set.

$$\begin{aligned} h \stackrel{\text{def}}{=} s : &\forall \sigma_i, \sigma_j \in s \text{ it is the case that } \sigma_i \neq \sigma_j \text{ and} \\ &\forall \sigma_k \in \Sigma \text{ it is the case that } \exists \sigma_i \in s \text{ such that } \sigma_k = \sigma_i \end{aligned} \tag{3.2}$$

where two members of the set Σ are equal if they are represented by the same symbol and not equal otherwise. The set of all potential histories, $\mathcal{H}_p$, is a set of all the sequences h that satisfy this definition.

3.1 The Possible Histories of CVD

Given that a history h contains all six events Σ in some order, there could be at most 720 ($_6\mathrm{P}_6 = 6! = 720$) potential histories. However, because of the causal requirements outlined in 2.4.2, we know that Vendor Awareness (**V**) must precede Fix Ready (**F**) and that Fix Ready must precede Fix Deployed (**D**).

The DFA developed in §2 provides the mechanism to validate histories: a history h is valid if the DFA accepts it as a valid input string. Once this constraint is applied, only 70 possible histories $h \in \mathcal{H}p$ remain viable. We denote the set of all such valid histories as $\mathcal{H}$ and have $|\mathcal{H}| = 70$. The set of possible histories $\mathcal{H}$ corresponds to the 70 allowable paths through $\mathcal{Q}$ as can be derived from Table 2.6 and Fig. 2.4.

The set of possible histories $\mathcal{H}$ is listed exhaustively in Table 3.1. Commas and parentheses indicating ordered sets were omitted from column h for readability. The skill ranking function on the histories will be defined in §4.4. The desirability of the history ($\mathbb{D}^h$) will be defined in §3.2. The expected frequency of each history f_h is explained in §4.1.

Table 3.1: Possible Histories $h \in \mathcal{H}$ of CVD

#	h	rank	$\lvert\mathbb{D}^h\rvert$	f_h	**A** ϒ **D**	**P** ϒ **D**	**X** ϒ **D**	**A** ϒ **F**	**P** ϒ **F**	**X** ϒ **F**	**A** ϒ **P**	**X** ϒ **P**	**A** ϒ **V**	**P** ϒ **V**	**X** ϒ **V**	**A** ϒ **X**
0	**AXPVFD**	1	0	0.0833	0	0	0	0	0	0	0	0	0	0	0	0
1	**APVXFD**	2	2	0.0417	0	0	0	0	0	0	0	1	0	0	1	0
2	**AVXPFD**	3	2	0.0278	0	0	0	0	0	0	0	0	0	1	1	0

Continued on next page

Table 3.1: Possible Histories $h \in \mathcal{H}$ of CVD

#	h	rank	$\lvert\mathbb{D}^h\rvert$	f_h	**D** $\overset{\mathbf{A}}{\gamma}$	**D** $\overset{\mathbf{P}}{\gamma}$	**D** $\overset{\mathbf{X}}{\gamma}$	**F** $\overset{\mathbf{A}}{\gamma}$	**F** $\overset{\mathbf{P}}{\gamma}$	**F** $\overset{\mathbf{X}}{\gamma}$	**P** $\overset{\mathbf{A}}{\gamma}$	**P** $\overset{\mathbf{X}}{\gamma}$	**V** $\overset{\mathbf{A}}{\gamma}$	**V** $\overset{\mathbf{P}}{\gamma}$	**V** $\overset{\mathbf{X}}{\gamma}$	**X** $\overset{\mathbf{A}}{\gamma}$
3	**XPVAFD**	4	3	0.1250	0	0	0	0	0	0	1	0	1	0	0	1
4	**VAXPFD**	5	3	0.0208	0	0	0	0	0	0	0	0	1	1	1	0
5	**PVAXFD**	6	4	0.0417	0	0	0	0	0	0	1	1	1	0	1	0
6	**AVPXFD**	7	3	0.0139	0	0	0	0	0	0	0	1	0	1	1	0
7	**APVFXD**	7	3	0.0208	0	0	0	0	0	1	0	1	0	0	1	0
8	**XPVFAD**	8	4	0.0625	0	0	0	1	0	0	1	0	1	0	0	1
9	**VAPXFD**	9	4	0.0104	0	0	0	0	0	0	0	1	1	1	1	0
10	**PVXAFD**	10	5	0.0417	0	0	0	0	0	0	1	1	1	0	1	1
11	**VPAXFD**	11	5	0.0104	0	0	0	0	0	0	1	1	1	1	1	0
12	**PVAFXD**	11	5	0.0208	0	0	0	0	0	1	1	1	1	0	1	0
13	**VXPAFD**	11	5	0.0312	0	0	0	0	0	0	1	0	1	1	1	1
14	**AVPFXD**	12	4	0.0069	0	0	0	0	0	1	0	1	0	1	1	0
15	**APVFDX**	13	4	0.0208	0	0	1	0	0	1	0	1	0	0	1	0
16	**VAPFXD**	14	5	0.0052	0	0	0	0	0	1	0	1	1	1	1	0
17	**XPVFDA**	15	5	0.0625	1	0	0	1	0	0	1	0	1	0	0	1
18	**PVXFAD**	16	6	0.0208	0	0	0	1	0	0	1	1	1	0	1	1
19	**AVFXPD**	17	4	0.0093	0	0	0	0	1	1	0	0	0	1	1	0
20	**VPXAFD**	18	6	0.0104	0	0	0	0	0	0	1	1	1	1	1	1
21	**PVFAXD**	19	6	0.0139	0	0	0	1	0	1	1	1	1	0	1	0
22	**VXPFAD**	19	6	0.0156	0	0	0	1	0	0	1	0	1	1	1	1
23	**VPAFXD**	20	6	0.0052	0	0	0	0	0	1	1	1	1	1	1	0
24	**VAFXPD**	21	5	0.0069	0	0	0	0	1	1	0	0	1	1	1	0
25	**PVAFDX**	22	6	0.0208	0	0	1	0	0	1	1	1	1	0	1	0
26	**AVPFDX**	23	5	0.0069	0	0	1	0	0	1	0	1	0	1	1	0
27	**AVFPXD**	24	5	0.0046	0	0	0	0	1	1	0	1	0	1	1	0
28	**PVFXAD**	25	7	0.0139	0	0	0	1	0	1	1	1	1	0	1	1
29	**VPXFAD**	25	7	0.0052	0	0	0	1	0	0	1	1	1	1	1	1
30	**VAPFDX**	26	6	0.0052	0	0	1	0	0	1	0	1	1	1	1	0
31	**VAFPXD**	27	6	0.0035	0	0	0	0	1	1	0	1	1	1	1	0
32	**PVXFDA**	28	7	0.0208	1	0	0	1	0	0	1	1	1	0	1	1
33	**VPFAXD**	29	7	0.0035	0	0	0	1	0	1	1	1	1	1	1	0
34	**VFAXPD**	30	6	0.0052	0	0	0	1	1	1	0	0	1	1	1	0
35	**VXPFDA**	31	7	0.0156	1	0	0	1	0	0	1	0	1	1	1	1
36	**PVFADX**	32	7	0.0139	0	0	1	1	0	1	1	1	1	0	1	0
37	**VPAFDX**	33	7	0.0052	0	0	1	0	0	1	1	1	1	1	1	0
38	**VPFXAD**	34	8	0.0035	0	0	0	1	0	1	1	1	1	1	1	1
39	**AVFPDX**	35	6	0.0046	0	0	1	0	1	1	0	1	0	1	1	0
40	**VFAPXD**	36	7	0.0026	0	0	0	1	1	1	0	1	1	1	1	0
41	**VPXFDA**	37	8	0.0052	1	0	0	1	0	0	1	1	1	1	1	1
42	**PVFXDA**	37	8	0.0139	1	0	0	1	0	1	1	1	1	0	1	1
43	**VAFPDX**	38	7	0.0035	0	0	1	0	1	1	0	1	1	1	1	0
44	**VPFADX**	39	8	0.0035	0	0	1	1	0	1	1	1	1	1	1	0
45	**VFPAXD**	40	8	0.0026	0	0	0	1	1	1	1	1	1	1	1	0
46	**VFXPAD**	41	8	0.0078	0	0	0	1	1	1	1	0	1	1	1	1
47	**AVFDXP**	42	6	0.0046	0	1	1	0	1	1	0	0	0	1	1	0
48	**PVFDAX**	43	8	0.0139	1	0	1	1	0	1	1	1	1	0	1	0
49	**VAFDXP**	44	7	0.0035	0	1	1	0	1	1	0	0	1	1	1	0
50	**VPFXDA**	45	9	0.0035	1	0	0	1	0	1	1	1	1	1	1	1
51	**VFAPDX**	46	8	0.0026	0	0	1	1	1	1	0	1	1	1	1	0
52	**VFPXAD**	46	9	0.0026	0	0	0	1	1	1	1	1	1	1	1	1
53	**AVFDPX**	47	7	0.0046	0	1	1	0	1	1	0	1	0	1	1	0

Continued on next page

Table 3.1: Possible Histories $h \in \mathcal{H}$ of CVD

#	h	rank	$\lvert\mathbb{D}^h\rvert$	f_h	$\mathbf{A} \curlyvee \mathbf{D}$	$\mathbf{P} \curlyvee \mathbf{D}$	$\mathbf{X} \curlyvee \mathbf{D}$	$\mathbf{A} \curlyvee \mathbf{F}$	$\mathbf{P} \curlyvee \mathbf{F}$	$\mathbf{X} \curlyvee \mathbf{F}$	$\mathbf{A} \curlyvee \mathbf{P}$	$\mathbf{X} \curlyvee \mathbf{P}$	$\mathbf{A} \curlyvee \mathbf{V}$	$\mathbf{P} \curlyvee \mathbf{V}$	$\mathbf{X} \curlyvee \mathbf{V}$	$\mathbf{A} \curlyvee \mathbf{X}$
54	**PVFDXA**	48	9	0.0139	1	0	1	1	0	1	1	1	1	0	1	1
55	**VPFDAX**	49	9	0.0035	1	0	1	1	0	1	1	1	1	1	1	0
56	**VFXPDA**	50	9	0.0078	1	0	0	1	1	1	1	0	1	1	1	1
57	**VFPADX**	51	9	0.0026	0	0	1	1	1	1	1	1	1	1	1	0
58	**VAFDPX**	52	8	0.0035	0	1	1	0	1	1	0	1	1	1	1	0
59	**VFADXP**	53	8	0.0026	0	1	1	1	1	1	0	0	1	1	1	0
60	**VPFDXA**	54	10	0.0035	1	0	1	1	0	1	1	1	1	1	1	1
61	**VFPXDA**	55	10	0.0026	1	0	0	1	1	1	1	1	1	1	1	1
62	**VFADPX**	56	9	0.0026	0	1	1	1	1	1	0	1	1	1	1	0
63	**VFPDAX**	57	10	0.0026	1	0	1	1	1	1	1	1	1	1	1	0
64	**VFDAXP**	58	9	0.0026	1	1	1	1	1	1	0	0	1	1	1	0
65	**VFPDXA**	59	11	0.0026	1	0	1	1	1	1	1	1	1	1	1	1
66	**VFDAPX**	60	10	0.0026	1	1	1	1	1	1	0	1	1	1	1	0
67	**VFDXPA**	61	11	0.0052	1	1	1	1	1	1	1	0	1	1	1	1
68	**VFDPAX**	61	11	0.0026	1	1	1	1	1	1	1	1	1	1	1	0
69	**VFDPXA**	62	12	0.0026	1	1	1	1	1	1	1	1	1	1	1	1

Now that we have defined the set of histories $\mathcal{H}$, we can summarize the effects of the transition function δ developed in §2.4 (Table 2.6) as a set of patterns it imposes on all histories $h \in \mathcal{H}$. First, the causality constraint of the vendor fix path must hold.

$$\mathbf{V} \prec \mathbf{F} \prec \mathbf{D} \tag{3.3}$$

Second, the model makes the simplifying assumption that vendors know at least as much as the public does. In other words, all histories must meet one of two criteria: either Vendor Awareness precedes Public Awareness (**P**) or Vendor Awareness must immediately follow it.

$$\mathbf{V} \prec \mathbf{P} \text{ or } \mathbf{P} \rightarrow \mathbf{V} \tag{3.4}$$

Third, the model assumes that the public can be informed about a vulnerability by a public exploit. Therefore, either Public Awareness precedes Exploit Public (**X**) or must immediately follow it.

$$\mathbf{P} \prec \mathbf{X} \text{ or } \mathbf{X} \rightarrow \mathbf{P} \tag{3.5}$$

This model is amenable for analysis of CVD, but we need to add a way to express preferences before it is complete. Thus we are part way through **RQ1**. §6.2 will address how this model can generalize from CVD to MPCVD.

3.2 On the Desirability of Possible Histories

All histories are not equally preferable. Some are quite bad—for example, those in which attacks precede vendor awareness ($\mathbf{A} \prec \mathbf{V}$). Others are very desirable—for example, those in which fixes are deployed before either an exploit is made public ($\mathbf{D} \prec \mathbf{X}$) or attacks occur ($\mathbf{D} \prec \mathbf{A}$).

In pursuit of a way to reason about our preferences for some histories over others, we define the following preference criteria: history h_a is preferred over history h_b if, all else being equal,

a more desirable event σ_1 precedes a less desirable event σ_2. This preference is denoted as $\sigma_1 \prec \sigma_2$. We define the following ordering preferences:

- $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$, or $\mathbf{V} \prec \mathbf{A}$—Vendors can take no action to produce a fix if they are unaware of the vulnerability. Public awareness prior to vendor awareness can cause increased support costs for vendors at the same time they are experiencing increased pressure to prepare a fix. If public awareness of the vulnerability prior to vendor awareness is bad, then a public exploit is at least as bad because it encompasses the former and makes it readily evident that adversaries have exploit code available for use. Attacks prior to vendor awareness represent a complete failure of the vulnerability remediation process because they indicate that adversaries are far ahead of defenders.
- $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$, or $\mathbf{F} \prec \mathbf{A}$—As noted above, the public can take no action until a fix is ready. Because public awareness also implies adversary awareness, the vendor/adversary race becomes even more critical if this condition is not met. When fixes exist before exploits or attacks, defenders are better able to protect their users.
- $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, or $\mathbf{D} \prec \mathbf{A}$—Even better than vendor awareness and fix availability prior to public awareness, exploit publication or attacks are scenarios in which fixes are deployed prior to one or more of those transitions.
- $\mathbf{P} \prec \mathbf{X}$ or $\mathbf{P} \prec \mathbf{A}$—In many cases, fix deployment ($\mathbf{D}$) requires system owners to take action, which implies a need for public awareness of the vulnerability. We therefore prefer histories in which public awareness happens prior to either exploit publication or attacks.
- $\mathbf{X} \prec \mathbf{A}$—This criteria is not about whether exploits should be published or not.[1] It is about whether we should prefer histories in which exploits are published *before* attacks happen over histories in which exploits are published *after* attacks happen. Our position is that attackers have more advantages in the latter case than the former, and therefore we should prefer histories in which $\mathbf{X} \prec \mathbf{A}$.

Equation 3.6 formalizes our definition of desired orderings $\mathbb{D}$. Table 3.3 displays all 36 possible orderings of paired transitions and whether they are considered impossible, required (as defined by (3.3), (3.4), and (3.5)), desirable (as defined by (3.6)), or undesirable (the complement of the set defined in (3.6)).

Before proceeding, we note that our model focuses on the ordering of transitions, not their timing. We acknowledge that in some situations, the interval between transitions may be of more interest than merely the order of those transitions, as a rapid tempo of transitions can alter the options available to stakeholders in their response. We discuss this limitation further in §8; however, the following model posits event sequence timing on a human-oriented timescale measured in minutes to weeks.

$$\begin{aligned}\mathbb{D} \stackrel{\text{def}}{=} \{&\mathbf{V} \prec \mathbf{P}, \mathbf{V} \prec \mathbf{X}, \mathbf{V} \prec \mathbf{A},\\ &\mathbf{F} \prec \mathbf{P}, \mathbf{F} \prec \mathbf{X}, \mathbf{F} \prec \mathbf{A},\\ &\mathbf{D} \prec \mathbf{P}, \mathbf{D} \prec \mathbf{X}, \mathbf{D} \prec \mathbf{A},\\ &\mathbf{P} \prec \mathbf{X}, \mathbf{P} \prec \mathbf{A}, \mathbf{X} \prec \mathbf{A}\}\end{aligned} \tag{3.6}$$

An element $d \in \mathbb{D}$ is of the form $\sigma_i \prec \sigma_j$. More formally, d is a relation of the form $d(\sigma_1, \sigma_2, \prec)$. $\mathbb{D}$ is a set of such relations.

[1] Although we do believe there is some value in exploit publication because it allows defenders to develop detection controls (e.g., in the form of behavioral patterns or signatures). Even if those detection mechanisms are imperfect, it seems better that they be in place prior to adversaries using them than the opposite.

Table 3.2: Desired event precedence mapped to subsets of states

Event Precedence (d)	State Subsets to Prefer	State Subsets to Avoid
$\mathbf{V} \prec \mathbf{X}$	Vx	vX
$\mathbf{V} \prec \mathbf{A}$	Va	vA
$\mathbf{V} \prec \mathbf{P}$	Vp	vP
$\mathbf{P} \prec \mathbf{X}$	Px	pX
$\mathbf{F} \prec \mathbf{X}$	VFx	fdX
$\mathbf{P} \prec \mathbf{A}$	Pa	pA
$\mathbf{F} \prec \mathbf{A}$	VFa	fdA
$\mathbf{F} \prec \mathbf{P}$	VFp	fdP
$\mathbf{D} \prec \mathbf{X}$	$VFDx$	dX
$\mathbf{X} \prec \mathbf{A}$	Xa	xA
$\mathbf{D} \prec \mathbf{A}$	$VFDa$	dA
$\mathbf{D} \prec \mathbf{P}$	$VFDp$	dP

Some states are preferable to others The desiderata in (3.6) address the preferred ordering of transitions in CVD histories, which imply that one should prefer to pass through some states and avoid others. For example, $\mathbf{V} \prec \mathbf{P}$ implies that we prefer the paths $vp \xrightarrow{\mathbf{V}} Vp \xrightarrow{\mathbf{P}} VP$ over the paths $vp \xrightarrow{\mathbf{P}} vP \xrightarrow{\mathbf{V}} VP$. In Table 3.2 we adapt those desiderata into specific subsets of states that should be preferred or avoided if the criteria is to be met.

A partial order over possible histories Given the desired preferences over orderings of transitions ($\mathbb{D}$ in (3.6)), we can construct a partial ordering over all possible histories $\mathcal{H}$, as defined in (3.8). This partial order requires a formal definition of which desiderata are met by a given history, provided by (3.7).

$$\begin{aligned} \mathbb{D}^h \stackrel{\text{def}}{=} \{d \in \mathbb{D} \text{ such that } d \text{ is true for } h\}, \text{ for } h \in \mathcal{H} \\ \text{where } d(\sigma_1, \sigma_2, \prec) \text{ is true for } h \text{ if and only if:} \\ \exists \sigma_i, \sigma_j \in h \text{ such that } \sigma_i = \sigma_1 \text{ and } \sigma_j = \sigma_2 \text{ and } h \text{ satisfies the relation } d(\sigma_i, \sigma_j, \prec) \end{aligned} \tag{3.7}$$

$$(\mathcal{H}, \leq_H) \stackrel{\text{def}}{=} \forall h_a, h_b \in \mathcal{H} \text{ it is the case that } h_b \leq_H h_a \text{ if and only if } \mathbb{D}^{h_b} \subseteq \mathbb{D}^{h_a} \tag{3.8}$$

A visualization of the resulting partially ordered set, or poset, $(\mathcal{H}, \leq_H)$ is shown as a Hasse Diagram in Figure 3.1. Hasse Diagrams represent the transitive reduction of a poset. Each node in the diagram represents an individual history h_a from Table 3.1; labels correspond to the index of the table. Figure 3.1 follows (3.8), in that h_a is higher in the order than h_b when h_a contains all the desiderata from h_b and at least one more. Histories that do not share a path are incomparable (formally, two histories incomparable if both $\mathbb{D}^{h_a} \not\supset \mathbb{D}^{h_b}$ and $\mathbb{D}^{h_a} \not\subset \mathbb{D}^{h_b}$). The diagram flows from least desirable histories at the bottom to most desirable at the top. This model satisfies **RQ1**; §4 and §5 will demonstrate that the model is amenable to analysis and §6.2.2 will lay out the criteria for extending it to cover MPCVD.

The poset $(\mathcal{H}, \leq_H)$, has as its upper bound

$$h_{69} = (\mathbf{V}, \mathbf{F}, \mathbf{D}, \mathbf{P}, \mathbf{X}, \mathbf{A})$$

while its lower bound is

$$h_0 = (\mathbf{A}, \mathbf{X}, \mathbf{P}, \mathbf{V}, \mathbf{F}, \mathbf{D}).$$

Thus far, we have made no assertions about the relative desirability of any two desiderata (that is, $d_i, d_j \in \mathbb{D}$ where $i \neq j$). In the next section we will expand the model to include a partial order over our desiderata, but for now it is sufficient to note that any simple ordering

Table 3.3: Ordered pairs of events where $row \prec col$ (Key: - = impossible, r = required, d = desired, u = undesired)

	V	**F**	**D**	**P**	**X**	**A**
V	-	r	r	d	d	d
F	-	-	r	d	d	d
D	-	-	-	d	d	d
P	u	u	u	-	d	d
X	u	u	u	u	-	d
A	u	u	u	u	u	-

over $\mathbb{D}$ would remain compatible with the partial order given in (3.8). In fact, a total order on $\mathbb{D}$ would create a linear extension of the poset defined here, whereas a partial order on $\mathbb{D}$ would result in a more constrained poset of which this poset would be a subset.

3.3 A Random Walk through CVD States

To begin to differentiate skill from chance in the next few sections, we need a model of what the CVD world would look like without any skill. We cannot derive this model by observation. Even when CVD was first practiced in the 1980s, some people may have had social, technical, or organizational skills that transferred to better CVD. We follow the principle of indifference, as stated in [23]:

> **Principle of Indifference:** Let $X = \{x_1, x_2, ..., x_n\}$ be a partition of the set W of possible worlds into n mutually exclusive and jointly exhaustive possibilities. In the absence of any relevant evidence pertaining to which cell of the partition is the true one, a rational agent should assign an equal initial credence of n to each cell.

While the principle of indifference is rather strong, it is inherently difficult to reason about absolutely skill-less CVD when the work of CVD is, by its nature, a skilled job. Given the set of states and allowable transitions between them, we can apply the principle of indifference to define a baseline against which measurement can be meaningful.

Estimating State Transition Probabilities Our assumption is that *transitions* are equally probable, not that *states* or *histories* are. The events $\sigma \in \Sigma$ trigger state transitions according to δ and the histories $h \in \mathcal{H}$ are paths (traces) through the states. This meets the definition above because each $\sigma \in \Sigma$ is unique (mutually exclusive) and δ defines an exhaustive set of valid σ at each state $q \in \mathcal{Q}$. For example, because (3.3) requires $\mathbf{V} \prec \mathbf{F}$ and $\mathbf{F} \prec \mathbf{D}$, only four of the six events in Σ are possible at the beginning of each history at $q_0 = vfdpxa$: $\{\mathbf{V}, \mathbf{P}, \mathbf{X}, \mathbf{A}\}$. Since the principle of indifference assigns each possible transition event as equally probable in this model of unskilled CVD, we assign an initial probability of 0.25 to each possible event.

$$p(\mathbf{V}|q_0) = p(\mathbf{P}|q_0) = p(\mathbf{X}|q_0) = p(\mathbf{A}|q_0) = 0.25$$
$$p(\mathbf{F}|q_0) = p(\mathbf{D}|q_0) = 0$$

From there, we see that the other rules dictate possible transitions from each subsequent state. For example, (3.4) says that any h starting with $(\mathbf{P})$ must start with $(\mathbf{P}, \mathbf{V})$. And (3.5) requires any h starting with $(\mathbf{X})$ must proceed through $(\mathbf{X}, \mathbf{P})$ and again (3.4) gets us to $(\mathbf{X}, \mathbf{P}, \mathbf{V})$. Therefore, we expect histories starting with $(\mathbf{P}, \mathbf{V})$ or $(\mathbf{X}, \mathbf{P}, \mathbf{V})$ to occur with frequency 0.25 as well. We can use these transition probabilities to estimate a neutral baseline expectation of which states would be common if we weren't doing anything special to coordinate vulnerability disclosures. Specifically, for each state we set the transition probability

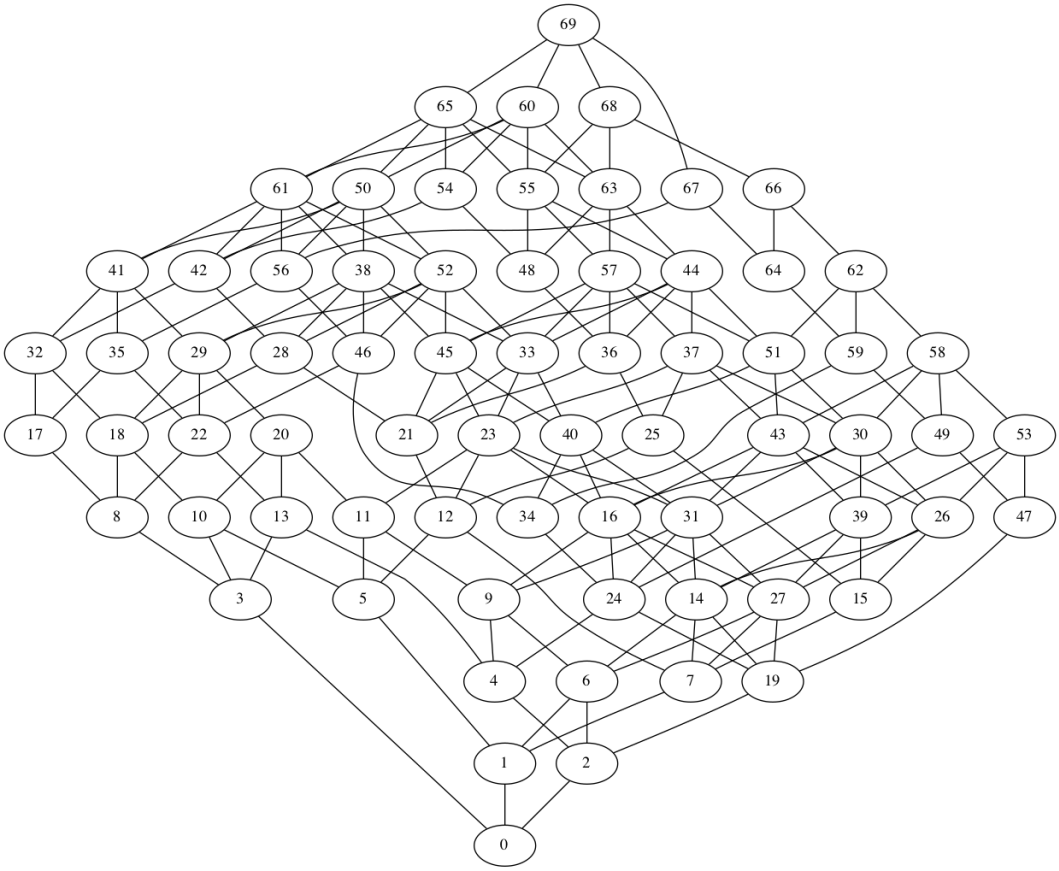


Figure 3.1: The Lattice of Possible CVD Histories: A Hasse Diagram of the partial ordering $(\mathcal{H}, \leq_H)$ *of* $h_a \in \mathcal{H}$ *given* $\mathbb{D}$ *as defined in* (3.8)*. The diagram flows from least desirable histories at the bottom to most desirable at the top. Histories that do not share a path are incomparable. Labels indicate the index (row number)* a *of* h_a *in Table 3.1.*

Table 3.4: Sparse state transition matrix and state PageRank assuming equiprobable transitions in a random walk over $\mathcal{S}$ as shown Figure 2.4.)

Start State	Next State(s)	$p(transition)$	PageRank
$vfdpxa$	$vfdpxA$, $vfdpXa$, $vfdPxa$, $Vfdpxa$	0.250	0.123
$vfdpxA$	$vfdpXA$, $vfdPxA$, $VfdpxA$	0.333	0.031
$vfdpXa$	$vfdPXa$	1.000	0.031
$vfdpXA$	$vfdPXA$	1.000	0.013
$vfdPxa$	$VfdPxa$	1.000	0.031
$vfdPxA$	$VfdPxA$	1.000	0.013
$vfdPXa$	$VfdPXa$	1.000	0.031
$vfdPXA$	$VfdPXA$	1.000	0.016
$Vfdpxa$	$VfdpxA$, $VfdpXa$, $VfdPxa$, $VFdpxa$	0.250	0.031
$VfdpxA$	$VfdpXA$, $VfdPxA$, $VFdpxA$	0.333	0.020
$VfdpXa$	$VfdPXa$	1.000	0.011
$VfdpXA$	$VfdPXA$	1.000	0.010
$VfdPxa$	$VfdPxA$, $VfdPXa$, $VFdPxa$	0.333	0.037
$VfdPxA$	$VfdPXA$, $VFdPxA$	0.500	0.032
$VfdPXa$	$VfdPXA$, $VFdPXa$	0.500	0.051
$VfdPXA$	$VFdPXA$	1.000	0.063
$VFdpxa$	$VFdpxA$, $VFdpXa$, $VFdPxa$, $VFDpxa$	0.250	0.011
$VFdpxA$	$VFdpXA$, $VFdPxA$, $VFDpxA$	0.333	0.013
$VFdpXa$	$VFdPXa$	1.000	0.007
$VFdpXA$	$VFdPXA$	1.000	0.008
$VFdPxa$	$VFdPxA$, $VFdPXa$, $VFDPxa$	0.333	0.018
$VFdPxA$	$VFdPXA$, $VFDPxA$	0.500	0.027
$VFdPXa$	$VFdPXA$, $VFDPXa$	0.500	0.037
$VFdPXA$	$VFDPXA$	1.000	0.092
$VFDpxa$	$VFDpxA$, $VFDpXa$, $VFDPxa$	0.333	0.007
$VFDpxA$	$VFDpXA$, $VFDPxA$	0.500	0.010
$VFDpXa$	$VFDPXa$	1.000	0.007
$VFDpXA$	$VFDPXA$	1.000	0.009
$VFDPxa$	$VFDPxA$, $VFDPXa$	0.500	0.012
$VFDPxA$	$VFDPXA$	1.000	0.026
$VFDPXa$	$VFDPXA$	1.000	0.031
$VFDPXA$	$\emptyset$	0.000	0.139

to any other state proportional to the inverse of the outdegree of the state, as shown in the $p(transition)$ column of Table 3.4. Real world data is unlikely to ever reflect such a sad state of affairs (because CVD *is* happening after all).

Using PageRank to Estimate Baseline State Probabilities We use the PageRank algorithm to provide state probability estimates. The PageRank algorithm provides a probability estimate for each state based on a Markov random walk of the directed graph of states [46]. PageRank assumes each available transition is equally probable, consistent with our model. To avoid becoming stuck in dead ends, PageRank adds a *teleport* feature by which walks can, with a small probability, randomly jump to another node in the graph.

Before proceeding, we need to make a small modification of our state digraph. Without modification, the PageRank algorithm will tend to be biased toward later states because the only way to reach the earlier states is for the algorithm to teleport there. Teleportation chooses states uniformly, so for example there is only a 1/32 chance of our actual start state ($q_0 = vfdpxa$) ever being chosen. Therefore, to ensure that the early states in our process are fairly represented we add a single link from $VFDPXA$ to $vfdpxa$, representing a model reset whenever the end state is reached. This modification allows PageRank traversals to wrap

around naturally and reach the early states in the random walk process without needing to rely on teleportation. With our modification in place, we are ready to compute the PageRank of each node in the graph. Results are shown in Table 3.4

4 Reasoning over Possible Histories

Our goal in this section is to formulate a way to rank our undifferentiated desiderata $\mathbb{D}$ from §3.2 in order to develop the concept of CVD skill and its measurement in §5. This will provide a baseline expectation about events (**RQ2**).

4.1 History Frequency Analysis

As described in §3.3, we apply the principle of indifference to the available transition events σ_{i+1} at each state q for each of the possible histories to compute the expected frequency of each history, which we denote as f_h. The frequency of a history f_h is the cumulative product of the probability p of each event σ_i in the history h. We are only concerned with histories that meet our sequence constraints, namely $h \in \mathcal{H}$.

$$f_h = \prod_{i=0}^{5} p(\sigma_{i+1}|h_i) \tag{4.1}$$

Table 3.1 displays the value of f_h for each history. Having an expected frequency (f_h) for each history h will allow us to examine how often we might expect our desiderata $d \in \mathbb{D}$ to occur across $\mathcal{H}$.

Choosing uniformly over event transitions is more useful than treating the six-element histories as uniformly distributed. For example, $\mathbf{P} \prec \mathbf{A}$ in 59% of valid histories, but when histories are weighted by the assumption of uniform state transitions $\mathbf{P} \prec \mathbf{A}$ is expected to occur in 67% of the time. These differences arise due to the dependencies between some states. Since CVD practice is comprised of a sequence of events, each informed by the last, our uniform distribution over events is more likely a useful baseline than a uniform distribution over histories.

4.2 Event Order Frequency Analysis

Each of the event pair orderings in Table 3.3 can be treated as a Boolean condition that either holds or does not hold in any given history. In §4.1 we described how to compute the expected frequency of each history (f_h) given the presumption of indifference to possible events at each step. We can use f_h as a weighting factor to compute the expected frequency of event orderings ($\sigma_i \prec \sigma_j$) across all possible histories H. Equations 4.2 and 4.3 define the frequency of an ordering $f_{\sigma_i \prec \sigma_j}$ as the sum over all histories in which the ordering occurs ($H^{\sigma_i \prec \sigma_j}$) of the frequency of each such history (f_h) as shown in Table 3.1.

$$H^{\sigma_i \prec \sigma_j} \stackrel{\text{def}}{=} \{h \in H \text{ where } \sigma_i \prec \sigma_j \text{ is true for } h \text{ and } i \neq j\} \tag{4.2}$$

$$f_{\sigma_i \prec \sigma_j} \stackrel{\text{def}}{=} \sum_{h \in H^{\sigma_i \prec \sigma_j}} f_h \tag{4.3}$$

Table 4.1 displays the results of this calculation. Required event orderings have an expected frequency of 1, while impossible orderings have an expected frequency of 0. As defined in §3.2, each desiderata $d \in \mathbb{D}$ is specified as an event ordering of the form $\sigma_i \prec \sigma_j$. We use f_d to denote the expected frequency of a given desiderata $d \in \mathbb{D}$. The values for the relevant f_d appear in the upper right of Table 4.1. Some event orderings have higher expected frequencies than

Table 4.1: Expected Frequency of $row \prec col$ when events are chosen uniformly from possible transitions in each state

	V	**F**	**D**	**P**	**X**	**A**
V	0	1	1	0.333	0.667	0.750
F	0	0	1	0.111	0.333	0.375
D	0	0	0	0.037	0.167	0.187
P	0.667	0.889	0.963	0	0.500	0.667
X	0.333	0.667	0.833	0.500	0	0.500
A	0.250	0.625	0.812	0.333	0.500	0

others. For example, vendor awareness precedes attacks in 3 out of 4 histories in a uniform distribution of event transitions ($f_{\mathbf{V} \prec \mathbf{A}} = 0.75$), whereas fix deployed prior to public awareness holds in less than 1 out of 25 ($f_{\mathbf{D} \prec \mathbf{P}} = 0.037$) histories generated by a uniform distribution over event transitions.

4.3 A Partial Order on Desiderata

Any observations of phenomena in which we measure the performance of human actors can attribute some portion of the outcome to skill and some portion to chance [36, 19]. It is reasonable to wonder whether good outcomes in CVD are the result of luck or skill. How can we tell the difference?

We begin with a simple model in which outcomes are a combination of luck and skill.

$$o_{observed} = o_{luck} + o_{skill}$$

In other words, outcomes due to skill are what remains when you subtract the outcomes due to luck from the outcomes you observe. In this model, we treat *luck* as a random component: the contribution of chance. In a world where neither attackers nor defenders held any advantage and events were chosen uniformly from Σ whenever they were possible, we would expect to see the preferred orderings occur with probability equivalent to their frequency f_d as shown in Table 4.1.

Skill, on the other hand, accounts for the outcomes once luck has been accounted for. So the more likely an outcome is due to luck, the less skill we can infer when it is observed. As an example, from Table 4.1 we see that fix deployed before the vulnerability is public is the rarest of our desiderata with $f_{\mathbf{D} \prec \mathbf{P}} = 0.037$, and thus exhibits the most skill when observed. On the other hand, vendor awareness before attacks is expected to be a common occurrence with $f_{\mathbf{V} \prec \mathbf{A}} = 0.75$.

We can therefore use the set of f_d to construct a partial order over $\mathbb{D}$ in which we prefer desiderata d which are more rare (and therefore imply more skill when observed) over those that are more common. We create the partial order on $\mathbb{D}$ as follows: for any pair $d_1, d_2 \in \mathbb{D}$, we say that d_2 exhibits less skill than d_1 if d_2 occurs more frequently in $\mathcal{H}$ than d_1.

$$(\mathbb{D}, \leq_{\mathbb{D}}) \stackrel{\text{def}}{=} d_2 \leq_{\mathbb{D}} d_1 \iff f_{d_2} \stackrel{\mathbb{R}}{\geq} f_{d_1} \tag{4.4}$$

Note that the inequalities on the left and right sides of (4.4) are flipped because skill is inversely proportional to luck. Also, while $\leq_{\mathbb{D}}$ on the left side of (4.4) defines a preorder over the poset $\mathcal{H}$, the $\stackrel{\mathbb{R}}{\geq}$ is the usual ordering over the set of real numbers. The result is a partial order $(\mathbb{D}, \leq_{\mathbb{D}})$ because a few d have the same f_d ($f_{\mathbf{F} \prec \mathbf{X}} = f_{\mathbf{V} \prec \mathbf{P}} = 0.333$ for example). The full Hasse Diagram for the partial order $(\mathbb{D}, \leq_{\mathbb{D}})$ is shown in Figure 4.1.

4.4 Ordering Possible Histories by Skill

Next we develop a new partial order on $\mathcal{H}$ given the partial order $(\mathbb{D}, \leq_{\mathbb{D}})$ just described. We observe that $\mathbb{D}^h$ acts as a Boolean vector of desiderata met by a given h. Since $0 \leq f_d \leq 1$, simply taking its inverse could in the general case lead to some large values for rare events. For convenience, we use $-log(f_d)$ as our proxy for skill. For example, if a desideratum were found to occur in every case (indicating no skill required), $f_d = 1$ and therefore $-log(1) = 0$. On the other hand, for increasingly rare desiderata, our skill metric rises: $\lim_{f_d \to 0} -log(f_d) = +\infty$.

Taking the dot product of $\mathbb{D}^h$ with the set of $-log(f_d)$ values for each $d \in \mathbb{D}$ represented as a vector, we arrive at a single value representing the skill exhibited for each history h. Careful readers may note that this value is equivalent to the Term Frequency—Inverse Document Frequency (TF-IDF) score for a search for the "skill terms" represented by $\mathbb{D}$ across the corpus of possible histories $\mathcal{H}$.

We have now computed a skill value for every $h \in \mathcal{H}$, which allows us to sort $\mathcal{H}$ and assign a rank to each history h contained therein. The rank is shown in Table 3.1. Rank values start at 1 for least skill up to a maximum of 62 for most skill. Owing to the partial order $(\mathbb{D}, \leq_{\mathbb{D}})$, some h have the same computed skill values, and these are given the same rank.

The ranks for $h \in \mathcal{H}$ lead directly to a new poset $(\mathcal{H}, \leq_{\mathbb{D}})$. It is an extension of and fully compatible with $(\mathcal{H}, \leq_H)$ as developed in §3.2.

The resulting Hasse Diagram would be too large to reproduce here. Instead, we include the resulting rank for each h as a column in Table 3.1. In the table, rank is ordered from least desirable and skillful histories to most. Histories having identical rank are incomparable to each other within the poset. The refined poset $(\mathcal{H}, \leq_{\mathbb{D}})$ is much closer to a total order on $\mathcal{H}$, as indicated by the relatively few histories having duplicate ranks.

The remaining incomparable histories are the direct result of the incomparable d in $(\mathbb{D}, \leq_{\mathbb{D}})$, corresponding to the branches in Figure 4.1. Achieving a total order on $\mathbb{D}$ would require determining a preference for one of each of the following:

- that fix ready precede exploit publication ($\mathbf{F} \prec \mathbf{X}$) or that vendor awareness precede public awareness ($\mathbf{V} \prec \mathbf{P}$)
- that public awareness precede exploit publication ($\mathbf{P} \prec \mathbf{X}$) or that exploit publication precede attacks ($\mathbf{X} \prec \mathbf{A}$)
- that public awareness precede attacks ($\mathbf{P} \prec \mathbf{A}$) or vendor awareness precede exploit publication ($\mathbf{V} \prec \mathbf{X}$)

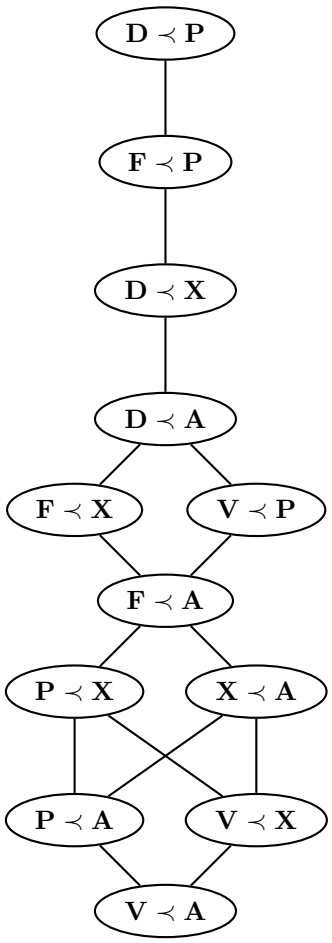


Figure 4.1: Hasse Diagram of the partial order $(\mathbb{D}, \leq_{\mathbb{D}})$ defined in Eq. 4.4 where the rarity of each d as shown in Table 4.1 is taken to reflect skill. Nodes at the top of the diagram reflect the most skill.

Recognizing that readers may have diverse opinions on all three items, we leave further consideration of the answers to these as future work.

This is just one example of how poset refinements might be used to order $\mathcal{H}$. Different posets

on $\mathbb{D}$ would lead to different posets on $\mathcal{H}$. For example, one might construct a different poset if certain d were considered to have much higher financial value when achieved than others.

5 Discriminating Skill and Luck in Observations

This section defines a method for measuring skillful behavior in CVD, which we will need to answer **RQ3** about measuring and evaluating CVD "in the wild." The measurement method makes use of all the modeling tools and baselines established thus far: a comprehensive set of possible histories $\mathcal{H}$, a partial order over them in terms of the presence of desired event precedence $(\mathcal{H}, \leq_{\mathbb{D}})$, and the *a priori* expected frequency of each desiderata $d \in \mathbb{D}$.

If we expected to be able to observe all events in all CVD cases, we could be assured of having complete histories and could be done here. But the real world is messy. Not all events $\mathbf{e} \in \mathcal{E}$ are always observable. We need to develop a way to make sense of what we *can* observe, regardless of whether we are ever able to capture complete histories. Continuing towards our goal of measuring efficacy, we return to considering the balance between skill and luck in determining our observed outcomes.

5.1 A Measure of Skill in CVD

There are many reasons why we might expect our observations to differ from the expected frequencies we established in §4. Adversaries might be rare, or conversely very well equipped. Vendors might be very good at releasing fixes faster than adversaries can discover vulnerabilities and develop exploits for them. System owners might be diligent at applying patches. (We did say *might*, didn't we?) Regardless, for now we will lump all of those possible explanations into a single attribute called "skill."

In a world of pure skill, one would expect that a player could achieve all 12 desiderata $d \in \mathbb{D}$ consistently. That is, a maximally skillful player could consistently achieve the specific ordering $h = (\mathbf{V}, \mathbf{F}, \mathbf{D}, \mathbf{P}, \mathbf{X}, \mathbf{A})$ with probability $p_{skill} = 1$.

Thus, we construct the following model: for each of our preferred orderings $d \in \mathbb{D}$, we model their occurrence due to luck using the binomial distribution with parameter $p_{luck} = f_d$ taken from Table 4.1.

Recall that the mean of a binomial distribution is simply the probability of success p, and that the mean of a weighted mixture of two binomial distributions is simply the weighted mixture of the individual means. Therefore our model adds a parameter α_d to represent the weighting between our success rates arising from skill p_{skill} and luck p_{luck}. Because there are 12 desiderata $d \in \mathbb{D}$, each d will have its own observations and corresponding value for α_d for each history h_a.

$$f_d^{obs} = \alpha_d \cdot p_{skill} + (1 - \alpha_d) \cdot p_{luck} \tag{5.1}$$

Where f_d^{obs} is the observed frequency of successes for desiderata d. Because $p_{skill} = 1$, one of those binomial distributions is degenerate. Substituting $p_{skill} = 1$, $p_{luck} = f_d$ and solving Eq. 5.1 for α, we get

$$\alpha_d \stackrel{\text{def}}{=} \frac{f_d^{obs} - f_d}{1 - f_d} \tag{5.2}$$

The value of α_d therefore gives us a measure of the observed skill normalized against the background success rate provided by luck f_d. We denote the set of α_d values for a given history as $\alpha_{\mathbb{D}}$. When we refer to the α_d coefficient for a specific d we will use the specific ordering as the subscript, for example: $\alpha_{\mathbf{F} \prec \mathbf{P}}$.

$$\alpha_{\mathbb{D}} \stackrel{\text{def}}{=} \{\alpha_d : d \in \mathbb{D}\} \tag{5.3}$$

The concept embodied by f_d is founded on the idea that if attackers and defenders are in a state of equilibrium, the frequency of observed outcomes (i.e., how often each desiderata d and history h actually occurs) will appear consistent with those predicted by chance. So another way of interpreting α_d is as a measure of the degree to which a set of observed histories is out of equilibrium.

The following are a few comments on how α_d behaves. Note that $\alpha_d < 0$ when $0 \leq f_d^{obs} < f_d$ and $0 \leq \alpha_d \leq 1$ when $f_d \leq f_d^{obs} \leq 1$. The implication is that a negative value for α_d indicates that our observed outcomes are actually *worse* than those predicted by pure luck. In other words, we can only infer positive skill when the observations are higher ($f_d^{obs} > f_d$). That makes intuitive sense: if you are likely to win purely by chance, then you have to attribute most of your wins to luck rather than skill. From Table 4.1, the largest value for any $d \in \mathbb{D}$ is $f_{\mathbf{V} \prec \mathbf{A}} = 0.75$, implying that even if a vendor knows about 7 out of 10 vulnerabilities before attacks occur ($f_{\mathbf{V} \prec \mathbf{A}}^{obs} = 0.7$), they are still not doing better than random.

On the other hand, when f_d is small it is easier to infer skill should we observe anything better than f_d. However, it takes larger increments of observations f_d^{obs} to infer growth in skill when f_d is small than when it is large. The smallest f_d we see in Table 4.1 is $f_{\mathbf{D} \prec \mathbf{P}} = 0.037$.

Inherent to the binomial distribution is the expectation that the variance of results is lower for both extremes (as p approaches either 0 or 1) and highest at $p = 0.5$. Therefore we should generally be less certain of our observations when they fall in the middle of the distribution. We address uncertainty further in §5.1.2.

5.1.1 Computing α_d from Observations

Although Eq. (5.2) develops a skill metric from observed frequencies, our observations will in fact be based on counts. Observations consist of some number of successes S_d^{obs} out of some number of trials T, i.e.,

$$f_d^{obs} = \frac{S_d^{obs}}{T} \tag{5.4}$$

We likewise revisit our interpretation of f_d.

$$f_d = \frac{S_d^{luck}}{T} \tag{5.5}$$

where S_d^{luck} is the number of successes at d we would expect due to luck in T trials.

Substituting (5.4) and (5.5) into (5.2), and recalling that $p_{skill} = 1$ because a maximally skillful player succeeds in T out of T trials, we get

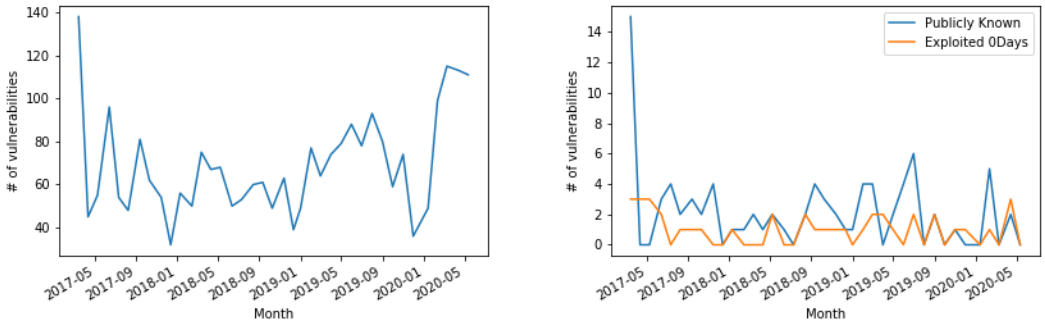
$$\alpha_d = \frac{\frac{S_d^{obs}}{T} - \frac{S_d^{luck}}{T}}{\frac{T}{T} - \frac{S_d^{l}}{T}} \tag{5.6}$$

Rearranging (5.5) to $S_d^{luck} = f_d T$, substituting into (5.6), and simplifying, we arrive at:

$$\alpha_d = \frac{S_d^{obs} - f_d T}{(1 - f_d)T} \tag{5.7}$$

Hence for any of our desiderata $\mathbb{D}$ we can compute α_d given S_d^{obs} observed successes out of T trials in light of f_d taken from Table 4.1.

Before we address the data analysis we take a moment to discuss uncertainty.



(a) Vulnerabilities Patched

(b) Vulnerabilities known or attacked prior to patch availability

Figure 5.1: Publicly Disclosed Microsoft Vulnerabilities 2017-2020

5.1.2 Calculating Measurement Error

We have already described the basis of our f_d^{obs} model in the binomial distribution. While we could just estimate the error in our observations using the binomial's variance $np(1-p)$, because of boundary conditions at 0 and 1 we should not assume symmetric error. An extensive discussion of uncertainty in the binomial distribution is given in [12].

However, for our purpose the Beta distribution lends itself to this problem nicely. The Beta distribution is specified by two parameters (a, b). It is common to interpret a as the number of successes and b as the number of failures in a set of observations of Bernoulli trials to estimate the mean of the binomial distribution from which the observations are drawn. For any given mean, the width of the Beta distribution narrows as the total number of trials increases.

We use this interpretation to estimate a 95% credible interval for f_d^{obs} using a Beta distribution with parameters $a = S_d^{obs}$ as successes and $b = T - S_d^{obs}$ representing the number of failures using the `scipy.stats.beta.interval` function in Python. This gives us an upper and lower estimate for f_d^{obs}, which we multiply by T to get upper and lower estimates of S_d^{obs} as in (5.4).

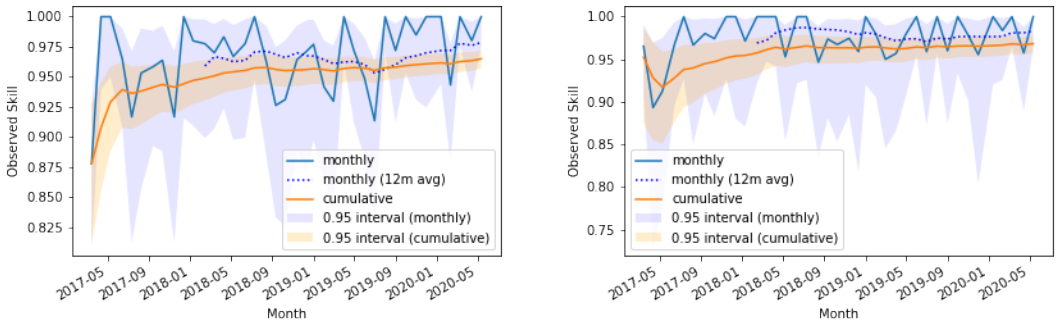
5.2 Observing CVD in the Wild

As a proof of concept, we apply the model to two data sets: Microsoft's security updates from 2017 through early 2020 in §5.2.1, and commodity public exploits from 2015-2019 in §5.2.2.

5.2.1 Microsoft 2017-2020

We are now ready to proceed with our data analysis. First, we examine Microsoft's monthly security updates for the period between March 2017 and May 2020, as curated by the Zero Day Initiative blog[1]. Figure 5.1a shows monthly totals for all vulnerabilities while 5.1b has monthly observations of $\mathbf{P} \prec \mathbf{F}$ and $\mathbf{A} \prec \mathbf{F}$. This data set allowed us to compute the monthly counts for two of our desiderata, $\mathbf{F} \prec \mathbf{P}$ and $\mathbf{F} \prec \mathbf{A}$.

[1] `https://www.zerodayinitiative.com/blog`. The ZDI blog posts were more directly useful than the monthly Microsoft security updates because ZDI had already condensed the counts of how many vulnerabilities were known ($\mathbf{P}$) or exploited ($\mathbf{A}$) prior to their fix readiness $\mathbf{F}$. Retrieving this data from Microsoft's published vulnerability information requires collecting it from all the individual vulnerabilities patched each month. We are grateful to ZDI for providing this summary and saving us the effort.



(a) $\alpha_{\mathbf{F} \prec \mathbf{P}}$ *(b)* $\alpha_{\mathbf{F} \prec \mathbf{A}}$

Figure 5.2: Selected Skill Measurement for Publicly Disclosed Microsoft Vulnerabilities 2017-2020

Observations of $\mathbf{F} \prec \mathbf{P}$*:* In total, Microsoft issued patches for 2,694 vulnerabilities; 2,610 (0.97) of them met the fix-ready-before-public-awareness ($\mathbf{F} \prec \mathbf{P}$) objective. The mean monthly $\alpha_{\mathbf{F} \prec \mathbf{P}} = 0.967$, with a range of [0.878, 1.0]. We can also use the cumulative data to estimate an overall skill level for the observation period, which gives us a bit more precision on $\alpha_{\mathbf{F} \prec \mathbf{P}} = 0.969$ with the 0.95 interval of [0.962, 0.975]. Figure 5.2a shows the trend for both the monthly observations and the cumulative estimate of $\alpha_{\mathbf{F} \prec \mathbf{P}}$.

Observations of $\mathbf{F} \prec \mathbf{A}$*:* Meanwhile, 2,655 (0.99) vulnerabilities met the fix-ready-before-attacks-observed ($\mathbf{F} \prec \mathbf{A}$) criteria. Thus we compute a mean monthly $\alpha_{\mathbf{F} \prec \mathbf{A}} = 0.976$ with range [0.893, 1.0]. The cumulative estimate yields $\alpha_{\mathbf{F} \prec \mathbf{A}} = 0.986$ with an interval of [0.980, 0.989]. The trend for both is shown in Figure 5.2b.

Inferring Histories from Observations: Another possible application of our model is to estimate unobserved α_d based on the cumulative observations of both $f^{obs}_{\mathbf{F} \prec \mathbf{P}}$ and $f^{obs}_{\mathbf{F} \prec \mathbf{A}}$ above. Here we estimate the frequency f_d of the other $d \in \mathbb{D}$ for this period. Our procedure is as follows:

1. For 10000 rounds, draw an f^{est}_d for both $\mathbf{F} \prec \mathbf{P}$ and $\mathbf{F} \prec \mathbf{A}$ from the Beta distribution with parameters $a = S^{obs}_d$ and $b = T - S^{obs}_d$ where S^{obs}_d is 2,610 or 2,655, respectively, and T is 2,694.
2. Assign each $h \in \mathcal{H}$ a weight according to standard joint probability based whether it meets both, either, or neither $A = \mathbf{F} \prec \mathbf{P}$ and $B = \mathbf{F} \prec \mathbf{A}$, respectively.

$$w_h = \begin{cases} p_{AB} = f_A * f_B \text{ if } A \text{ and } B \\ p_{Ab} = f_A * f_b \text{ if } A \text{ and } \neg B \\ p_{aB} = f_a * f_B \text{ if } \neg A \text{ and } B \\ p_{ab} = f_a * f_b \text{ if } \neg A \text{ and } \neg B \end{cases} \tag{5.8}$$

   where $f_a = 1 - f_A$ and $f_b = 1 - f_B$
3. Draw a weighted sample (with replacement) of size $N = 2,694$ from $\mathcal{H}$ according to these weights.
4. Compute the sample frequency $f^{sample}_d = S^{sample}_d / N$ for each $d \in \mathbb{D}$, and record the median rank of all histories h in the sample.

5. Compute the estimated frequency as the mean of the sample frequencies, namely $f_d^{est} = \langle f_d^{sample} \rangle$, for each $d \in \mathbb{D}$.
6. Compute α_d from f_d^{est} for each $d \in \mathbb{D}$.

As one might expect given the causal requirement that vendor awareness precedes fix availability, the estimated values of α_d are quite high $(0.96 - 0.99)$ for our desiderata involving either $\mathbf{V}$ or $\mathbf{F}$. We also estimate that α_d is positive—indicating that we are observing skill over and above mere luck—for all d except $\mathbf{P} \prec \mathbf{A}$ and $\mathbf{X} \prec \mathbf{A}$ which are slightly negative. The results are shown in Figure 5.3. The most common sample median history rank across all runs is 53, with all sample median history ranks falling between 51-55. The median rank of possible histories weighted according to the assumption of equiprobable transitions is 11. We take this as evidence that the observations are indicative of skill.

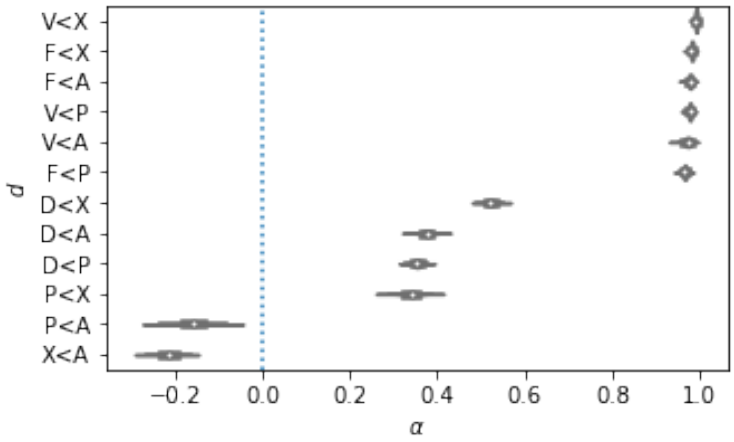


Figure 5.3: Simulated skill α_d for Microsoft 2017-2020 based on observations of $\mathbf{F} \prec \mathbf{P}$ and $\mathbf{F} \prec \mathbf{A}$ over the period.

5.2.2 Commodity Exploits 2015-2019

Next, we examine the overall trend in $\mathbf{P} \prec \mathbf{X}$ for commodity exploits between 2015 and 2019. The data set is based on the National Vulnerability Database [40], in conjunction with the CERT Vulnerability Data Archive [15]. Between these two databases, a number of candidate dates are available to represent the date a vulnerability was made public. We use the minimum of these as the date for P.

To estimate the exploit availability ($\mathbf{X}$) date, we extracted the date a CVE ID appeared in the git logs for Metasploit [48] or Exploitdb [50]. When multiple dates were available for a CVE ID, we kept the earliest. Note that commodity exploit tools such as Metasploit and Exploitdb represent a non-random sample of the exploits available to adversaries. These observations should be taken as a lower bounds estimate of exploit availability, and therefore an upper bounds estimate of observed desiderata d and skill α_d.

During the time period from 2013-2019, the data set contains $N = 73,474$ vulnerabilities. Of these, 1,186 were observed to have public exploits ($\mathbf{X}$) prior to the earliest observed vulnerability disclosure date ($\mathbf{P}$), giving an overall success rate for $\mathbf{P} \prec \mathbf{X}$ of 0.984. The mean monthly $\alpha_{\mathbf{P} \prec \mathbf{X}}$ is 0.966 with a range of [0.873, 1.0]. The volatility of this measurement appears to be higher than that of the Microsoft data. The cumulative $\alpha_{\mathbf{P} \prec \mathbf{X}}$ comes in at 0.968 with an interval spanning [0.966, 0.970]. A chart of the trend is shown in Fig. 5.4.

To estimate unobserved α_d from the commodity exploit observations, we repeat the procedure

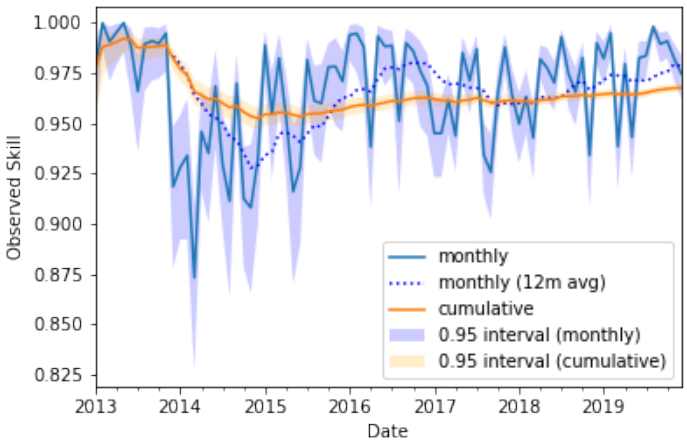


*Figure 5.4: $\alpha_{\mathbf{P} \prec \mathbf{X}}$ for all NVD vulnerabilities 2013-2019 (**X** observations based on Metasploit and ExploitDb)*

outlined in §5.2.1. This time, we use $N = 73,474$ and estimate f_d^{est} for $\mathbf{P} \prec \mathbf{X}$ with Beta parameters $a = 72,288$ and $b = 1186$. As above, we find evidence of skill in positive estimates of α_d for all desiderata except $\mathbf{P} \prec \mathbf{A}$ and $\mathbf{X} \prec \mathbf{A}$, which are negative. The most common sample median history rank in this estimate is 33 with a range of [32,33], which while lower than the median rank of 53 in the Microsoft estimate from §5.2.1, still beats the median rank of 11 assuming uniform event probabilities. The results are shown in Figure 5.5.

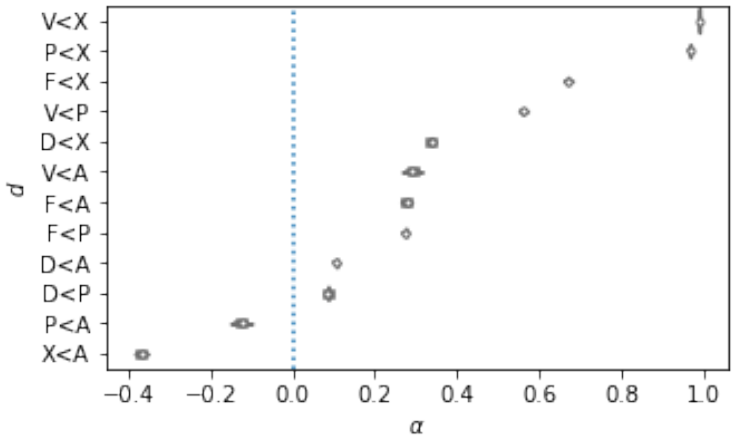


Figure 5.5: Simulated skill α_d for all NVD vulnerabilities 2013-2019 based on observations of $\mathbf{P} \prec \mathbf{X}$ over the period.

6 Discussion

The observational analysis in §5.2 supports an affirmative response to **RQ3**: vulnerability disclosure as currently practiced demonstrates skill. In both data sets examined, our estimated α_d is positive for most $d \in \mathbb{D}$. However, there is uncertainty in our estimates due to the application of the principle of indifference to unobserved data. This principle assumes a uniform distribution across event transitions in the absence of CVD, which is an assumption we cannot readily test. The spread of the estimates in Figures 5.3 and 5.5 represents the variance in our samples, not this assumption-based uncertainty. Our interpretation of α_d values near zero is therefore that they reflect an absence of evidence rather than evidence that skill is absent. While we cannot rule definitively on luck or low skill, values of $\alpha_d > 0.9$ should reliably indicate skillful defenders.

If, as seems plausible from the evidence, it turns out that further observations of h are significantly skewed toward the higher end of the poset $(\mathcal{H}, \leq_{\mathbb{D}})$, then it may be useful to empirically calibrate our metrics rather than using the *a priori* frequencies in Table 4.1 as our baseline. This analysis baseline would provide context on "more skillful than the average for some set of teams" rather than more skillful than blind luck. §6.1 discusses this topic, which should be viewed as an examination of what "reasonable" in **RQ2** should mean in the context of "reasonable baseline expectation."

§6.2 suggests how the model might be applied to establish benchmarks for CVD processes involving any number of participants, which closes the analysis of **RQ1** in relation to MPCVD. §6.3 surveys the stakeholders in CVD and how they might use our model; the stakeholders are vendors, system owners, the security research community, coordinators, and governments. In particular, we focus on how these stakeholders might respond to the affirmative answer to **RQ3** and a method to measure skill in a way more tailored to each stakeholder group. §6.4 discusses the potential for formalizing disclosure policy specifications using the model. §6.5 offers formal definitions of some common terms in vulnerability disclosure. We then proceed to address vulnerability response situation awareness in §6.6, with a brief note about the Vulnerability Equities Process (VEP) in relation to this model in §6.7. Finally, a set of state-based rules for CVD actions is given in §6.8.

6.1 CVD Benchmarks

As described above, in an ideal CVD situation, each observed history would achieve all 12 desiderata $\mathbb{D}$. Realistically, this is unlikely to happen. We can at least state that we would prefer that most cases reach fix ready before attacks ($\mathbf{F} \prec \mathbf{A}$). Per Table 4.1, even in a world without skill we would expect $\mathbf{F} \prec \mathbf{A}$ to hold in 73% of cases. This means that $\alpha_{\mathbf{F} \prec \mathbf{A}} < 0$ for anything less than a 0.73 success rate. In fact, we propose to generalize this for any $d \in \mathbb{D}$, such that α_d should be greater than some benchmark constant c_d:

$$\alpha_d \geq c_d \geq 0 \tag{6.1}$$

where c_d is a based on observations of α_d collected across some collection of CVD cases.

We propose as a starting point a naïve benchmark of $c_d = 0$. This is a low bar, as it only requires that CVD actually do better than possible events which are independent and identically distributed (i.i.d.) within each case. For example, given a history in which $(\mathbf{V}, \mathbf{F}, \mathbf{P})$ have already happened (i.e., state $q \in VFdPxa$), $\mathbf{D}$, $\mathbf{X}$, or $\mathbf{A}$ are equally likely to occur next.

The i.i.d. assumption may not be warranted. We anticipate that event ordering probabilities might be conditional on history: for example, exploit publication may be more likely when the vulnerability is public ($p(\mathbf{X}|q \in \mathcal{Q}_P) > p(\mathbf{X}|q \in \mathcal{Q}_p)$) or attacks may be more likely when an exploit is public ($p(\mathbf{A}|q \in \mathcal{Q}_X) > p(\mathbf{A}|q \in \mathcal{Q}_x)$). If the i.i.d. assumption fails to hold for transition events $\sigma \in \Sigma$, observed frequencies of $h \in \mathcal{H}$ could differ significantly from the rates predicted by the uniform probability assumption behind Table 4.1.

Some example suggestive observations are:

- There is reason to suspect that only a fraction of vulnerabilities ever reach the *exploit public* event **X**, and fewer still reach the *attack* event **A**. Recent work by the Cyentia Institute found that "5% of all CVEs are both observed within organizations AND known to be exploited" [1], which suggests that $f_{\mathbf{D} \prec \mathbf{A}} \approx 0.95$.
- Likewise, $\mathbf{D} \prec \mathbf{X}$ holds in 28 of 70 (0.4) h. However Cyentia found that "15.6% of all open vulnerabilities observed across organizational assets in our sample have known exploits" [1], which suggests that $f_{\mathbf{D} \prec \mathbf{X}} \approx 0.844$.

We might therefore expect to find many vulnerabilities remaining indefinitely in $VFDPxa$.

On their own these observations can equally well support the idea that we are broadly observing skill in vulnerability response, rather than that the world is biased from some other cause. However, we could choose a slightly different goal than differentiating skill and "blind luck" as represented by the i.i.d. assumption. One could aim to measure "more skillful than the average for some set of teams" rather than more skillful than blind luck.

If this were the "reasonable" baseline expectation (**RQ2**), the primary limitation is available observations. This model helps overcome this limitation because it provides a clear path toward collecting relevant observations. For example, by collecting dates for the six $\sigma \in \Sigma$ for a large sample of vulnerabilities, we can get better estimates of the relative frequency of each history h in the real world. It seems as though better data would serve more to improve benchmarks rather than change expectations of the role of chance.

As an applied example, if we take the first item in the above list as a broad observation of $f_{\mathbf{D} \prec \mathbf{A}} = 0.95$, we can plug into (5.2) to get a potential benchmark of $\alpha_{\mathbf{D} \prec \mathbf{A}} = 0.94$, which is considerably higher than the naïve generic benchmark $\alpha_d = 0$. It also implies that we should expect actual observations of histories $h \in \mathcal{H}$ to skew toward the 19 h in which $\mathbf{D} \prec \mathbf{A}$ nearly 20x as often as the 51 h in which $\mathbf{A} \prec \mathbf{D}$. Similarly, if we interpret the second item as a broad observation of $f_{\mathbf{D} \prec \mathbf{X}} = 0.844$, we can then compute a benchmark $\alpha_{\mathbf{D} \prec \mathbf{X}} = 0.81$, which is again a significant improvement over the naïve $\alpha_d = 0$ benchmark.

6.2 Multi-Party Coordinated Vulnerability Disclosure

Multi-Party Coordinated Vulnerability Disclosure (MPCVD) is the process of coordinating the creation, release, publication, and potentially the deployment of fixes for vulnerabilities across a number of vendors and their respective products. The need for MPCVD arises due to the inherent nature of the software supply chain [30]. A vulnerability that affects a low-level component (such as a library or operating system API) can require fixes from both the originating vendor and any vendor whose products incorporate the affected component. Alternatively, vulnerabilities are sometimes found in protocol specifications or other design-time issues where multiple vendors may have each implemented their own components based on a vulnerable design. §6.2.1 applies the state-based view of our model to MPCVD, while §6.2.2 addresses the topic from the possible history perspective.

6.2.1 State Tracking in MPCVD

Applying our state-based model to MPCVD requires a forking approach to the state tracking. At the time of discovery, the vulnerability is in state $vfdpxa$. Known only to its finder, the vulnerability can be described by that singular state.

As it becomes clear that the vulnerability affects multiple vendors' products, both finder/reporters and coordinators might begin to track the state of each individual vendor as a separate instance of the model. For example, if 3 vendors are known to be affected, but only 1 has been notified, the case might be considered to be in a superposition[1] of states $\{Vfdpxa, vfdpxa, vfdpxa\}$.

Each vendor, in turn, might then ascertain whether they are able to produce fixes for all their available products at once, or if those fixes will be staggered over time. In either case, the vendor might track the case as a superposition of states for each affected product dependent on its fix readiness status. For example, a vendor might have four products affected, with two in the fix-ready state and two fixes still in progress. Should they opt for transparency into their internal process, they might communicate their status as the superposition of $\{VFdpxa, VFdpxa, Vfdpxa, Vfdpxa\}$. Alternatively, they might choose to only share the lowest state across their products, which in this example would be $\{Vfdpxa\}$.

This implies a need to expand our notation. In the MPCVD case, we need to think of each state $q \in \mathcal{Q}$ as a set of states q_M:

$$q_M \stackrel{\text{def}}{=} \{q_1, q_2, \ldots, q_n\} \tag{6.2}$$

where q_i represents the state q for the ith affected vendor and/or product. For example, $\{Vfdpxa_1, vfdpxa_2\}$ would represent the state in which vendor 1 has been notified but vendor 2 has not.

State transitions across vendors need not be simultaneous. Very often, vendor notification occurs as new products and vendors are identified as affected in the course of analyzing a vulnerability report. So the individual events $\mathbf{V}_i$ in $\mathbf{V}_M$ (representing the status of all the vendor notifications) might be spread out over a period of time.

Some transitions are more readily synchronized than others. For example, if an MPCVD case is already underway, and information about the vulnerability appears in a public media report, we can say that $\mathbf{P}_M$ occurred simultaneously for all coordinating vendors.

Regardless, in the maximal case, each vendor-product pair is effectively behaving independently of all the others. Thus the maximum dimensionality of the MPCVD model for a case is

$$D_{max} = 5 * N_{vprod} \tag{6.3}$$

where N_{vprod} represents the number of vendor-product pairs.

This is of course undesirable, as it would result in a wide distribution of realized histories that more closely resemble the randomness assumptions outlined above than a skillful, coordinated effort. Further discussion of measuring MPCVD skill can be found in 6.2.2. For now, though, we posit that the goal of a good MPCVD process is to reduce the dimensionality of a given MPCVD case as much as is possible (i.e., to the 5 dimensions of a single vendor CVD case we have presented above). Experience shows that a full dimension reduction is unlikely in most cases, but that does not detract from the value of having the goal.

Vendors may be able to reduce their internal tracking dimensionality—which may be driven by things like component reuse across products or product lines—through in-house coordination of fix development processes. Within an individual vendor organization, Product Security

[1] Borrowing the terminology of quantum mechanics

Incident Response Teams (PSIRTs) are a common organizational structure to address this internal coordination process. The FIRST PSIRT Services Framework provides guidance regarding vendors' internal processes for coordinating vulnerability response [43]. Additional guidance can be found in ISO-IEC 30111 [32].

Regardless, the cross-vendor dimension is largely the result of component reuse across vendors, for example through the inclusion of third party libraries or Original Equipment Manufacturer (OEM) Software Development Kits (SDKs). Visibility of cross-vendor component reuse remains an unsolved problem, although efforts such as NTIA's SBOM [55] efforts are promising in this regard. Thus, dimensionality reduction can be achieved through both improved transparency of the software supply chain and the process of coordination toward synchronized state transitions, especially for **P**, if not for **F** and **D** as well.

As a result of the dimensionality problem, coordinators and other parties to an MPCVD case need to decide how to apply disclosure policy rules in cases where different products or vendors occupy different case states with potentially contradictory recommended actions. For example, when four out of five vendors involved in a case have reached $VFdpxa$ and are ready to publish, but the fifth is still in $Vfdpxa$. Essentially this can be expected to take the form of a weighting function that acts on a vector of case states, and outputs a set of recommended actions derived from those states.

One possible function would be to apply a simple voting heuristic such as waiting for a simple majority of vendors to reach a state before taking action as that state recommends. In our 4/5 $VFdpxa$ example, the coordinating parties would simply behave as if the case were in that state for all. Another could be to weight vendors and products by some other factor, such as size of the installed user base, or even based on a risk analysis of societal costs associated with potential actions. Here we acknowledge that our example is under-specified: does the fifth vendor (the one in $Vfdpxa$) represent a sizable fraction of the total user base? Or does it concentrate the highest risk use cases for the software? Challenges in efficiently assessing consistent answers to these questions are easy to imagine. The status quo for MPCVD appears consistent with defaulting to simple majority in the absence of additional information, with consideration given to the distribution of both users and risk on a case-by-case basis. At present, there is no clear consensus on such policies, although we hope that future work can use the model presented here to formalize the necessary analysis.

Integrating FIRST MPCVD Guidance FIRST has published MPCVD guidance [42]. Their guidance describes four use cases, along some with variations. Each use case variant includes a list of potential causes along with recommendations for prevention and responses when the scenario is encountered. A mapping of which use cases and variants apply to which subsets of states is given in Table 6.1.

6.2.2 MPCVD Benchmarks

A common problem in MPCVD is that of fairness: coordinators are often motivated to optimize the CVD process to maximize the deployment of fixes to as many end users as possible while minimizing the exposure of users of other affected products to unnecessary risks.

The model presented in this paper provides a way for coordinators to assess the effectiveness of their MPCVD cases. In an MPCVD case, each vendor/product pair effectively has its own 6-event history h_a. We can therefore recast MPCVD as a set of histories $\mathcal{M}$ drawn from the possible histories $\mathcal{H}$:

$$\mathcal{M} = \{h_1, h_2, ..., h_m \text{ where each } h_a \in H\} \tag{6.4}$$

Where $m = |\mathcal{M}| \geq 1$. The edge case when $|\mathcal{M}| = 1$ is simply the regular (non-multiparty) case.

Table 6.1: Applicability of FIRST MPCVD scenarios to subsets of states in our model

States	FIRST Use Case	Description
n/a	0	No vulnerability exists
$VFDp$	1	Vulnerability with no affected users
Vp	1 Variant 1	Product is deployed before vulnerability is discovered or fixed
f	2	Vulnerability with coordinated disclosure
fdP	2 Variant 1	Finder makes the vulnerability details public prior to remediation
$VFdP$	2 Variant 2	Users do not deploy remediation immediately
(multiparty sync)	2 Variant 3	Missing communication between upstream and downstream vendors
$VfdP$	2 Variant 4	Vendor makes the vulnerability details public prior to remediation
Vfd	2 Variant 5	Vendor does not remediate a reported vulnerability
(multiparty sync)	2 Variant 6	Missing communication between peer vendors impedes coordination
fdP	2 Variant 7	Coordinator makes vulnerability details public prior to remediation
$Vfdp$, $vfdp$	2 Variant 8	Finder reports a vulnerability to one vendor that may affect others using the same component
fdP	3	Public disclosure of limited vulnerability information prior to remediation
vP, vX, vA	4	Public disclosure or exploitation of vulnerability prior to vendor awareness
$vfPX$, $vfPA$	4 Variant 1	Finder publishes vulnerability details and vulnerability is exploited
vpA	4 Variant 2	Previously undisclosed vulnerability used in attacks

Table 6.2: CVD Roles and the transitions they can control. Roles can be combined (vendor + deployer, finder + coordinator, etc.). Roles are based on [30].

Role	**V**	**F**	**D**	**P**	**X**	**A**
Finder/Reporter	✓	·	·	✓	✓	·
Vendor	·	✓	·	✓	✓	·
Deployer	·	·	✓	·	·	·
Coordinator	✓	·	·	✓	✓	·
Adversary	·	·	·	·	·	✓

We can then set desired criteria for the set $\mathcal{M}$, as in the benchmarks described in §6.1. In the MPCVD case, we propose to generalize the benchmark concept such that the median $\tilde{\alpha_d}$ should be greater than some benchmark constant c_d:

$$\tilde{\alpha_d} \geq c_d \geq 0 \tag{6.5}$$

In real-world cases where some outcomes across different vendor/product pairs will necessarily be lower than others, we can also add the criteria that we want the variance of each α_d to be low. An MPCVD case having high median α_d with low variance across vendors and products involved will mean that most vendors achieved acceptable outcomes.

To summarize:

- The median α_d for all histories $h \in \mathcal{M}$ should be positive and preferably above some benchmark constant c_d, which may be different for each $d \in \mathbb{D}$.

$$Median(\{\alpha_d(h) : h \in \mathcal{M}\}) \geq c_d > 0 \tag{6.6}$$

- The variance of each α_d for all histories $h \in \mathcal{M}$ should be low. The constant ε is presumed to be small.

$$\sigma^2(\{\alpha_d(h) : h \in \mathcal{M}\}) \leq \varepsilon \tag{6.7}$$

6.3 CVD Roles and Their Influence

CVD stakeholders include vendors, system owners, the security research community, coordinators, and governments [30]. Of interest here are the main roles: *finder/reporter*, *vendor*, *deployer*, and *coordinator*. Each of the roles corresponds to a set of transitions they can cause. For example, a *coordinator* can notify the *vendor* (**V**) but not create the fix (**F**), whereas a *vendor* can create the fix but not notify itself (although a *vendor* with an in-house vulnerability discovery capability might also play the role of a *finder/reporter* as well). A mapping of CVD Roles to the transitions they can control can be found in Table 6.2. We also included a role of *adversary* just to cover the **A** transition.

Different stakeholders might want different things, although most benevolent parties will likely seek some subset of $\mathbb{D}$. Because $\mathcal{H}$ is the same for all stakeholders, the expected frequencies shown in Table 4.1 will be consistent across any such variations in desiderata. A discussion of some stakeholder preferences is given below, while a summary can be found in Table 6.3. We notate these variations of the set of desiderata $\mathbb{D}$ with subscripts: $\mathbb{D}_v$ for vendors, $\mathbb{D}_s$ for system owners, $\mathbb{D}_c$ for coordinators, and $\mathbb{D}_g$ for governments. In Table 3.3 we defined a preference ordering between every possible pairing of events, therefore $\mathbb{D}$ is the largest possible set of desiderata. We thus expect the desiderata of benevolent stakeholders to be a subset of $\mathbb{D}$ in most cases. That said, we note a few exceptions in the text that follows.

Table 6.3: Ordering Preferences for Selected Stakeholders.

$d \in \mathbb{D}$	Vendor ($\mathbb{D}_v$)	SysOwner ($\mathbb{D}_s$)	Coordinator ($\mathbb{D}_c$)
$\mathbf{V} \prec \mathbf{P}$	yes	maybe[4]	yes
$\mathbf{V} \prec \mathbf{X}$	yes	maybe[4]	yes
$\mathbf{V} \prec \mathbf{A}$	yes	maybe[4]	yes
$\mathbf{F} \prec \mathbf{P}$	yes	maybe[5]	yes
$\mathbf{F} \prec \mathbf{X}$	yes	yes	yes
$\mathbf{F} \prec \mathbf{A}$	yes	yes	yes
$\mathbf{D} \prec \mathbf{P}$	maybe[1]	maybe[1]	yes
$\mathbf{D} \prec \mathbf{X}$	maybe[2]	maybe[5]	yes
$\mathbf{D} \prec \mathbf{A}$	maybe[2]	yes	yes
$\mathbf{P} \prec \mathbf{X}$	yes	yes	yes
$\mathbf{P} \prec \mathbf{A}$	yes	yes	yes
$\mathbf{X} \prec \mathbf{A}$	maybe[3]	maybe[3]	maybe[3]

[1] When vendors control deployment, both vendors and system owners likely prefer $\mathbf{D} \prec \mathbf{P}$. When system owners control deployment, $\mathbf{D} \prec \mathbf{P}$ is impossible.
[2] Vendors should care about orderings involving $\mathbf{D}$ when they control deployment, but might be less concerned if deployment responsibility is left to system owners.
[3] Exploit publication can be controversial. To some, it enables defenders to test deployed systems for vulnerabilities or detect attempted exploitation. To others, it provides unnecessary adversary advantage.
[4] System owners may only be concerned with orderings involving $\mathbf{V}$ insofar as it is a prerequisite for $\mathbf{F}$.
[5] System owners might be indifferent to $\mathbf{F} \prec \mathbf{P}$ and $\mathbf{D} \prec \mathbf{X}$ depending on their risk tolerance.

6.3.1 Vendors

As shown in Table 6.3, we expect vendors' desiderata $\mathbb{D}_v$ to be a subset of $\mathbb{D}$. It seems reasonable to expect vendors to prefer that a fix is ready before either exploit publication or attacks ($\mathbf{F} \prec \mathbf{X}$ and $\mathbf{F} \prec \mathbf{A}$, respectively). Fix availability implies vendor awareness ($\mathbf{V} \prec \mathbf{F}$), so we would expect vendors' desiderata to include those orderings as well ($\mathbf{V} \prec \mathbf{X}$ and $\mathbf{V} \prec \mathbf{A}$, respectively).

Vendors typically want to have a fix ready before the public finds out about a vulnerability ($\mathbf{F} \prec \mathbf{P}$). We surmise that a vendor's preference for this item could be driven by at least two factors: the vendor's tolerance for potentially increased support costs (e.g., fielding customer support calls while the fix is being prepared), and the perception that public awareness without an available fix leads to a higher risk of attacks. As above, vendor preference for $\mathbf{F} \prec \mathbf{P}$ implies a preference for $\mathbf{V} \prec \mathbf{P}$ as well.

When a vendor has control over fix deployment ($\mathbf{D}$), it will likely prefer that deployment precede public awareness, exploit publication, and attacks ($\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, and $\mathbf{D} \prec \mathbf{A}$, respectively).[2] However, when fix deployment depends on system owners to take action, the feasibility of $\mathbf{D} \prec \mathbf{P}$ is limited.[3] Regardless of the vendor's ability to deploy fixes or influence their deployment, it would not be unreasonable for them to prefer that public awareness precedes both public exploits and attacks ($\mathbf{P} \prec \mathbf{X}$ and $\mathbf{P} \prec \mathbf{A}$, respectively).

Ensuring the ease of patch deployment by system owners remains a likely concern for vendors. Conscientious vendors might still prefer $\mathbf{D} \prec \mathbf{X}$ and $\mathbf{D} \prec \mathbf{A}$ even if they have no direct control over those factors. However, vendors may be indifferent to $\mathbf{X} \prec \mathbf{A}$.

Although our model only addresses event ordering, not timing, a few comments about timing of events are relevant since they reflect the underlying state transition process from which $\mathcal{H}$ arises. Vendors have significant influence over the speed of $\mathbf{V}$ to $\mathbf{F}$ based on their vulnerability handling, remediation, and development processes [32]. They can also influence how early $\mathbf{V}$ happens based on promoting a cooperative atmosphere with the security researcher community [31]. Vendor architecture and business decisions affect the speed of $\mathbf{F}$ to $\mathbf{D}$. Cloud-based services and automated patch delivery can shorten the lag between $\mathbf{F}$ and $\mathbf{D}$. Vendors that leave deployment contingent on system owner action can be expected to have longer lags, making it harder to achieve the $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, and $\mathbf{D} \prec \mathbf{A}$ objectives, respectively.

6.3.2 System Owners

System owners ultimately determine the lag from $\mathbf{F}$ to $\mathbf{D}$ based on their processes for system inventory, scanning, prioritization, patch testing, and deployment—in other words, their Vulnerability Management (VM) practices. In cases where the vendor and system owner are distinct entities, system owners should optimize to minimize the lag between $\mathbf{F}$ and $\mathbf{D}$ in order to improve the chances of meeting the $\mathbf{D} \prec \mathbf{X}$ and $\mathbf{D} \prec \mathbf{A}$ objectives, respectively. Enabling automatic updates for security patches is one way to improve $\mathbf{F}$ to $\mathbf{D}$ performance, although not all system owners find the resulting risk of operational impact to be acceptable to their change management process.

System owners might select a different desiderata subset than vendors $\mathbb{D}_s \subseteq \mathbb{D}$, $\mathbb{D}_s \neq \mathbb{D}_v$. In general, system owners are primarily concerned with the $\mathbf{F}$ and $\mathbf{D}$ events relative to $\mathbf{X}$ and $\mathbf{A}$. Therefore, we expect system owners to be concerned about $\mathbf{F} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{X}$, and $\mathbf{D} \prec \mathbf{A}$. As discussed above, $\mathbf{D} \prec \mathbf{P}$ is only possible when the vendor controls $\mathbf{D}$. Depending

[2] On the other hand, some vendors might actually prefer public awareness before fix deployment even if they have the ability to deploy fixes, for example in support of transparency or trust building. In that case, $\mathbb{D}_\mathbb{V} \not\subseteq \mathbb{D}$, and some portions of the analysis presented here may not apply.

[3] "Silent patches" can obviously occur when vendors fix a vulnerability but do not make that fact known. In principle, silent patches could achieve $\mathbf{D} \prec \mathbf{P}$ even in traditional COTS or OSS distribution models. However, in practice silent patches result in poor deployment rates precisely because they lack an explicit imperative to deploy the fix.

on the system owner's risk tolerance, $\mathbf{F} \prec \mathbf{P}$ and $\mathbf{D} \prec \mathbf{X}$ may or may not be preferred. Some system owners may find $\mathbf{X} \prec \mathbf{A}$ useful for testing their infrastructure, others might prefer that no public exploits be available.

6.3.3 Security Researchers

The "friendly" offensive security community (i.e., those who research vulnerabilities, report them to vendors, and sometimes release proof-of-concept exploits for system security evaluation purposes) can do their part to ensure that vendors are aware of vulnerabilities as early as possible prior to public disclosure (6.8).

$$vfdpxa \xrightarrow{\mathbf{V}} Vfdpxa \implies \mathbf{V} \prec \mathbf{P},\ \mathbf{V} \prec \mathbf{X} \text{ and } \mathbf{V} \prec \mathbf{A} \tag{6.8}$$

Security researchers can also delay the publication of exploits until after fixes exist (6.9), are public (6.10), and possibly even until most system owners have deployed the fix (6.11).

$$\mathbf{X}|q \in VFdpx \implies \mathbf{F} \prec \mathbf{X} \tag{6.9}$$

$$\mathbf{X}|q \in VFdPx \implies \mathbf{F} \prec \mathbf{X} \text{ and } \mathbf{P} \prec \mathbf{X} \tag{6.10}$$

$$\mathbf{X}|q \in VFDPx \implies \mathbf{F} \prec \mathbf{X},\ \mathbf{P} \prec \mathbf{X} \text{ and } \mathbf{D} \prec \mathbf{X} \tag{6.11}$$

This does not preclude adversaries from doing their own exploit development on the way to **A**, but it avoids providing them with unnecessary assistance.

6.3.4 Coordinators

Coordinators have been characterized as seeking to balance the social good across both vendors and system owners [9]. This implies that they are likely interested in the union of the vendors' and system owners' preferences. In other words, coordinators want the full set of desiderata ($\mathbb{D}_c = \mathbb{D}$).

We pause for a brief aside about the design of the model with respect to the coordination role. We considered adding a *Coordinator Awareness* (**C**) event, but this would expand $|\mathcal{H}|$ from 70 to 452 because it could occur at any point in any h. There is not much for a coordinator to do once the fix is deployed, however, so we could potentially reduce $|\mathcal{H}|$ to 329 by only including positions in $\mathcal{H}$ that precede the **D** event. This is still too large and unwieldy for meaningful analysis within our scope; instead, we simply provide the following comment.

The goal of coordination is this: regardless of which stage a coordinator becomes involved in a case, the objective is to choose actions that make preferred histories more likely and non-preferred histories less likely.

The rules outlined in §6.8 suggest available actions to improve outcomes. Namely, this means focusing coordination efforts as needed on vendor awareness, fix availability, fix deployment, and the appropriately timed public awareness of vulnerabilities and their exploits (**V**,**F**,**D**, **P**, and **X**).

6.3.5 Governments

In their defensive roles, governments act as a combination of system owners, vendors, and—increasingly—coordinators. Therefore we might anticipate $\mathbb{D}_g = \mathbb{D}_c = \mathbb{D}$.

However, governments sometimes also have an adversarial role to play for national security, law enforcement, or other reasons. The model presented in this paper could be adapted to that role by drawing some desiderata from the lower left triangle of Table 3.3. While defining such adversarial desiderata ($\mathbb{D}_a$) is out of scope for this paper, we leave the topic with our expectation that $\mathbb{D}_a \not\subseteq \mathbb{D}$.

6.4 Disclosure Policy Formalization

In this section, we apply our model to the formalization of vulnerability disclosure policies. This discussion is not intended to be a complete formalization of all possible policy choices; rather, we seek to relate how some aspects of disclosure policies might be formalized using our model. In particular, we will look at applications of the model to embargoes and service level expectations.

6.4.1 Embargo Initiation Policies

An agreement between coordinating stakeholders to keep information about the vulnerability private until some exit condition has been met is called an *embargo*.[4] Examples of exit conditions for CVD embargoes include the expiration of a timer or a the occurrence of a triggering event such as fix availability. The model gives us a way of formally specifying the conditions under which initiating or maintaining an embargo may or may not be appropriate.

Let us first examine which states are eligible for embargoes to be initiated. The whole point of an embargo is to restrict public knowledge of the vulnerability for some period of time, corresponding to a desire to avoid entering $\mathcal{Q}_P$ until the embargo ends. Therefore, it follows that our set of embargo initiation states must reside in $\mathcal{Q}_p$. Furthermore, as we have discussed, pX is inherently unstable because publication of an exploit necessarily leads to public awareness of the vulnerability. Because pX leads immediately to PX, we can infer that our embargo entry points must be in px.

Many disclosure policies—including CERT/CC's—eschew embargoes when attacks are underway ($\mathcal{Q}_A$). This implies we should be looking in pxa. We further observe that there is little reason to initiate an information embargo about a vulnerability after the fix has been deployed[5] (D) and thus continue with $dpxa$.

In cases where a single vendor and product are affected, then fix readiness is truly binary (it either is or is not ready), and there may be no need to enter into an embargo when the fix is ready (i.e., in $Fdpxa$). However, while may be tempted to expand the requirement and narrow the states of interest to $fdpxa$, we must allow for the multiparty situation in which some vendors have a fix ready (Fp) while others do not (fp). We discuss MPCVD further in section §6.2. Here we note that in MPCVD cases, prudence requires us to allow for a (hopefully brief) embargo period to enable more vendors to achieve $fp \xrightarrow{\mathbf{F}} Fp$ prior to public disclosure ($\mathbf{F} \prec \mathbf{P}$). Therefore we stick with $dpxa$ for the moment.

Finally, because embargoes are typically an agreement between the vendor and other coordinating parties, it might appear that we should expect embargoes to begin in $Vdpxa$. However, doing so would neglect the possibility of embargoes entered into by finders, reporters, and coordinators prior to vendor awareness—i.e., in $vfdpxa$. In fact, the very concept of CVD is built on the premise that every newly discovered vulnerability should have a default embargo at least until the vendor has been informed about the vulnerability (i.e., $vfdpxa \xrightarrow{\mathbf{V}} Vfdpxa$ is CVD's preferred initial state transition). And so, having considered all possible states, we conclude that embargoes can only begin from $dpxa$, with the caveat that practitioners should carefully consider why they would enter an embargo from $Fdpxa$.

This leaves us with only few states from which we can enter an embargo, which we denote as $\mathcal{Q}_E^0$:

$$\mathcal{Q}_E^0 \stackrel{\text{def}}{=} dpxa = \{vfdpxa, Vfdpxa, VFdpxa\} \tag{6.12}$$

[4] A CVD embargo is analogous to a news embargo used in journalism, often in the context of scientific publications. Like CVD embargoes, the use of scientific news embargoes is not without controversy. [2, 18, 44]

[5] It is of course appropriate to use discretion as to how much detail is released.

6.4.2 Embargo Continuation Policies

Having shown the limited number of states from which an embargo can begin, we turn next to the states in which an embargo remains viable. By *viable* we mean that maintaining the embargo is reasonable and not contraindicated by the facts at hand. It is of course possible for parties to attempt to maintain embargoes in spite of facts indicating they should do otherwise. We are not here to support such contrived arguments.

By definition an embargo must be viable in all the states it can start from, so we'll begin with $\mathcal{Q}_E^0$ ($dpxa$) and consider which restrictions we can relax, as there may be states in which an existing embargo remains viable even if one should not initiate a new one.

To begin, states in $\mathcal{Q}_p$ remain a necessary condition because keeping the public from becoming aware of the vulnerability too early is the key state the embargo is intended to maintain. Furthermore, because our starting criteria is already indifferent to vendor awareness and fix availability, we will not revisit those here.

It appears we can relax $\mathcal{Q}_d$ because it is certainly possible in some situations to maintain an embargo after a fix is deployed, as is common in CVD cases regarding specific instances of vulnerabilities. For example, web site vulnerabilities are often only published after the system is no longer vulnerable. So our set of viable states is currently pxa.

Next, we address two potential sticking points, **X** and **A**. First, how long can an existing embargo persist in pX? The embargo can persist only for as long as it takes for public awareness of its existence to take hold $pX \xrightarrow{\mathbf{P}} PX$. This might be reasonable for a few hours or at best a few days, but not much longer.[6] In other words, an active embargo for a vulnerability case in a state $q \in pX$ has either already failed or is at imminent risk of failure to prevent public awareness. For this reason, it seems better to presume embargo non-viability in pX.

Second, what should we do if an embargo is in place and we find out that attacks are happening—i.e., we assess that we are in a state $q \in pA$? Cases in pA give the attacker an advantage over defenders insofar as attackers are able to exploit vulnerabilities while deployers remain ignorant of steps they might take to defend their systems. For this reason, many organizations' disclosure policies explicitly call out observed attacks as a reason to break an embargo. Therefore, while it may be technically possible to maintain an embargo in pxA, we remain skeptical of the reason for doing so, if not to maintain the ability for attacks to remain stealthy.

In the interest of avoiding justification for bad faith disclosure behaviors, we hold that embargoes remain viable only in pxa, with the caveat that only limited circumstances as described above justify maintaining embargoes once **F** or **D** have occurred ($VFdpxa$ and $VFDpxa$, respectively).

$$\mathcal{Q}_E \stackrel{\text{def}}{=} pxa = \{vfdpxa, Vfdpxa, VFdpxa, VFDpxa\} \tag{6.13}$$

In summary, embargoes can be initiated if the case is in $\mathcal{Q}_E^0$ as in Eq. (6.12), and remain viable through any state in $\mathcal{Q}_E$ as in Eq. (6.13). This in turn gives us specific things to look for in order to determine when to end an embargo:

- The embargo timer has expired.
- Any observation of **P**, **X**, or **A** has been made ($q \notin pxa$).
- **F** or **D** have occurred ($q \in \{VFdpxa, VFDpxa\}$), and no reasons specific to the case to maintain the embargo remain.

[6] Public awareness notwithstanding, an engaged adversary can begin using a public exploit as soon as it becomes available.

- Any other embargo exit rules—such as those specified in the relevant disclosure policies—have been triggered.

6.4.3 CVD Service Level Expectations

Closely related to CVD embargoes are CVD Service Level Expectations (SLEs). Disclosure policies specify commitments by coordinating parties to ensure the occurrence of certain state transitions within a specific period of time. While the model presented here does not directly address timing choices, we can point out some ways to relate the model to those choices. Specifically, we intend to demonstrate how disclosure policy SLEs can be stated as rules triggered within subsets of states or by particular transitions between subsets of states in $\mathcal{Q}$.

For example, a finder, reporter, or coordinator might commit to publishing information about a vulnerability 30 days after vendor notification. This translates to starting a timer at $v \xrightarrow{\mathbf{V}} V$ and ensuring $Vp \xrightarrow{\mathbf{P}} VP$ when the timer expires. Notice that the prospect of $Vfp \xrightarrow{\mathbf{P}} VfP$ is often used to motivate vendors to ensure a reasonable SLE to produce fixes ($Vf \xrightarrow{\mathbf{F}} VF$) [9].

Similarly, a vendor might commit to providing public fixes within 5 business days of report receipt. In that case, the SLE timer would start at $vfp \xrightarrow{\mathbf{V}} Vfp$ and and end at one of two transitions: First, the "normal" situation in which the vendor creates a fix and makes it public along with the vulnerability ($\mathbf{F} \prec \mathbf{P}$, i.e., $VFp \xrightarrow{\mathbf{P}} VFP$). Second, a "zero day" situation[7] in which events outside the vendor's control cause the $Vfp \xrightarrow{\mathbf{P}} VfP$ transition to occur prior to the fix being ready ($VfP \xrightarrow{\mathbf{F}} VFP$), i.e, $\mathbf{P} \prec \mathbf{F}$. Likewise, the $Vfp \xrightarrow{\mathbf{P}} VfP \xrightarrow{\mathbf{F}} VFP$ path might also occur when a vendor has set their embargo timer too aggressively for their development process to keep up.

It is therefore in the vendor's interest to tune their SLE to reduce the likelihood for unexpected public awareness ($\mathbf{P}$) while providing sufficient time for $\mathbf{F}$ to occur, optimizing to achieve $\mathbf{F} \prec \mathbf{P}$ in a substantial fraction of cases. As future work, measurement of both the incidence and timing of embargo failures through observation of $\mathbf{P}$, $\mathbf{X}$, and $\mathbf{A}$ originating from $\mathcal{Q}_E$ could give insight into appropriate vendor SLEs for fix readiness ($\mathbf{F}$).

Service providers and VM practitioners might similarly describe their SLEs in terms of timing between states. Such policies will likely take the form of commitments to limit the time spent in FdP. When the vendor has already produced a patch, the clock starts at $Fdp \xrightarrow{\mathbf{P}} FdP$, whereas if the vulnerability becomes public prior to patch the clock starts at $fdP \xrightarrow{\mathbf{F}} FdP$. In both cases, the timer ends at $FdP \xrightarrow{\mathbf{D}} FDP$.

Future formal definitions of policy statements might take the general form of specifications including

- Starting state ($q \in \mathcal{Q}$) or subset of states ($S \subset \mathcal{Q}$)
- Expected transitions ($\sigma \in \Sigma$) and SLEs around their timing, including possible constraints such as "not before" and "no later than" specifications
- An indication of constraint rigidity (negotiable, fixed, *MUST*, *SHOULD*, *MAY* [11], etc.)
- Potential exceptions in the form of other transitions ($\sigma \in \Sigma$) that might alter expected behavior, and a description of the anticipated response.

One reason to formalize policy definitions is the potential to automatically resolve embargo negotiations when those policies are compatible. It may be possible to resolve some apparent policy conflicts by delaying notifications to some vendors to ensure that all embargo end

[7] The phrase *zero day* means many things to many people. We provide more formal definitions in §6.5

timers expire simultaneously. We informally refer to this as the *Thanksgiving Dinner problem*, due to its similarity to the familiar annual holiday meal in which a number of dishes with varying preparation times are expected to be delivered to the table simultaneously, each at its appropriate temperature and doneness. For example, when one party has a "minimum 90 days to publish" policy and another has a "maximum 5 days to publish" policy, the resolution could be to notify the first vendor 85 days before notifying the second. Such a model of graduated disclosure could work well for cases where notifications are automated and vendor dependencies are small or where coordinating parties' policies can be resolved as compatible. Technology such as Theorem Provers and Solvers seem likely to be suited to this sort of problem.

However, when formal policies are incompatible, the failure to resolve them automatically becomes an opportunity for human intervention as the exception handler of last resort. Of potential concern here are MPCVD cases in which a vendor with a long policy timer is dependent on one with a short policy timer to provide fixes. The serial nature of the dependency creates the potential for a compatibility conflict. For example, this can happen when an OS kernel provider's disclosure policy specifies a shorter embargo timer than the policies of device manufacturers whose products depend on that OS kernel. Automation can draw attention to these sorts of conflicts, but their resolution is likely to require human intervention for some time to come.

6.5 Improving Definitions of Common Terms

Some terms surrounding CVD and VM have been ambiguously defined in common usage. One benefit of the definition of events, states, and possible CVD histories presented in this whitepaper is an opportunity to clarify definitions of related terms. In this section we will use our model to formally define their meaning.

6.5.1 Zero Day

The information security community uses a variety of common phrases that contain the words *zero day*. This creates confusion. For example, a reviewer stated that they prefer to define "zero day vulnerability" as $\mathbf{X} \prec \mathbf{V}$ and not ($\mathbf{P} \prec \mathbf{F}$ or $\mathbf{A} \prec \mathbf{F}$). We should seek these precise definitions because sometimes both $\mathbf{X} \prec \mathbf{V}$ and $\mathbf{P} \prec \mathbf{F}$ are true, in which case two people might agree that an instance is a "zero day" without realizing that they disagree on its definition. We can resolve these formally using our model. This section extends prior work by one of the authors in [28].

zero day vulnerability Two common definitions for this term are in widespread use; a third is drawn from an important policy context. The two commonly-used definitions can be considered a relatively *low* threat level because they only involve states $q \in xa$ where no exploits are public and no attacks have occurred. We ordered all three definitions in approximately descending risk due to the expected duration until $\mathbf{D}$ can be achieved.

1. $q \in vp$ The United States Vulnerability Equities Process [27] defines *zero day vulnerability* in a manner consistent with $q \in vp$. Further discussion appears in §6.7.
2. $vp \xrightarrow{\mathbf{P}} vP$ when the vulnerability becomes public before the vendor is aware of it. Note that our model assumes that states in vP are unstable and resolve to $vP \xrightarrow{\mathbf{V}} VP$ in the next step.
3. $fp \xrightarrow{\mathbf{P}} fP$ when the vulnerability becomes public before a fix is available, regardless of the vendor's awareness. Some states in fP—specifically, those in VfP—are closer to $\mathbf{F}$ (and therefore $\mathbf{D}$) occuring than others (i.e., vfP), thus this definition could imply less time spent at risk than the first.

zero day exploit This term has three common definitions. Each can be considered a *moderate* threat level because they involve transition from $xa \xrightarrow{\mathbf{X}} Xa$. However, we ordered them in approximately descending risk due to the expected duration until **D** can be achieved.

1. $vfdx \xrightarrow{\mathbf{X}} vfdX$ when an exploit is released before the vendor is aware of the vulnerability.
2. $fdx \xrightarrow{\mathbf{X}} fdX$ when an exploit is released before a fix is available for the vulnerability. Because $\mathcal{Q}_{vf} \subset \mathcal{Q}_f$, any scenario matching the previous definition also matches this one.
3. $px \xrightarrow{\mathbf{X}} pX$ when an exploit is released before the public is aware of the vulnerability. Note that our model assumes that states in pX are unstable and transition $pX \xrightarrow{\mathbf{P}} PX$ in the next step.

zero day attack We have identified three common definitions of this term. Each can be considered a *high* threat level because they involve the **A** transition. However, we ordered them in approximately descending risk due to the expected duration until **D** can be achieved.

1. $vfda \xrightarrow{\mathbf{A}} vfdA$ when attacks against the vulnerability occur before the vendor is aware of the vulnerability.
2. $fda \xrightarrow{\mathbf{A}} fdA$ when attacks against the vulnerability occur before a fix is available for the vulnerability. As with *zero day exploit*, because $\mathcal{Q}_{vf} \subset \mathcal{Q}_f$, any scenario matching the previous definition also matches this one.
3. $pa \xrightarrow{\mathbf{A}} pA$ when attacks against the vulnerability occur before the public is aware of the vulnerability. Note that this definition disregards the vendor entirely since it makes no reference to either **V** or **F**.

6.5.2 Forever Day

In common usage, a *forever day* vulnerability is one that is expected to remain unpatched indefinitely [26]. In other words, the vulnerability is expected to remain in d forever. This situation can occur when deployed code is abandoned for a number of reasons, including:

1. The vendor has designated the product as End-of-Life (EoL) and thereby declines to fix any further security flaws, usually implying $q \in Vfd$. Vendors should evaluate their support posture for EoL products when they are aware of vulnerabilities in $VfdX$ or $VfdA$. Potential vendor responses include issuing additional guidance or an out-of-support patch.
2. The vendor no longer exists, implying a state $q \in vfd$. Neither **F** nor **D** transitions can be expected although **P**, **X**, and **A** remain possible. For this reason alone, coordinators or other stakeholders may choose to publish anyway to cause **P**. In this situation, if deployers are to respond at all, states in $vfdP$ are preferable to states in $vfdp$. Defender options in this case are usually limited to retiring or otherwise isolating affected systems, especially for vulnerabilities in either $vfdPX$ or $vfdPA$.
3. The deployer chooses to never deploy, implying an expectation to remain in d until the affected systems are retired or otherwise removed from service. This situation may be more common in deployments of safety-critical systems and Operational Technology (OT) than it is in Information Technology (IT) deployments. It is also the most reversible of the three *forever day* scenarios, because the deployer can always reverse their decision as long as a fix is available ($q \in VF$). In deployment environments where

Table 6.4: Mapping Subsets of States $\mathcal{Q}$ to SSVC v2.0

States	SSVC Decision Point	SSVC Value
xa	Exploitation	None
Xa	Exploitation	PoC (Proof of Concept)
A	Exploitation	Active
p	Report Public	No
P	Report Public	Yes
V	Supplier Contacted	Yes
v	Supplier Contacted	No (but see text for caveat)
p	Public Value Added	Precedence
$VFdp$ or dP	Public Value Added	Ampliative
VFP	Public Value Added	Limited

other mitigations are in place and judged to be adequate, and where the risk posed by **X** and/or **A** are perceived to be low, this can be a reasonable strategy within a VM program.

Scenarios in which the vendor has chosen not to develop a patch for an otherwise supported product, and which also imply $q \in Vfd$, are omitted from the above definition because as long as the vendor exists the choice to not develop a fix remains reversible. That said, such scenarios most closely follow the first bullet in the list above.

6.6 Vulnerability Response Situation Awareness

In this section, we demonstrate how the model can be applied to improve situation awareness for coordinating parties and other stakeholders.

SSVC v2.0 Vulnerability prioritization schemes such as SSVC [52, 53, 54] generally give increased priority to states in higher threat levels, corresponding to $q \in \mathcal{Q}_X \cup \mathcal{Q}_A$. SSVC also includes decision points surrounding other states in our model. A summary of the relevant SSVC decision points and their intersection with our model is given in Table 6.4.

Not all SSVC decision point values map as clearly onto states in this model however. For example, *Supplier Contacted=No* likely means $q \in \mathcal{Q}_v$ but it is possible that the vendor has found out another way, so one cannot rule out $q \in \mathcal{Q}_V$ on this basis alone. However, notifying the vendor yourself forces you into $q \in \mathcal{Q}_V$. Therefore it is always in the coordinator's interest to encourage, facilitate, or otherwise cause the vendor to be notified.

Other SSVC decision points may be informative about which transitions to expect in a case. Two examples apply here: First, *Supplier Engagement* acts to gauge the likelihood of the **F** transitions. Coordination becomes more necessary the lower that likelihood is. Second, *Utility* (the usefulness of the exploit to the adversary) acts to gauge the likelihood of the **A** transition.

Mapping to CVSS v3.1 Common Vulnerability Scoring System (CVSS) version 3.1 includes a few Temporal Metric variables that connect to this model [41]. Unfortunately, differences in abstraction between the models leaves a good deal of ambiguity in the translation. Table 6.5 shows the relationship between the two models.

Addressing Uncertainty in Situation Awareness It is possible to use this model to infer what other decisions can be made based on incomplete information about a case. For example, imagine that a vendor just found out about a vulnerability in a product and has taken no

Table 6.5: Mapping Subsets of States $\mathcal{Q}$ to CVSS v3.1

States	CVSS v3.1 Temporal Metric	CVSS v3.1 Temporal Metric Value(s)
XA	Exploit Maturity	High (H), or Functional (F)
X	Exploit Maturity	High (H), Functional (F), or Proof-of-Concept (P)
x	Exploit Maturity	Unproven (U) or Not Defined (X)
Vf	Remediation Level	Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T)
VF	Remediation Level	Temporary Fix (T) or Official Fix (O)

Table 6.6: PageRank and normalized state probabilities for states in Vf

State	PageRank	Normalized
$VfdPXA$	0.063	0.245
$VfdPXa$	0.051	0.200
$VfdPxa$	0.037	0.146
$VfdPxA$	0.032	0.126
$Vfdpxa$	0.031	0.120
$VfdpxA$	0.020	0.078
$VfdpXa$	0.011	0.044
$VfdpXA$	0.010	0.040

action yet. We know they are in Vf, but that leaves 8 possible states for the case to be in:

$$\begin{aligned} Vf = \{&Vfdpxa, VfdPxa, VfdpXa, \\ &VfdpxA, VfdPXa, VfdpXA, \\ &VfdPxA, VfdPXA\} \end{aligned}$$

Can we do better than simply assigning equal likelihood $p(q|Vf) = 0.125$ to each of these states? Yes: we can use the PageRank computations from Table 3.4 to inform our estimates.

To assess our presence in Vf, we can select just the subset of states we are interested in. But our PageRank values are computed across all 32 states and we are only interested in the relative probabilities within a subset of 8 states. Thus, we normalize the PageRank for the subset to find the results shown in Table 6.6. As a result, we find that the most likely state in Vf is $VfdPXA$ with probability 0.24, nearly twice what we would have expected ($1/8 = 0.125$) if we just assumed each state was equally probable.

6.7 Vulnerability Equities Process (VEP)

The Vulnerability Equities Process (VEP) is the United States government's process to decide whether to inform vendors about vulnerabilities they have discovered. The VEP Charter [27] describes the process:

> The Vulnerabilities Equities Process (VEP) balances whether to disseminate vulnerability information to the vendor/supplier in the expectation that it will be patched, or to temporarily restrict the knowledge of the vulnerability to the USG, and potentially other partners, so that it can be used for national security and law enforcement purposes, such as intelligence collection, military operations, and/or counterintelligence.

For each vulnerability that enters the process, the VEP results in a decision to *disseminate* or *restrict* the information.

In terms of our model:

disseminate is a decision to notify the vendor, thereby triggering the transition $\mathcal{Q}_v \xrightarrow{\mathbf{V}} \mathcal{Q}_V$.

restrict is a decision not to notify the vendor and remain in $\mathcal{Q}_v$.

VEP policy does not explicitly touch on any other aspect of the CVD process. By solely addressing **V**, VEP is mute regarding intentionally triggering the **P** or **X** transitions. It also makes no commitments about **F** or **D**, although obviously these are entirely dependent on **V** having occurred. However, preserving the opportunity to exploit the vulnerability implies a chance that such use would be observed by others, thereby resulting in the **A** transition.

The charter sets the following scope requirement as to which vulnerabilities are eligible for VEP:

> To enter the process, a *vulnerability* must be both *newly discovered* and not *publicly known*

given the following definitions (from Annex A of [27])

Newly Discovered After February 16, 2010, the effective date of the initial Vulnerabilities Equities Process, when the USG discovers a zero-day vulnerability or new zero-day vulnerability information, it will be considered newly discovered. This definition does NOT preclude entry of vulnerability information discovered prior to February 16, 2010.

Publicly known A vulnerability is considered publicly known if the vendor is aware of its existence and/or vulnerability information can be found in the public domain (e.g., published documentation, Internet, trade journals).

Vulnerability A weakness in an information system or its components (e.g., system security procedures, hardware design, internal controls) that could be exploited or impact confidentiality, integrity, or availability of information.

Zero-Day Vulnerability A type of vulnerability that is unknown to the vendor, exploitable, and not publicly known.

Mapping back to our model, the VEP definition of *newly discovered* hinges on the definition of *zero day vulnerability*. The policy is not clear what distinction is intended by the use of the term *exploitable* in the *zero day vulnerability* definition, as the definition of *vulnerability* includes the phrase "could be exploited," seeming to imply that a non-exploitable vulnerability might fail to qualify as a *vulnerability* altogether. Regardless, "unknown to the vendor" clearly matches with $\mathcal{Q}_v$, and "not publicly known" likewise matches with $\mathcal{Q}_p$. Thus we interpret their definition of *newly discovered* to be consistent with $q \in vp$.

VEP's definition of *publicly known* similarly specifies either "vendor is aware" ($\mathcal{Q}_V$) or "information can be found in the public domain" ($\mathcal{Q}_P$). As above, the logical negation of these two criteria puts us back in $q \in vp$ since $vp = \neg\mathcal{Q}_V \cap \neg\mathcal{Q}_P$. We further note that because a public exploit ($\mathcal{Q}_X$) would also meet the definition of "vulnerability information in the public domain," we can narrow the scope from vp to vpx. Lastly, we note that due to the vendor fix path causality rule in Eq. (3.3), vpx is equivalent to $vfdpx$, and therefore we can formally specify that VEP is only applicable to vulnerabilities in

$$\mathcal{S}_{VEP} = vfdpx = \{vfdpxa, vfdpxA\} \tag{6.14}$$

Vulnerabilities in any other state by definition should not enter into the Vulnerability Equities Process, as the first transition from $vfdpx$ (i.e., **V**, **P**, or **X**) exits the inclusion criteria. However it is worth mentioning that the utility of a vulnerability for offensive use continues throughout $\mathcal{Q}_d$, which is a considerably larger subset of states than $vfdpx$ ($|\mathcal{Q}_d| = 24$, $|\mathcal{Q}_{vfdpx}| = 2$).

6.8 Recommended Action Rules for CVD

Another application of this model is to recommend actions for coordinating parties in CVD based on the subset of states that currently apply to a case. What a coordinating party does depends on their role and where they engage, as shown in the list below. As described in §6.2, MPCVD attempts to synchronize state transitions across vendors.

A significant portion of Coordinated Vulnerability Disclosure can be formally described as a set of action rules based on this model. For our purposes, a CVD action rule consists of:

State subset The subset of states $Q \in \mathcal{Q}$ from which the action may be taken

Role(s) The role(s) capable of performing the action

Action A summary of the action to be taken

Reason The rationale for taking the action

Transition The state transition event $\sigma \in \Sigma$ induced by the action (if any)

This rule structure follows a common user story pattern:

> When a case is in a state $q \in Q \subseteq \mathcal{Q}$, a *Role* can do *Action* for *Reason*, resulting in the transition event $\sigma \in \Sigma$

The list in Table 6.7 can be built into a rules engine that translates each state in the model to a set of suggested CVD actions.

Table 6.7: CVD Action Rules based on States

State Subset	Role(s)	Action	Reason	σ
P	any	Terminate any existing embargo	Exit criteria met	-
X	any	Terminate any existing embargo	Exit criteria met	-
A	any	Terminate any existing embargo	Exit criteria met	-
x	any	Monitor for exploit publication	SA	-
X	any	Monitor for exploit refinement	SA	-
a	any	Monitor for attacks	SA	-
A	any	Monitor for additional attacks	SA	-
$vfdP$	vendor	Pay attention to public reports	SA	**V**
pX	any	Draw attention to published exploit(s)	SA	**P**
PX	any	Draw attention to published exploit(s)	SA	**P**
pxa	any	Maintain vigilance for embargo exit criteria	SA	-
$VfdP$	any	Escalate vigilance for exploit publication or attacks	SA, Coordination	-
X	any	Publish detection(s) for exploits	Detection	**P**
A	any	Publish detection(s) for attacks	Detection	**P**
Vp	any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
fdP	any	Publish mitigations	Defense	-
pX	any	Publish vul and any mitigations	Defense	**P**
PX	any	Publish vul and any mitigations	Defense	**P**
pA	any	Publish vul and any mitigations	Defense	**P**
$VfdP$	any	Publish mitigations	Defense	-

Continued on next page

Table 6.7: CVD Action Rules based on States

State Subset	Role(s)	Action	Reason	σ
$vfdp$	any	Publish vul and any mitigations (if no vendor exists)	Defense	**P**
$VfdP$	any	Ensure any available mitigations are publicized	Defense	-
Vfd	vendor	Create fix	Defense	**F**
$VFdp$	vendor, deployer	Deploy fix (if possible)	Defense	**D**
$VFdP$	deployer	Deploy fix	Defense	**D**
$fdPxA$	any	Publish exploit code	Defense, Detection	**X**
$VFdPxa$	any	Publish exploit code	Defense, Detection, Accelerate deployment	**X**
vfd	non-vendor	Notify vendor	Coordination	**V**
dP	any	Escalate response priority among responding parties	Coordination	-
dX	any	Escalate response priority among responding parties	Coordination	-
dA	any	Escalate response priority among responding parties	Coordination	-
Vfd	non-vendor	Encourage vendor to create fix	Coordination	-
pxa	any	Maintain any existing disclosure embargo	Coordination	-
$dpxa$	any	Negotiate or establish disclosure embargo	Coordination	-
$VfdP$	non-vendor	Escalate fix priority with vendor	Coordination	-
$Vfdp$	non-vendor	Publish vul	Coordination, Motivate vendor to fix	**P**
$Vfdp$	any	Publish vul	Coordination, Motivate deployers to mitigate	**P**
$VFdp$	non-vendor	Encourage vendor to deploy fix (if possible)	Coordination	-
$VFdpxa$	any	Scrutinize appropriateness of initiating a new embargo	Coordination	-
$VFdp$	any	Publish vul and fix details	Accelerate deployment	**P**
$VFdP$	any	Promote fix deployment	Accelerate deployment	-
$VFDp$	any	Publish vulnerability	Document for future reference	**P**

Continued on next page

Table 6.7: CVD Action Rules based on States

State Subset	Role(s)	Action	Reason	σ
$VFDp$	any	Publish vulnerability	Acknowledge contributions	**P**
$fdxa$	any	Discourage exploit publication until at least **F**	Limit attacker advantage	-
$vfdpx$	US Gov't	Initiate VEP (if applicable)	Policy	-
$VFDPXA$	any	Close case	No action required	-
$VFDPxa$	any	Close case (unless monitoring for X or A)	No action required	-
$VFDPXa$	any	Close case (unless monitoring for A)	No action required	-
$VFDPxA$	any	Close case (unless monitoring for X)	No action required	-

7 Related Work

Numerous models of the vulnerability life cycle and CVD have been proposed. Arbaugh, Fithen, and McHugh provide a descriptive model of the life cycle of vulnerabilities from inception to attacks and remediation [3], which we refined with those of Frei et al. [24], and Bilge and et al. [10] to form the basis of this model as described in §2.1. We also found Lewis' literature review of vulnerability lifecycle models to be useful [37].

Prescriptive models of the CVD process have also been proposed. Christey and Wysopal's 2002 IETF draft laid out a process for responsible disclosure geared towards prescribing roles, responsibilities for researchers, vendors, customers, and the security community [17]. The NIAC Vulnerability Disclosure Framework also prescribed a process for coordinating the disclosure and remediation of vulnerabilities [16]. The CERT Guide to Coordinated Vulnerability Disclosure provides a practical overview of the CVD process [30]. ISO/IEC 29147 describes standard externally-facing processes for vulnerability disclosure from the perspective of a vendor receiving vulnerability reports , while ISO/IEC 30111 describes internal vulnerability handling processes within a vendor [31, 32]. The Forum of Incident Response and Security Teams (FIRST) *PSIRT Services Framework* provides a practical description of the capabilities common to vulnerability response within vendor organizations [43]. The FIRST *Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure* provides a number of scenarios for MPCVD [42]. Many of these scenarios can be mapped directly to the histories $h \in H$ described in §6.2.

Benchmarking CVD capability is the topic of the Vulnerability Coordination Maturity Model (VCMM) from Luta Security [49]. The VCMM addresses five capability areas: organizational, engineering, communications, analytics, and incentives. Of these, our model is perhaps most relevant to the analytics capability, and the metrics described in §5 could be used to inform an organization's assessment of progress in this dimension. Concise description of case states using the model presented here could also be used to improve the communications dimension of the VCMM.

System dynamics and agent based models have been applied to the interactions between the vulnerability discovery, disclosure, and remediation processes. Ellis et al. analyzed the composition of the labor market for bug bounty programs, finding that a small core of high-volume reporters earn most of the bounties while a much larger group are infrequent low-volume reporters [21]. Lewis modeled the interaction of social and economic factors in the global vulnerability discovery and disclosure system [37]. The key systemic themes identified include:

> Perception of Punishment; Vendor Interactions; Disclosure Stance; Ethical Considerations; Economic factors for Discovery and Disclosure and Emergence of New Vulnerability Markets

Moore and Householder modeled cooperative aspects of the MPCVD process, noting, "it appears that adjusting the embargo period to increase the likelihood that patches can be developed by most vendors just in time is a good strategy for reducing cost"[38].

Economic analysis of CVD has also been done. Arora et al. explored the CVD process from an economic and social welfare perspective [8, 7, 4, 9, 5, 6]. More recently, so did Silfversten [51]. Cavusoglu and Cavusoglu model the mechanisms involved in motivating vendors to produce and release patches [13]. Ellis et al. examined the dynamics of labor market for bug bounties both within and across CVD programs [21]. Pupillo et al. explored the policy implications of CVD in Europe [47]. A model for prioritizing vulnerability response that considers **X** and **A**, among other impact factors, can be found in Spring et al. [53].

Other work has examined the timing of events in the lifecycle, sometimes with implications for forecasting. Ozment and Schechter examined the rate of vulnerability reports as software ages [45]. Bilge and Dumitraş studied 18 vulnerabilities in which $pa \xrightarrow{\mathbf{A}} pA \xrightarrow{\mathbf{P}} PA$, finding a lag of over 300 days [10]. Jacobs et al. proposed an Exploit Prediction Scoring System [33], which could provide insight into the relative frequencies of

$$vfda \xrightarrow{\mathbf{V}} Va \xrightarrow{\mathbf{A}} VA \text{ vs. } vfda \xrightarrow{\mathbf{A}} vA \xrightarrow{\mathbf{V}} VA$$
$$fda \xrightarrow{\mathbf{F}} Fa \xrightarrow{\mathbf{A}} FA \text{ vs. } fda \xrightarrow{\mathbf{A}} fA \xrightarrow{\mathbf{F}} FA$$

and possibly other transitions.

Future work might apply similar measurements of state subset populations over time to put better bounds on state transition probabilities than our simplified assumption of uniformity. Some possible starting points for such analysis follow.

Householder et al. found that only about 5% of vulnerabilities have public exploits available via commodity tools. However, for those that do, the median lag between transitions in $px \xrightarrow{\mathbf{P}} Px \xrightarrow{\mathbf{X}} PX$ was 2 days [29].

Frei et al. describe the timing of many of the events here, including **F**, **D**, **X**, **P**, and the elapsed time between them for the period 2000-2007 across a wide swath of industry [24]. Their analysis finds that $px \xrightarrow{\mathbf{X}} pX \xrightarrow{\mathbf{P}} PX$ in 15% of the vulnerabilities they analyzed, leaving 85% on the $px \xrightarrow{\mathbf{P}} Px \xrightarrow{\mathbf{X}} PX$ path. Similarly, they report that a patch is available on or before the date of public awareness in 43% of vulnerabilities. In other words, they find that $fp \xrightarrow{\mathbf{F}} Fp \xrightarrow{\mathbf{P}} FP$ 43% of the time, implying that $fp \xrightarrow{\mathbf{P}} fP \xrightarrow{\mathbf{F}} FP$ 57% of the time.

8 Limitations and Future Work

This section highlights some limitations of the current work and lays out a path for improving on those limitations in future work. Broadly, the opportunities for expanding the model include

- addressing the complexities of tracking CVD and MPCVD cases throughout their lifecycle
- addressing the importance of both state transition probabilities and the time interval between them
- options for modeling attacker behavior
- modeling multiple agents
- gathering more data about CVD in the world
- managing the impact of partial information
- working to account for fairness and the complexity of MPCVD

8.1 State Explosion

Although our discussion of MPCVD in §6.2 and §6.2.2 highlights one area in which the number of states to track can increase dramatically, an even larger problem could arise in the context of Vulnerability Management (VM) efforts even within normal CVD cases. Our model casts each event $\sigma \in \Sigma$ as a singular point event, even though some—such as fix deployed **D**—would be more accurately described as diffusion or multi-agent processes.

That is, by the time a vulnerability case reaches the point of remediating individual instances of vulnerable deployments, every such instance has its own state to track in regards to whether **D** has occurred yet. To apply this model to real world observations, it may be pragmatic to adapt the event definition to include some defined threshold criteria.

However, this problem is equivalent to an existing problem in VM practice: how best to address the question of whether the fix for a vulnerability has been deployed across the enterprise. Many organizations find a fixed quantile SLE to be a reasonable approach. For example, a stakeholder might set the SLE that 80% of known vulnerable systems will be patched within a certain timeframe. Other organizations might track fix deployments by risk groups, for example by differentiating between end user systems, servers, and network infrastructure. They then could observe the deployed fix ratio for their constituency and mark the event **D** as having occurred when certain thresholds are reached. Nothing in our model precludes those sorts of roll-up functions from being applied.

8.2 The Model Does Not Address Transition Probabilities

Although we posit a skill-less baseline in which each transition is equally likely whenever possible within the model, it is a reasonable criticism to point out that some transitions may be expected to change conditional on a history already in progress.

For example, many people believe that the publication of exploits increases the likelihood of attacks. Our model moves toward making this a testable hypothesis: Does $p(\mathbf{A}|q \in \mathcal{Q}_X) > p(\mathbf{A}|q \in \mathcal{Q}_x)$ over some set of cases? Other such hypotheses can be framed in terms of the model.

Does making vulnerabilities public prior to fix readiness increase attacks?

$$p(\mathbf{A}|q \in fP) > p(\mathbf{A}|q \in FP)?$$

Does notifying vendors prior to making vulnerability information public increase the likelihood that fixes will be deployed before attacks are observed?

$$p(\mathbf{D} \prec \mathbf{A}|\mathbf{V} \prec \mathbf{P}) > p(\mathbf{D} \prec \mathbf{A}|\mathbf{P} \prec \mathbf{V})?$$

The novelty here is not that these questions could not be asked or answered previously. Rather, it is that the formalism of our model allows them to be stated concisely and measured in terms of 6 events $\sigma \in \Sigma$, which points directly to the usefulness of collecting data about those events as part of ongoing CVD (including MPCVD) practices.

8.3 The Model Does Not Achieve a Total Order Over Histories

As described in §4.4, some ambiguity remains regarding preferences for elements of $\mathbb{D}$. These preferences would need to be addressed before the model can achieve a total order over histories $\mathcal{H}$. Specifically, we need to decide whether it is preferable

- that Fix Ready precede Exploit Publication ($\mathbf{F} \prec \mathbf{X}$) or that Vendor Awareness precede Public Awareness ($\mathbf{V} \prec \mathbf{P}$)
- that Public Awareness precede Exploit Publication ($\mathbf{P} \prec \mathbf{X}$) or that Exploit Publication Precede Attacks ($\mathbf{X} \prec \mathbf{A}$)
- that Public Awareness precede Attacks ($\mathbf{P} \prec \mathbf{A}$) or Vendor Awareness precede Exploit Publication ($\mathbf{V} \prec \mathbf{X}$)

We look forward to the ensuing "would you rather...?" discussions.

8.4 The Model Has No Sense of Timing

There is no concept of time in this model, but delays between events can make a big difference in history results. Two cases in which $\mathbf{F} \prec \mathbf{A}$ would be quite different if the time gap between these two events was 1 week versus 3 months, as this gap directly bears on the need for speed in deploying fixes. Organizations may wish to extend this model by setting timing expectations in addition to simple precedence preferences. For example, organizations may wish to specify SLEs for $\mathbf{V} \prec \mathbf{F}$, $\mathbf{F} \prec \mathbf{D}$, $\mathbf{F} \prec \mathbf{A}$, and so forth.

Furthermore, in the long run the elapsed time for $\mathbf{F} \prec \mathbf{A}$ essentially dictates the response time requirements for Vulnerability Management (VM) processes for system owners. Neither system owners nor vendors get to choose when attacks happen, so we should expect stochasticity to play a significant role in this timing. However, if an organization cannot consistently achieve a shorter lag between $\mathbf{F}$ and $\mathbf{D}$ than between $\mathbf{F}$ and $\mathbf{A}$ (i.e., achieving $\mathbf{D} \prec \mathbf{A}$) for a sizable fraction of the vulnerability cases they encounter, it's difficult to imagine that organization being satisfied with the effectiveness of their VM program.

8.5 Attacks As Random Events

In the model presented here, attacks are modeled as random events. However, attacks are not random. At an individual or organization level, attackers are intelligent adversaries and can be expected to follow their own objectives and processes to achieve their ends.

Modeling the details of various attackers is beyond the scope of this model. Thus we believe that a stochastic approach to adversarial actions is reasonable from the perspective of a vendor or system owner. Furthermore, if attacks were easily predicted, we would be having a very different conversation.

8.6 Modeling Multiple Agents

We agree with the reviewer who suggested that an agent-based model could allow deeper examination of the interactions between stakeholders in MPCVD. Many of the mechanisms and proximate causes underlying the events this model describes are hidden from the model, and would be difficult to observe or measure even if they were included.

Nevertheless, to reason about different stakeholders' strategies and approaches to MPCVD, we need a way to measure and compare outcomes. The model we present here gives us such a framework, but it does so by making a tradeoff in favor of generality over causal specificity. We anticipate that future agent-based models of MPCVD will be better positioned to address process mechanisms, whereas this model will be useful to assess outcomes independently of the mechanisms by which they arise.

8.7 Gather Data About CVD

§6.1 discusses how different benchmarks and "reasonable baseline expectations" might change the results of a skill assessment. It also proposes how to use observations of the actions a certain team or team performs to create a baseline which compares other CVD practitioners to the skill of that team or teams. Such data could also inform causal reasoning about certain event orderings and help identify effective interventions. For example, might causing $\mathbf{X} \prec \mathbf{F}$ be an effective method to improve the chances of $\mathbf{D} \prec \mathbf{A}$ in cases where the vendor is slow to produce a fix? Whether it is better to compare the skill of a team to blind luck via the i.i.d. assumption or to other teams via measurement remains an open question.

To address questions such as this, future research efforts must collect and collate a large amount of data about the timing sequences of events in the model for a variety of stakeholder groups and a variety of vulnerabilities. Deeper analysis using joint probabilities could then continue if the modeling choice is to base skill upon a measure from past observations.

While there is a modeling choice about using the uniformity assumption versus observations from past CVD (see §6.1), the model does not depend on whether the uniformity assumption actually holds. We have provided a means to calculate from observations a deviation from the desired "reasonable baseline," whether this is based on the i.i.d. assumption or not. Although, via our research questions, we have provided a method for evaluating skill in CVD, evaluating the overarching question of *fairness* in MPCVD requires a much broader sense of CVD practices.

8.8 Observation May Be Limited

Not all events $\sigma \in \Sigma$, and therefore not all desiderata $d \in \mathbb{D}$, will be observable by all interested parties. But in many cases at least some are, which can still help to infer reasonable limits on the others, as shown in §5.2.1.

Vendors are in a good position to observe most of the events in each case. This is even more so if they have good sources of threat information to bolster their awareness of the **X** and **A** events. A vigilant public can also be expected to eventually observe most of the events, although **V** might not be observable unless vendors, researchers, and/or coordinators are forthcoming with their notification timelines (as many increasingly are). **D** is probably the hardest event to observe for all parties, for the reasons described in the timing discussion above.

8.9 CVD Action Rules Are Not Algorithms

The rules given in §6.8 are not algorithms. We do not propose them as a set of required actions for every CVD case. However, following Atul Gawande's lead, we offer them as a mecha-

nism to generate CVD checklists:

> Good checklists, on the other hand are precise. They are efficient, to the point, and easy to use even in the most difficult situations. They do not try to spell out everything–a checklist cannot fly a plane. Instead, they provide reminders of only the most critical and important steps–the ones that even the highly skilled professional using them could miss. Good checklists are, above all, practical [25].

8.10 MPCVD Criteria Do Not Account for Equitable Resilience

The proposed criteria for MPCVD in §6.2.2 fail to account for either user populations or their relative importance. For example, suppose an MPCVD case had a total of 15 vendors, with 5 vendors representing 95% of the total userbase achieving highly preferred outcomes and 10 vendors with poor outcomes representing the remaining 5% of the userbase. The desired criteria (high median α score with low variance) would likely be unmet even though most users were protected.

Similarly, a smaller set of vendor/product pairs might represent a disproportionate concentration of the total risk posed by a vulnerability.[1] Again, aggregation across all vendor/product pairs could be misleading. In fact, risk concentration within a particular user population may lead to a need for strategies that appear inequitable at the vendor level while achieving greater outcome equity at a larger scale.

The core issue is that we lack a utility function to map from observed case histories to harm reduction.[2] Potential features of such a function include aggregation across vendors and/or users. Alternatively, it may be possible to devise a method for weighting the achieved histories in an MPCVD case by some proxy for total user risk. Other approaches remain possible—for example, employing a heuristic to avoid catastrophic outcomes for all, then applying a weighted sum over the impact to the remaining users. Future work might also consider whether criteria other than high median and low variance could be applied.

Regardless, achieving accurate estimates of such parameters is likely to remain challenging. Equity in MPCVD may be a topic of future interest to groups such as the FIRST Ethics Special Interest Group (SIG)[3].

8.11 MPCVD Is Still Hard

CVD is a wicked problem, and MPCVD even more so [30]. The model provided by this white paper offers structure to describe the problem space where there was little of it to speak of previously.

However, such a model does not significantly alter the complexity of the task of coordinating the response of multiple organizations, many of which identify as each others' competitors, in order to bring about a delicate social good in the face of many incentives for things to go otherwise. The social, business, and geopolitical concerns and interests that influence cybersecurity policy across the globe remain at the heart of the vulnerability disclosure problem for most stakeholders. Our hope is that the model found here will help to clarify decisions, communication, and policies that all have their part to play in MPCVD process improvement.

---

[1] User concentration is one way to think about risk, but it is not the only way. Value density, as defined in [53] is another.

[2] We also admit our omission from consideration of whether utilitarianism is even the best way to approach these problems; and if it is, which variety of utilitarianism may be best suited. Such topics, while both interesting and relevant, lie too far afield from our main topic for us to to them justice here. We direct interested readers toward [20] as an introduction to the general topic.

[3] https://ethicsfirst.org/

9 Conclusion

In this report, we developed a state-based model of the CVD process that enables us to enumerate all possible CVD histories $\mathcal{H}$ and defined a set of desired criteria $\mathbb{D}$ that are preferable in each history. This allowed us to create a partially ordered set over all histories and to compute a baseline expected frequency for each desired criteria. We also proposed a new performance indicator for comparing actual CVD experiences against a benchmark, and proposed an initial benchmark based on the expected frequency of each desired criteria. We demonstrated this performance indicator in a few examples, indicating that at least some CVD practices appear to be doing considerably better than random. Finally, we posited a way to apply these metrics to measure the efficacy of MPCVD processes.

The resulting state-transition model has numerous applications to formalizing the specification of CVD policies and processes. We discussed how the model can be used to specify embargo and disclosure policies, and to bring consistency to coordination practices. We further showed how the model can be used to reduce uncertainty regarding actions to take even in the presence of incomplete CVD information. We also suggested how the model can be used to normalize frequently-used terms that have lacked consistent definitions among practitioners of CVD and VM. Finally, we demonstrated the potential application of this model to US VEP scope definitions.

In combination, the model described in this report offers a way to observe, communicate, and measure the quality improvement of CVD and MPCVD practices across the board.

Request for Feedback

The CERT/CC is interested to receive feedback on this report. Although every action was taken to ensure the completeness and accuracy of the information contained within this report, the possibility for improvement still exists. Please feel free to contact the author providing recommendations, corrections, opinions, or requests for clarification.

Feedback may be submitted at `https://www.sei.cmu.edu/contact-us/`

To contact the author please address all mail to:
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA
Email: info@sei.cmu.edu

A Per-State Details

This appendix gives a brief description of each state $q \in \mathcal{Q}$ as developed in §2. See §2.3 for an explanation of the states in the model. States are presented in the order given in (2.3), which follows a hierarchy implied by traversal of the PXA submodel found in (2.7) for each step of the VFD submodel given in (2.5). See §2.4 for an explanation of the state transitions permitted by the model.

In this appendix, state transitions are cross-referenced by page number to enable easier navigation through the state descriptions. See §3.2 for more on transition ordering desiderata. Where applicable, the specific definitions of *zero day* matched by a given state are shown based on §6.5.1. Additional notes on each state are consistent with §6.6. Also included for each state is a table containing suggested actions as derived from §6.8. The embargo initiation, continuation, and exit advice in those rules are consistent with the discussion found in §6.4. Each state is given its own page to allow for consistent formatting.

A.1 vfdpxa

Vendor is unaware of vulnerability. Fix is not ready. Fix has not been deployed. Public is unaware of vulnerability. No exploits have been made public. No attacks have been observed.

Previous State(s): N/A

Next State(s): $vfdpxA$ (p.65), $vfdpXa$ (p.66), $vfdPxa$ (p.68), $Vfdpxa$ (p.72)

Desiderata met: N/A

Desiderata blocked: N/A

Zero Day Definitions Matched: Zero Day Vulnerability Type 1

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: Unproven (U) or Not Defined (X). Embargo continuation is viable. Embargo initiation may be appropriate. SSVC v2 Exploitation: None. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: No. VEP remains tenable. See Table A.1 for actions.

Table A.1: CVD Action Options for State $vfdpxa$

Role	Action	Reason	Transition
any	Publish vul and any mitigations (if no vendor exists)	Defense	**P**
non-vendor	Notify vendor	Coordination	**V**
any	Monitor for exploit publication	SA	-
any	Monitor for attacks	SA	-
any	Maintain vigilance for embargo exit criteria	SA	-
any	Maintain any existing disclosure embargo	Coordination	-
any	Negotiate or establish disclosure embargo	Coordination	-
any	Discourage exploit publication until at least F	Limit attacker advantage	-
US Gov't	Initiate VEP (if applicable)	Policy	-

A.2 vfdpxA

Vendor is unaware of vulnerability. Fix is not ready. Fix has not been deployed. Public is unaware of vulnerability. No exploits have been made public. Attacks have been observed.

Previous State(s): $vfdpxa$ (p.64)

Next State(s): $vfdpXA$ (p.67), $vfdPxA$ (p.69), $VfdpxA$ (p.73)

Desiderata met: N/A

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{X} \prec \mathbf{A}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 1, Zero Day Attack Type 1, Zero Day Attack Type 2, Zero Day Attack Type 3

Other notes: Attack success likely. Embargo is at risk. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: No. VEP remains tenable. See Table A.2 for actions.

Table A.2: CVD Action Options for State $vfdpxA$

Role	Action	Reason	Transition
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vul and any mitigations (if no vendor exists)	Defense	**P**
non-vendor	Notify vendor	Coordination	**V**
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit publication	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
US Gov't	Initiate VEP (if applicable)	Policy	-

A.3 vfdpXa

Vendor is unaware of vulnerability. Fix is not ready. Fix has not been deployed. Public is unaware of vulnerability. Exploit(s) have been made public. No attacks have been observed.

Previous State(s): $vfdpxa$ (p.64)

Next State(s): $vfdPXa$ (p.70)

Desiderata met: $\mathbf{X} \prec \mathbf{A}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 1, Zero Day Exploit Type 1, Zero Day Exploit Type 2, Zero Day Exploit Type 3

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H), functional (F), or proof of concept (P). Embargo is at risk. Expect both Vendor and Public awareness imminently. SSVC v2 Exploitation: PoC. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: No. VEP does not apply. See Table A.3 for actions.

Table A.3: CVD Action Options for State $vfdpXa$

Role	Action	Reason	Transition
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vul and any mitigations (if no vendor exists)	Defense	**P**
non-vendor	Notify vendor	Coordination	**V**
any	Terminate any existing embargo	Exploit is public	-
any	Monitor for exploit refinement	SA	-
any	Monitor for attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-

A.4 vfdpXA

Vendor is unaware of vulnerability. Fix is not ready. Fix has not been deployed. Public is unaware of vulnerability. Exploit(s) have been made public. Attacks have been observed.

Previous State(s): $vfdpxA$ (p.65)

Next State(s): $vfdPXA$ (p.71)

Desiderata met: N/A

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 1, Zero Day Exploit Type 1, Zero Day Exploit Type 2, Zero Day Exploit Type 3, Zero Day Attack Type 1, Zero Day Attack Type 2, Zero Day Attack Type 3

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H) or functional (F). Embargo is at risk. Expect both Vendor and Public awareness imminently. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: No. VEP does not apply. See Table A.4 for actions.

Table A.4: CVD Action Options for State $vfdpXA$

Role	Action	Reason	Transition
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vul and any mitigations (if no vendor exists)	Defense	**P**
non-vendor	Notify vendor	Coordination	**V**
any	Terminate any existing embargo	Exploit is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit refinement	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-

A.5 vfdPxa

Vendor is unaware of vulnerability. Fix is not ready. Fix has not been deployed. Public is aware of vulnerability. No exploits have been made public. No attacks have been observed.

Previous State(s): $vfdpxa$ (p.64)

Next State(s): $VfdPxa$ (p.76)

Desiderata met: $\mathbf{P} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{P}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 2, Zero Day Vulnerability Type 3

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: Unproven (U) or Not Defined (X). Embargo is no longer viable. Expect Vendor awareness imminently. SSVC v2 Exploitation: None. SSVC v2 Public Value Added: Ampliative. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: No. VEP does not apply. See Table A.5 for actions.

Table A.5: CVD Action Options for State $vfdPxa$

Role	Action	Reason	Transition
vendor	Pay attention to public reports	SA	**V**
non-vendor	Notify vendor	Coordination	**V**
any	Terminate any existing embargo	Vul is public	-
any	Monitor for exploit publication	SA	-
any	Monitor for attacks	SA	-
any	Publish mitigations	Defense	-
any	Escalate response priority among responding parties	Coordination	-
any	Discourage exploit publication until at least F	Limit attacker advantage	-

A.6 vfdPxA

Vendor is unaware of vulnerability. Fix is not ready. Fix has not been deployed. Public is aware of vulnerability. No exploits have been made public. Attacks have been observed.

Previous State(s): $vfdpxA$ (p.65)

Next State(s): $VfdPxA$ (p.77)

Desiderata met: $\mathbf{P} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{X} \prec \mathbf{A}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 2, Zero Day Vulnerability Type 3, Zero Day Attack Type 1, Zero Day Attack Type 2

Other notes: Attack success likely. Embargo is no longer viable. Expect Vendor awareness imminently. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Ampliative. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: No. VEP does not apply. See Table A.6 for actions.

Table A.6: CVD Action Options for State $vfdPxA$

Role	Action	Reason	Transition
any	Publish detection(s) for attacks	Detection	**P**
vendor	Pay attention to public reports	SA	**V**
non-vendor	Notify vendor	Coordination	**V**
any	Publish exploit code	Defense, Detection	**X**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit publication	SA	-
any	Monitor for additional attacks	SA	-
any	Publish mitigations	Defense	-
any	Escalate response priority among responding parties	Coordination	-

A.7 vfdPXa

Vendor is unaware of vulnerability. Fix is not ready. Fix has not been deployed. Public is aware of vulnerability. Exploit(s) have been made public. No attacks have been observed.

Previous State(s): $vfdpXa$ (p.66)

Next State(s): $VfdPXa$ (p.78)

Desiderata met: $\mathbf{P} \prec \mathbf{A}$, $\mathbf{X} \prec \mathbf{A}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 2, Zero Day Vulnerability Type 3, Zero Day Exploit Type 1, Zero Day Exploit Type 2

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H), functional (F), or proof of concept (P). Embargo is no longer viable. Expect Vendor awareness imminently. SSVC v2 Exploitation: PoC. SSVC v2 Public Value Added: Ampliative. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: No. VEP does not apply. See Table A.7 for actions.

Table A.7: CVD Action Options for State $vfdPXa$

Role	Action	Reason	Transition
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
vendor	Pay attention to public reports	SA	**V**
non-vendor	Notify vendor	Coordination	**V**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Exploit is public	-
any	Monitor for exploit refinement	SA	-
any	Monitor for attacks	SA	-
any	Publish mitigations	Defense	-
any	Escalate response priority among responding parties	Coordination	-

A.8 vfdPXA

Vendor is unaware of vulnerability. Fix is not ready. Fix has not been deployed. Public is aware of vulnerability. Exploit(s) have been made public. Attacks have been observed.

Previous State(s): $vfdpXA$ (p.67)

Next State(s): $VfdPXA$ (p.79)

Desiderata met: N/A

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 2, Zero Day Vulnerability Type 3, Zero Day Exploit Type 1, Zero Day Exploit Type 2, Zero Day Attack Type 1, Zero Day Attack Type 2

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H) or functional (F). Embargo is no longer viable. Expect Vendor awareness imminently. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Ampliative. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: No. VEP does not apply. See Table A.8 for actions.

Table A.8: CVD Action Options for State $vfdPXA$

Role	Action	Reason	Transition
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
vendor	Pay attention to public reports	SA	**V**
non-vendor	Notify vendor	Coordination	**V**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Exploit is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit refinement	SA	-
any	Monitor for additional attacks	SA	-
any	Publish mitigations	Defense	-
any	Escalate response priority among responding parties	Coordination	-

A.9 Vfdpxa

Vendor is aware of vulnerability. Fix is not ready. Fix has not been deployed. Public is unaware of vulnerability. No exploits have been made public. No attacks have been observed.

Previous State(s): $vfdpxa$ (p.64)

Next State(s): $VfdpxA$ (p.73), $VfdpXa$ (p.74), $VfdPxa$ (p.76), $VFdpxa$ (p.80)

Desiderata met: $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: N/A

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: Unproven (U) or Not Defined (X). CVSS 3.1 remediation level: Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T). Embargo continuation is viable. Embargo initiation may be appropriate. SSVC v2 Exploitation: None. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.9 for actions.

Table A.9: CVD Action Options for State $Vfdpxa$

Role	Action	Reason	Transition
vendor	Create fix	Defense	**F**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
non-vendor	Publish vul	Coordination, Motivate vendor to fix	**P**
any	Publish vul	Coordination, Motivate deployers to mitigate	**P**
any	Monitor for exploit publication	SA	-
any	Monitor for attacks	SA	-
any	Maintain vigilance for embargo exit criteria	SA	-
non-vendor	Encourage vendor to create fix	Coordination	-
any	Maintain any existing disclosure embargo	Coordination	-
any	Negotiate or establish disclosure embargo	Coordination	-
any	Discourage exploit publication until at least F	Limit attacker advantage	-

A.10 VfdpxA

Vendor is aware of vulnerability. Fix is not ready. Fix has not been deployed. Public is unaware of vulnerability. No exploits have been made public. Attacks have been observed.

Previous State(s): $vfdpxA$ (p.65), $Vfdpxa$ (p.72)

Next State(s): $VfdpXA$ (p.75), $VfdPxA$ (p.77), $VFdpxA$ (p.81)

Desiderata met: $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{X} \prec \mathbf{A}$

Zero Day Definitions Matched: Zero Day Attack Type 2, Zero Day Attack Type 3

Other notes: Attack success likely. CVSS 3.1 remediation level: Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T). Embargo is at risk. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.10 for actions.

Table A.10: CVD Action Options for State $VfdpxA$

Role	Action	Reason	Transition
vendor	Create fix	Defense	**F**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
non-vendor	Publish vul	Coordination, Motivate vendor to fix	**P**
any	Publish vul	Coordination, Motivate deployers to mitigate	**P**
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit publication	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to create fix	Coordination	-

A.11 VfdpXa

Vendor is aware of vulnerability. Fix is not ready. Fix has not been deployed. Public is unaware of vulnerability. Exploit(s) have been made public. No attacks have been observed.

Previous State(s): $Vfdpxa$ (p.72)

Next State(s): $VfdPXa$ (p.78)

Desiderata met: $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{X} \prec \mathbf{A}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Exploit Type 2, Zero Day Exploit Type 3

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H), functional (F), or proof of concept (P). CVSS 3.1 remediation level: Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T). Embargo is at risk. Expect Public awareness imminently. SSVC v2 Exploitation: PoC. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.11 for actions.

Table A.11: CVD Action Options for State $VfdpXa$

Role	Action	Reason	Transition
vendor	Create fix	Defense	**F**
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
non-vendor	Publish vul	Coordination, Motivate vendor to fix	**P**
any	Publish vul	Coordination, Motivate deployers to mitigate	**P**
any	Terminate any existing embargo	Exploit is public	-
any	Monitor for exploit refinement	SA	-
any	Monitor for attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to create fix	Coordination	-

A.12 VfdpXA

Vendor is aware of vulnerability. Fix is not ready. Fix has not been deployed. Public is unaware of vulnerability. Exploit(s) have been made public. Attacks have been observed.

Previous State(s): $VfdpxA$ (p.73)

Next State(s): $VfdPXA$ (p.79)

Desiderata met: $\mathbf{V} \prec \mathbf{P}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Exploit Type 2, Zero Day Exploit Type 3, Zero Day Attack Type 2, Zero Day Attack Type 3

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H) or functional (F). CVSS 3.1 remediation level: Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T). Embargo is at risk. Expect Public awareness imminently. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.12 for actions.

Table A.12: CVD Action Options for State $VfdpXA$

Role	Action	Reason	Transition
vendor	Create fix	Defense	**F**
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
non-vendor	Publish vul	Coordination, Motivate vendor to fix	**P**
any	Publish vul	Coordination, Motivate deployers to mitigate	**P**
any	Terminate any existing embargo	Exploit is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit refinement	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to create fix	Coordination	-

A.13 VfdPxa

Vendor is aware of vulnerability. Fix is not ready. Fix has not been deployed. Public is aware of vulnerability. No exploits have been made public. No attacks have been observed.

Previous State(s): $vfdPxa$ (p.68), $Vfdpxa$ (p.72)

Next State(s): $VfdPxA$ (p.77), $VfdPXa$ (p.78), $VFdPxa$ (p.84)

Desiderata met: $\mathbf{P} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{P}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 3

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: Unproven (U) or Not Defined (X). CVSS 3.1 remediation level: Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T). Embargo is no longer viable. SSVC v2 Exploitation: None. SSVC v2 Public Value Added: Ampliative. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.13 for actions.

Table A.13: CVD Action Options for State $VfdPxa$

Role	Action	Reason	Transition
vendor	Create fix	Defense	**F**
any	Terminate any existing embargo	Vul is public	-
any	Monitor for exploit publication	SA	-
any	Monitor for attacks	SA	-
any	Escalate vigilance for exploit publication or attacks	SA, Coordination	-
any	Publish mitigations	Defense	-
any	Ensure any available mitigations are publicized	Defense	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to create fix	Coordination	-
non-vendor	Escalate fix priority with vendor	Coordination	-
any	Discourage exploit publication until at least F	Limit attacker advantage	-

A.14 VfdPxA

Vendor is aware of vulnerability. Fix is not ready. Fix has not been deployed. Public is aware of vulnerability. No exploits have been made public. Attacks have been observed.

Previous State(s): $vfdPxA$ (p.69), $VfdpxA$ (p.73), $VfdPxa$ (p.76)

Next State(s): $VfdPXA$ (p.79), $VFdPxA$ (p.85)

Desiderata met: $\mathbf{P} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{X} \prec \mathbf{A}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 3, Zero Day Attack Type 2

Other notes: Attack success likely. CVSS 3.1 remediation level: Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T). Embargo is no longer viable. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Ampliative. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.14 for actions.

Table A.14: CVD Action Options for State $VfdPxA$

Role	Action	Reason	Transition
vendor	Create fix	Defense	**F**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish exploit code	Defense, Detection	**X**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit publication	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate vigilance for exploit publication or attacks	SA, Coordination	-
any	Publish mitigations	Defense	-
any	Ensure any available mitigations are publicized	Defense	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to create fix	Coordination	-
non-vendor	Escalate fix priority with vendor	Coordination	-

A.15 VfdPXa

Vendor is aware of vulnerability. Fix is not ready. Fix has not been deployed. Public is aware of vulnerability. Exploit(s) have been made public. No attacks have been observed.

Previous State(s): $vfdPXa$ (p.70), $VfdpXa$ (p.74), $VfdPxa$ (p.76)

Next State(s): $VfdPXA$ (p.79), $VFdPXa$ (p.86)

Desiderata met: $\mathbf{P} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{X} \prec \mathbf{A}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 3, Zero Day Exploit Type 2

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H), functional (F), or proof of concept (P). CVSS 3.1 remediation level: Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T). Embargo is no longer viable. SSVC v2 Exploitation: PoC. SSVC v2 Public Value Added: Ampliative. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.15 for actions.

Table A.15: CVD Action Options for State $VfdPXa$

Role	Action	Reason	Transition
vendor	Create fix	Defense	**F**
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Exploit is public	-
any	Monitor for exploit refinement	SA	-
any	Monitor for attacks	SA	-
any	Escalate vigilance for exploit publication or attacks	SA, Coordination	-
any	Publish mitigations	Defense	-
any	Ensure any available mitigations are publicized	Defense	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to create fix	Coordination	-
non-vendor	Escalate fix priority with vendor	Coordination	-

A.16 VfdPXA

Vendor is aware of vulnerability. Fix is not ready. Fix has not been deployed. Public is aware of vulnerability. Exploit(s) have been made public. Attacks have been observed.

Previous State(s): $vfdPXA$ (p.71), $VfdpXA$ (p.75), $VfdPxA$ (p.77), $VfdPXa$ (p.78)

Next State(s): $VFdPXA$ (p.87)

Desiderata met: N/A

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Vulnerability Type 3, Zero Day Exploit Type 2, Zero Day Attack Type 2

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H) or functional (F). CVSS 3.1 remediation level: Not Defined (X), Unavailable (U), Workaround (W), or Temporary Fix (T). Embargo is no longer viable. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Ampliative. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.16 for actions.

Table A.16: CVD Action Options for State $VfdPXA$

Role	Action	Reason	Transition
vendor	Create fix	Defense	**F**
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Exploit is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit refinement	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate vigilance for exploit publication or attacks	SA, Coordination	-
any	Publish mitigations	Defense	-
any	Ensure any available mitigations are publicized	Defense	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to create fix	Coordination	-
non-vendor	Escalate fix priority with vendor	Coordination	-

A.17 VFdpxa

Vendor is aware of vulnerability. Fix is ready. Fix has not been deployed. Public is unaware of vulnerability. No exploits have been made public. No attacks have been observed.

Previous State(s): $Vfdpxa$ (p.72)

Next State(s): $VFdpxA$ (p.81), $VFdpXa$ (p.82), $VFdPxa$ (p.84), $VFDpxa$ (p.88)

Desiderata met: $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: N/A

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: Unproven (U) or Not Defined (X). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo continuation is viable. Embargo initiation may be appropriate. Embargo initiation with careful consideration only. SSVC v2 Exploitation: None. SSVC v2 Public Value Added: Ampliative. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.17 for actions.

Table A.17: CVD Action Options for State $VFdpxa$

Role	Action	Reason	Transition
vendor, deployer	Deploy fix (if possible)	Defense	**D**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and fix details	Accelerate deployment	**P**
any	Monitor for exploit publication	SA	-
any	Monitor for attacks	SA	-
any	Maintain vigilance for embargo exit criteria	SA	-
any	Maintain any existing disclosure embargo	Coordination	-
any	Negotiate or establish disclosure embargo	Coordination	-
non-vendor	Encourage vendor to deploy fix (if possible)	Coordination	-
any	Scrutinize appropriateness of initiating a new disclosure embargo	Coordination	-

A.18 VFdpxA

Vendor is aware of vulnerability. Fix is ready. Fix has not been deployed. Public is unaware of vulnerability. No exploits have been made public. Attacks have been observed.

Previous State(s): $VfdpxA$ (p.73), $VFdpxa$ (p.80)

Next State(s): $VFdpXA$ (p.83), $VFdPxA$ (p.85), $VFDpxA$ (p.89)

Desiderata met: $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{X} \prec \mathbf{A}$

Zero Day Definitions Matched: Zero Day Attack Type 3

Other notes: Attack success likely. CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is at risk. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Ampliative. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.18 for actions.

Table A.18: CVD Action Options for State $VFdpxA$

Role	Action	Reason	Transition
vendor, deployer	Deploy fix (if possible)	Defense	**D**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vul and fix details	Accelerate deployment	**P**
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit publication	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to deploy fix (if possible)	Coordination	-

A.19 VFdpXa

Vendor is aware of vulnerability. Fix is ready. Fix has not been deployed. Public is unaware of vulnerability. Exploit(s) have been made public. No attacks have been observed.

Previous State(s): $VFdpxa$ (p.80)

Next State(s): $VFdPXa$ (p.86)

Desiderata met: $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{X} \prec \mathbf{A}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Exploit Type 3

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H), functional (F), or proof of concept (P). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is at risk. Expect Public awareness imminently. SSVC v2 Exploitation: PoC. SSVC v2 Public Value Added: Ampliative. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.19 for actions.

Table A.19: CVD Action Options for State $VFdpXa$

Role	Action	Reason	Transition
vendor, deployer	Deploy fix (if possible)	Defense	**D**
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vul and fix details	Accelerate deployment	**P**
any	Terminate any existing embargo	Exploit is public	-
any	Monitor for exploit refinement	SA	-
any	Monitor for attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to deploy fix (if possible)	Coordination	-

A.20 VFdpXA

Vendor is aware of vulnerability. Fix is ready. Fix has not been deployed. Public is unaware of vulnerability. Exploit(s) have been made public. Attacks have been observed.

Previous State(s): $VFdpxA$ (p.81)

Next State(s): $VFdPXA$ (p.87)

Desiderata met: $\mathbf{F} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{P}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Exploit Type 3, Zero Day Attack Type 3

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H) or functional (F). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is at risk. Expect Public awareness imminently. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Ampliative. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.20 for actions.

Table A.20: CVD Action Options for State $VFdpXA$

Role	Action	Reason	Transition
vendor, deployer	Deploy fix (if possible)	Defense	**D**
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vul and fix details	Accelerate deployment	**P**
any	Terminate any existing embargo	Exploit is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit refinement	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
non-vendor	Encourage vendor to deploy fix (if possible)	Coordination	-

A.21 VFdPxa

Vendor is aware of vulnerability. Fix is ready. Fix has not been deployed. Public is aware of vulnerability. No exploits have been made public. No attacks have been observed.

Previous State(s): $VfdPxa$ (p.76), $VFdpxa$ (p.80)

Next State(s): $VFdPxA$ (p.85), $VFdPXa$ (p.86), $VFDPxa$ (p.92)

Desiderata met: $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{P}$

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: Unproven (U) or Not Defined (X). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is no longer viable. SSVC v2 Exploitation: None. SSVC v2 Public Value Added: Ampliative. SSVC v2 Public Value Added: Limited. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.21 for actions.

Table A.21: CVD Action Options for State $VFdPxa$

Role	Action	Reason	Transition
deployer	Deploy fix	Defense	**D**
any	Publish exploit code	Defense, Detection, Accelerate deployment	**X**
any	Terminate any existing embargo	Vul is public	-
any	Monitor for exploit publication	SA	-
any	Monitor for attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
any	Promote fix deployment	Accelerate deployment	-

A.22 VFdPxA

Vendor is aware of vulnerability. Fix is ready. Fix has not been deployed. Public is aware of vulnerability. No exploits have been made public. Attacks have been observed.

Previous State(s): $VfdPxA$ (p.77), $VFdpxA$ (p.81), $VFdPxa$ (p.84)

Next State(s): $VFdPXA$ (p.87), $VFDPxA$ (p.93)

Desiderata met: $\mathbf{F} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{P}$, $\mathbf{X} \prec \mathbf{A}$

Other notes: Attack success likely. CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is no longer viable. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Ampliative. SSVC v2 Public Value Added: Limited. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.22 for actions.

Table A.22: CVD Action Options for State $VFdPxA$

Role	Action	Reason	Transition
deployer	Deploy fix	Defense	**D**
any	Publish detection(s) for attacks	Detection	**P**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit publication	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
any	Promote fix deployment	Accelerate deployment	-

A.23 VFdPXa

Vendor is aware of vulnerability. Fix is ready. Fix has not been deployed. Public is aware of vulnerability. Exploit(s) have been made public. No attacks have been observed.

Previous State(s): $VfdPXa$ (p.78), $VFdpXa$ (p.82), $VFdPxa$ (p.84)

Next State(s): $VFdPXA$ (p.87), $VFDPXa$ (p.94)

Desiderata met: $\mathbf{F} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{X} \prec \mathbf{A}$

Desiderata blocked: $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H), functional (F), or proof of concept (P). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is no longer viable. SSVC v2 Exploitation: PoC. SSVC v2 Public Value Added: Ampliative. SSVC v2 Public Value Added: Limited. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.23 for actions.

Table A.23: CVD Action Options for State $VFdPXa$

Role	Action	Reason	Transition
deployer	Deploy fix	Defense	**D**
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Exploit is public	-
any	Monitor for exploit refinement	SA	-
any	Monitor for attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
any	Promote fix deployment	Accelerate deployment	-

A.24 VFdPXA

Vendor is aware of vulnerability. Fix is ready. Fix has not been deployed. Public is aware of vulnerability. Exploit(s) have been made public. Attacks have been observed.

Previous State(s): $VfdPXA$ (p.79), $VFdpXA$ (p.83), $VFdPxA$ (p.85), $VFdPXa$ (p.86)

Next State(s): $VFDPXA$ (p.95)

Desiderata met: N/A

Desiderata blocked: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$

Other notes: Attack success likely. CVSS 3.1 Exploit Maturity: high (H) or functional (F). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is no longer viable. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Ampliative. SSVC v2 Public Value Added: Limited. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.24 for actions.

Table A.24: CVD Action Options for State $VFdPXA$

Role	Action	Reason	Transition
deployer	Deploy fix	Defense	**D**
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Exploit is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit refinement	SA	-
any	Monitor for additional attacks	SA	-
any	Escalate response priority among responding parties	Coordination	-
any	Promote fix deployment	Accelerate deployment	-

A.25 VFDpxa

Vendor is aware of vulnerability. Fix is ready. Fix has been deployed. Public is unaware of vulnerability. No exploits have been made public. No attacks have been observed.

Previous State(s): $VFdpxa$ (p.80)

Next State(s): $VFDpxA$ (p.89), $VFDpXa$ (p.90), $VFDPxa$ (p.92)

Desiderata met: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: N/A

Other notes: Attack success unlikely. CVSS 3.1 Exploit Maturity: Unproven (U) or Not Defined (X). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Do not initiate a new disclosure embargo, but an existing embargo may continue. Embargo continuation is viable. SSVC v2 Exploitation: None. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.25 for actions.

Table A.25: CVD Action Options for State $VFDpxa$

Role	Action	Reason	Transition
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vulnerability	Document for future reference	**P**
any	Publish vulnerability	Acknowledge contributions	**P**
any	Monitor for exploit publication	SA	-
any	Monitor for attacks	SA	-
any	Maintain vigilance for embargo exit criteria	SA	-
any	Maintain any existing disclosure embargo	Coordination	-

A.26 VFDpxA

Vendor is aware of vulnerability. Fix is ready. Fix has been deployed. Public is unaware of vulnerability. No exploits have been made public. Attacks have been observed.

Previous State(s): $VFdpxA$ (p.81), $VFDpxa$ (p.88)

Next State(s): $VFDpXA$ (p.91), $VFDPxA$ (p.93)

Desiderata met: $\mathbf{D} \prec \mathbf{P}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{P} \prec \mathbf{A}$, $\mathbf{X} \prec \mathbf{A}$

Zero Day Definitions Matched: Zero Day Attack Type 3

Other notes: Attack success unlikely. CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is at risk. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.26 for actions.

Table A.26: CVD Action Options for State $VFDpxA$

Role	Action	Reason	Transition
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vulnerability	Document for future reference	**P**
any	Publish vulnerability	Acknowledge contributions	**P**
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit publication	SA	-
any	Monitor for additional attacks	SA	-

A.27 VFDpXa

Vendor is aware of vulnerability. Fix is ready. Fix has been deployed. Public is unaware of vulnerability. Exploit(s) have been made public. No attacks have been observed.

Previous State(s): $VFDpxa$ (p.88)

Next State(s): $VFDPXa$ (p.94)

Desiderata met: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{P}$, $\mathbf{X} \prec \mathbf{A}$

Desiderata blocked: $\mathbf{P} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Exploit Type 3

Other notes: Attack success unlikely. CVSS 3.1 Exploit Maturity: high (H), functional (F), or proof of concept (P). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is at risk. Expect Public awareness imminently. SSVC v2 Exploitation: PoC. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.27 for actions.

Table A.27: CVD Action Options for State $VFDpXa$

Role	Action	Reason	Transition
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vulnerability	Document for future reference	**P**
any	Publish vulnerability	Acknowledge contributions	**P**
any	Terminate any existing embargo	Exploit is public	-
any	Monitor for exploit refinement	SA	-
any	Monitor for attacks	SA	-

A.28 VFDpXA

Vendor is aware of vulnerability. Fix is ready. Fix has been deployed. Public is unaware of vulnerability. Exploit(s) have been made public. Attacks have been observed.

Previous State(s): $VFDpxA$ (p.89)

Next State(s): $VFDPXA$ (p.95)

Desiderata met: $\mathbf{D} \prec \mathbf{P}$, $\mathbf{F} \prec \mathbf{P}$, $\mathbf{V} \prec \mathbf{P}$

Desiderata blocked: $\mathbf{P} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{X}$

Zero Day Definitions Matched: Zero Day Exploit Type 3, Zero Day Attack Type 3

Other notes: Attack success unlikely. CVSS 3.1 Exploit Maturity: high (H) or functional (F). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is at risk. Expect Public awareness imminently. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Precedence. SSVC v2 Report Public: No. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.28 for actions.

Table A.28: CVD Action Options for State $VFDpXA$

Role	Action	Reason	Transition
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations (if no active embargo)	Defense	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Publish vulnerability	Document for future reference	**P**
any	Publish vulnerability	Acknowledge contributions	**P**
any	Terminate any existing embargo	Exploit is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit refinement	SA	-
any	Monitor for additional attacks	SA	-

A.29 VFDPxa

Vendor is aware of vulnerability. Fix is ready. Fix has been deployed. Public is aware of vulnerability. No exploits have been made public. No attacks have been observed.

Previous State(s): $VFdPxa$ (p.84), $VFDpxa$ (p.88)

Next State(s): $VFDPxA$ (p.93), $VFDPXa$ (p.94)

Desiderata met: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: N/A

Other notes: Attack success unlikely. CVSS 3.1 Exploit Maturity: Unproven (U) or Not Defined (X). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is no longer viable. SSVC v2 Exploitation: None. SSVC v2 Public Value Added: Limited. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.29 for actions.

Table A.29: CVD Action Options for State $VFDPxa$

Role	Action	Reason	Transition
any	Terminate any existing embargo	Vul is public	-
any	Monitor for exploit publication	SA	-
any	Monitor for attacks	SA	-
any	Close case (unless monitoring for X or A)	No further action required	-

A.30 VFDPxA

Vendor is aware of vulnerability. Fix is ready. Fix has been deployed. Public is aware of vulnerability. No exploits have been made public. Attacks have been observed.

Previous State(s): $VFdPxA$ (p.85), $VFDpxA$ (p.89), $VFDPxa$ (p.92)

Next State(s): $VFDPXA$ (p.95)

Desiderata met: $\mathbf{D} \prec \mathbf{X}$, $\mathbf{F} \prec \mathbf{X}$, $\mathbf{P} \prec \mathbf{X}$, $\mathbf{V} \prec \mathbf{X}$

Desiderata blocked: $\mathbf{X} \prec \mathbf{A}$

Other notes: Attack success unlikely. CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is no longer viable. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Limited. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.30 for actions.

Table A.30: CVD Action Options for State $VFDPxA$

Role	Action	Reason	Transition
any	Publish detection(s) for attacks	Detection	**P**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit publication	SA	-
any	Monitor for additional attacks	SA	-
any	Close case (unless monitoring for X)	No further action required	-

A.31 VFDPXa

Vendor is aware of vulnerability. Fix is ready. Fix has been deployed. Public is aware of vulnerability. Exploit(s) have been made public. No attacks have been observed.

Previous State(s): $VFdPXa$ (p.86), $VFDpXa$ (p.90), $VFDPxa$ (p.92)

Next State(s): $VFDPXA$ (p.95)

Desiderata met: $\mathbf{D} \prec \mathbf{A}$, $\mathbf{F} \prec \mathbf{A}$, $\mathbf{P} \prec \mathbf{A}$, $\mathbf{V} \prec \mathbf{A}$, $\mathbf{X} \prec \mathbf{A}$

Desiderata blocked: N/A

Other notes: Attack success unlikely. CVSS 3.1 Exploit Maturity: high (H), functional (F), or proof of concept (P). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is no longer viable. SSVC v2 Exploitation: PoC. SSVC v2 Public Value Added: Limited. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.31 for actions.

Table A.31: CVD Action Options for State $VFDPXa$

Role	Action	Reason	Transition
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Exploit is public	-
any	Monitor for exploit refinement	SA	-
any	Monitor for attacks	SA	-
any	Close case (unless monitoring for A)	No further action required	-

A.32 VFDPXA

Vendor is aware of vulnerability. Fix is ready. Fix has been deployed. Public is aware of vulnerability. Exploit(s) have been made public. Attacks have been observed.

Previous State(s): $VFdPXA$ (p.87), $VFDpXA$ (p.91), $VFDPxA$ (p.93), $VFDPXa$ (p.94)

Next State(s): N/A

Desiderata met: N/A

Desiderata blocked: N/A

Other notes: Attack success unlikely. CVSS 3.1 Exploit Maturity: high (H) or functional (F). CVSS 3.1 remediation level: Temporary Fix (T) or Official Fix (O). Embargo is no longer viable. SSVC v2 Exploitation: Active. SSVC v2 Public Value Added: Limited. SSVC v2 Report Public: Yes. SSVC v2 Supplier Contacted: Yes. VEP does not apply. See Table A.32 for actions.

Table A.32: CVD Action Options for State $VFDPXA$

Role	Action	Reason	Transition
any	Draw attention to published exploit(s)	SA	**P**
any	Publish detection(s) for exploits	Detection	**P**
any	Publish detection(s) for attacks	Detection	**P**
any	Publish vul and any mitigations	Defense	**P**
any	Terminate any existing embargo	Vul is public	-
any	Terminate any existing embargo	Exploit is public	-
any	Terminate any existing embargo	Attacks observed	-
any	Monitor for exploit refinement	SA	-
any	Monitor for additional attacks	SA	-
any	Close case	No further action required	-

References/Bibliography

URLs are valid as of the publication date of this document.

[1] Prioritization to prediction volume 2: Getting real about remediation. Technical report, Cyentia Institute, LLC, 2019.

[2] Marcia Angell and Jerome P. Kassirer. The ingelfinger rule revisited. *New England Journal of Medicine*, 325(19):1371–1373, 1991. PMID: 1669838.

[3] William A Arbaugh, William L Fithen, and John McHugh. Windows of vulnerability: A case study analysis. *Computer*, 33(12):52–59, 2000.

[4] Ashish Arora, Jonathan P Caulkins, and Rahul Telang. Research note—sell first, fix later: Impact of patching on software quality. *Management Science*, 52(3):465–471, 2006.

[5] Ashish Arora, Chris Forman, Anand Nandkumar, and Rahul Telang. Competition and patching of security vulnerabilities: An empirical analysis. *Information Economics and Policy*, 22(2):164–177, 2010.

[6] Ashish Arora, Ramayya Krishnan, Rahul Telang, and Yubao Yang. An empirical analysis of software vendors' patch release behavior: impact of vulnerability disclosure. *Information Systems Research*, 21(1):115–132, 2010.

[7] Ashish Arora, Anand Nandkumar, and Rahul Telang. Does information security attack frequency increase with vulnerability disclosure? an empirical analysis. *Information Systems Frontiers*, 8(5):350–362, 2006.

[8] Ashish Arora and Rahul Telang. Economics of software vulnerability disclosure. *IEEE Security & Privacy*, 3(1):20–25, 2005.

[9] Ashish Arora, Rahul Telang, and Hao Xu. Optimal policy for software vulnerability disclosure. *Management Science*, 54(4):642–656, 2008.

[10] Leyla Bilge and Tudor Dumitraş. Before we knew it: an empirical study of zero-day attacks in the real world. In *Computer and communications security*, pages 833–844. ACM, 2012.

[11] Scott Bradner. RFC2119: Key words for use in RFCs to indicate requirement levels. `https://datatracker.ietf.org/doc/html/rfc2119`, 1997.

[12] Lawrence D Brown, T Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical science*, pages 101–117, 2001.

[13] Hasan Cavusoglu, Huseyin Cavusoglu, and Srinivasan Raghunathan. Efficiency of vulnerability disclosure mechanisms to disseminate vulnerability knowledge. *IEEE Transactions on Software Engineering*, 33(3):171–185, 2007.

[14] National Cyber Security Centre. Coordinated vulnerability disclosure: the guideline. Technical report, National Cyber Security Centre, Netherlands (NCSC-NL), October 2018.

[15] CERT Coordination Center (CERT/CC). CERT vulnerability data archive. `https://github.com/CERTCC/Vulnerability-Data-Archive`. Accessed: 2020-06-08.

[16] John T. Chambers and John W. Thomson. National Infrastructure Advisory Council's vulnerability disclosure framework: Final report and recommendations. `https://www.cisa.gov/publication/niac-vulnerability-framework-final-report`, 2004. Accessed: 2020-07-27.

[17] Steve Christey and Chris Wysopal. Responsible vulnerability disclosure process. https://tools.ietf.org/html/draft-christey-wysopal-vuln-disclosure-00, February 2002. Accessed: 2020-07-27.

[18] Melina Delkic. Ready, set, embargo. https://www.nytimes.com/2018/08/11/insider/embargoes-reporting.html, Aug 2018.

[19] Marcel Dreef, Peter Borm, and Ben Van der Genugten. Measuring skill in games: Several approaches discussed. *Mathematical methods of operations Research*, 59(3):375–391, 2004.

[20] Julia Driver. The History of Utilitarianism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2014 edition, 2014.

[21] Ryan Ellis, Keman Huang, Michael Siegel, Katie Moussouris, and James Houghton. *New Solutions for Cybersecurity*, chapter Fixing a Hole: The Labor Market for Bugs., pages 129–159. MIT Press, 2018.

[22] FIRST Ethics SIG. EthicsfIRST: Ethics for Incident Response and Security Teams. https://www.first.org/global/sigs/ethics/ethics-first, Dec 2019.

[23] Benjamin Eva. Principles of indifference. http://philsci-archive.pitt.edu/16041/, April 2019.

[24] Stefan Frei, Dominik Schatzmann, Bernhard Plattner, and Brian Trammell. Modeling the security ecosystem-the dynamics of (in) security. In *Economics of Information Security and Privacy*, pages 79–106. Springer, 2010.

[25] Atul Gawande. *The checklist manifesto: How to get things right*. Profile Books, 2011.

[26] Dan Goodin. Rise of "forever day" bugs in industrial systems threatens critical infrastructure. https://arstechnica.com/information-technology/2012/04/rise-of-ics-forever-day-vulnerabiliities-threaten-critical-infrastructure/, April 2012. Accessed: 2021-06-10.

[27] United States Government. Vulnerabilities Equities Policy and Process for the United States Government. https://trumpwhitehouse.archives.gov/sites/whitehouse.gov/files/images/External%20-%20Unclassified%20VEP%20Charter%20FINAL.PDF, Nov 2017. Accessed: 2021-02-22.

[28] Allen D Householder. Like nailing jelly to the wall: Difficulties in defining "zero-day exploit". https://insights.sei.cmu.edu/cert/2015/07/like-nailing-jelly-to-the-wall-difficulties-in-defining-zero-day-exploit.html, Jul 2015.

[29] Allen D Householder, Jeff Chrabaszcz, Trent Novelly, David Warren, and Jonathan M Spring. Historical analysis of exploit availability timelines. In *Workshop on Cyber Security Experimentation and Test*. USENIX, 2020.

[30] Allen D Householder, Garret Wassermann, Art Manion, and Chris King. The CERT guide to coordinated vulnerability disclosure. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Pittsburgh United States, 2017.

[31] ISO. Information technology — security techniques — vulnerability disclosure. Standard 29147:2018, International Organization for Standardization, Geneva, CH, October 2018.

[32] ISO. Information technology — security techniques — vulnerability handling processes. Standard 30111:2019, International Organization for Standardization, Geneva, CH, October 2019.

[33] Jay Jacobs, Sasha Romanosky, Idris Adjerid, and Wade Baker. Improving vulnerability remediation through better exploit prediction. *Journal of Cybersecurity*, 6(1), 09 2020. tyaa015.

[34] Jay Jacobs, Sasha Romanosky, Benjamin Edwards, Michael Roytman, and Idris Adjerid. Exploit prediction scoring system (EPSS). *arXiv preprint arXiv:1908.04856*, 2019.

[35] Shyamalendu Kandar. *Introduction to automata theory, formal languages and computation*. Always learning. Pearson, 1st edition edition, 2013.

[36] Patrick Larkey, Joseph B Kadane, Robert Austin, and Shmuel Zamir. Skill in games. *Management Science*, 43(5):596–609, 1997.

[37] Paul Simon Lewis. *The global vulnerability discovery and disclosure system: a thematic system dynamics approach*. PhD thesis, 2017.

[38] Andrew P Moore and Allen D Householder. Multi-method modeling and analysis of the cybersecurity vulnerability management ecosystem. In *37th International Conference of the System Dynamics Society*, 2019.

[39] Lily Hay Newman. Senators fear meltdown and spectre disclosure gave china an edge. `https://www.wired.com/story/meltdown-and-spectre-intel-china-disclosure/`, Jul 2018.

[40] NIST. National vulnerability database. `https://nvd.nist.gov`. Accessed: 2020-06-08.

[41] Forum of Incident Response and Security Teams. Common vulnerability scoring system v3.1: Specification document. `https://www.first.org/cvss/v3.1/specification-document`, 2019. Accessed: 2021-05-18.

[42] Forum of Incident Response and Security Teams. Guidelines and practices for multi-party vulnerability coordination and disclosure. `https://www.first.org/global/sigs/vulnerability-coordination/multiparty/guidelines-v1.1`, 2020. Accessed: 2020-07-27.

[43] Forum of Incident Response and Security Teams. Product security incident response team (psirt) services framework version 1.1. `https://www.first.org/standards/frameworks/psirts/psirt_services_framework_v1.1`, 2020. Accessed: 2021-05-17.

[44] Ivan Oransky. Why science news embargoes are bad for the public. `https://www.vox.com/science-and-health/2016/11/29/13765458/science-news-embargoes-bad-for-public`, Nov 2016.

[45] Andy Ozment and Stuart E Schechter. Milk or wine: does software security improve with age? In *USENIX Security Symposium*, volume 6, 2006.

[46] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[47] Lorenzo Pupillo, Afonso Ferreira, and Gianluca Varisco. Software vulnerability disclosure in Europe: Technology, policies and legal challenges. Technical report, Center for European Policy Studies (CEPS), 2018.

[48] Rapid7. Metasploit framework. `https://github.com/rapid7/metasploit-framework`. Accessed: 2020-06-08.

[49] Luta Security. Vulnerability coordination security model. `https://www.lutasecurity.com/vcmm`, 2020. Accessed: 2020-09-17.

[50] Offensive Security. Exploit db. `https://github.com/offensive-security/exploitdb`. Accessed: 2020-06-08.

[51] Erik Silfversten, William D Phillips, Giacomo Persi Paoli, and Cosmin Ciobanu. Economics of vulnerability disclosure. Technical report, European Union Agency for Network and Information Security (ENISA), 2018.

[52] Jonathan M Spring, Eric Hatleback, Allen D. Householder, Art Manion, and Deana Shick. Prioritizing vulnerability response: A stakeholder-specific vulnerability categorization. Technical report, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2019.

[53] Jonathan M Spring, Eric Hatleback, Allen D. Householder, Art Manion, and Deana Shick. Prioritizing vulnerability response: A stakeholder-specific vulnerability categorization. In *Workshop on the Economics of Information Security*, Brussels, Belgium, December 2020.

[54] Jonathan M Spring, Allen Householder, Eric Hatleback, Art Manion ad Madison Oliver, Vijay Sarvapalli, Laurie Tyzenhaus, and Charles Yarbrough. Prioritizing vulnerability response: A stakeholder-specific vulnerability categorization (version 2.0). Technical report, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2021.

[55] National Telecommunications and Information Administration. Software bill of materials. `https://www.ntia.gov/SBOM`. Accessed: 2021-05-18.

[56] Yang Xiao, Bihuan Chen, Chendong Yu, Zhengzi Xu, Zimu Yuan, Feng Li, Binghong Liu, Yang Liu, Wei Huo, Wei Zou, et al. Mvp: Detecting vulnerabilities using patch-enhanced vulnerability signatures. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1165–1182, 2020.

[57] Yifei Xu, Zhengzi Xu, Bihuan Chen, Fu Song, Yang Liu, and Ting Liu. Patch based vulnerability matching for binary programs. In *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 376–387, 2020.

Acronym List

CERT/CC	CERT® Coordination Center operated by Carnegie Mellon University
CVD	Coordinated Vulnerability Disclosure
CVSS	Common Vulnerability Scoring System, maintained by FIRST
DFA	Deterministic Finite Automaton
EoL	End-of-Life
FIRST	Forum of Incident Response and Security Teams
MPCVD	Multi-Party Coordinated Vulnerability Disclosure
NTIA	National Telecommunications and Information Administration
PSIRT	Product Security Incident Response Team
SBOM	Software Bill of Materials
SLE	Service Level Expectation
VEP	Vulnerability Equities Process
VCMM	Vulnerability Coordination Maturity Model
VM	Vulnerability Management