"Oh, what people would do with my knife?" Navigating the Dual-Use Dilemma in PoC Exploit Development, Disclosure, and Community Dynamics

Arwa Al Alsadi
TU Delft

Lorenz Kustosch
TU Delft

Lamya Alowain
Independent

Michel van Eeten
TU Delft

Carlos H. Gañán
TU Delft

Abstract

The cybersecurity landscape faces an escalating challenge as proof-of-concept (PoC) exploits transition from demonstrations to weaponized attacks within minutes of disclosure. While research has documented temporal dynamics and malicious deployment, a critical gap remains in understanding the human factors underlying PoC creation. Through semi-structured interviews with 16 PoC developers across diverse regions, we apply Expectancy-Value Theory to reveal PoC development as a complex motivational ecosystem where technical confidence, value assessments, and risk calculations intersect within dual-use tensions. We demonstrate that PoC development spans a continuum from crash demonstrations to weaponized exploits, shaped by multifaceted calculus rather than binary ethics. We identify three theoretical extensions: dual-use moral reasoning enabling responsibility externalization, dynamic value assessment where vendor behavior reshapes disclosure decisions, and identity navigation between ethical research and technical mastery. Vendor responsiveness, community dynamics, and legal constraints significantly influence disclosure strategies. PoC developers adopt risk-mitigation approaches when navigating tensions between security improvement and potential misuse, challenging binary conceptualizations of "responsible" versus "irresponsible" disclosure.

1 Introduction

The rapid weaponization of vulnerabilities poses a critical challenge for cybersecurity. Proof-of-Concept (PoC) exploits—intentionally developed code demonstrating vulnerability exploitability, ranging from simple crash demonstrations to fully weaponized implementations—initially requiring hours or even days to translate into active threats, are now being leveraged in alarmingly short timeframes.

Early reports indicated an average of one hour and seven minutes for PoC exploits to be weaponized after public disclosure [22], but recent evidence demonstrates a dramatic reduction, with instances of exploitation occurring within a mere 22 minutes of disclosure [45].

While existing research has examined the malicious use of public PoC exploits [22, 25, 45] and the rapid adoption of exploits by attackers, these studies primarily rely on observational data, focusing on the "what" and "when" of deployment. Less attention has been paid to the "who" and "why"—the motivations, processes, and decision-making of PoC developers creating and disseminating PoC. This leaves a significant gap in our understanding of the PoC lifecycle, specifically the factors influencing PoC creation, reporting, and publication decisions. Understanding this phenomenon requires examining not just the technical artifacts themselves, but the complex motivational landscape that drives their creation. To address this challenge, we apply Expectancy-Value Theory (EVT) as our theoretical lens, which provides a systematic framework for understanding how PoC developers navigate the inherent tensions between beneficial security research and potential harmful applications.

Existing research reveals three important gaps that our study addresses. First, while prior work examines hacker motivations broadly [20, 30, 46], no study has applied motivational frameworks specifically to PoC developers within contemporary dual-use disclosure ecosystems. Second, comprehensive examination of the full PoC lifecycle—including creation, validation, reporting, and publication—remains limited. Third, informal PoC ecosystems characterized by pseudonymity and legal uncertainty have been largely ignored, as research focuses on formal disclosure channels or observational studies.

Our work extends existing frameworks by demonstrating that PoC developer motivations shift dynamically in response to vendor behavior, with vendor responsiveness as a primary driver reshaping decisions over time. We also surface how institutional factors—CVD norms, liability concerns, weaponization timelines—shape contemporary decision-making in ways prior motivational research has not systematically addressed.

We fill these gaps by extending EVT with three key contributions: (i) dual-use moral reasoning, which enables PoC

developers to externalize responsibility for potential misuse; (ii) dynamic value assessment, where vendor behavior, disclosure outcomes, and community feedback continuously reshape PoC developers' subjective task values in real time; and (iii) identity navigation, emphasizing the ongoing negotiation between ethical research commitments and technical mastery identities. We also incorporate contextual factors outside traditional EVT—vendor relationship dynamics, community trust patterns, and legal constraints—that critically influence dual-use decision-making. We map these extensions to three research questions aligned with EVT components: ***(RQ1)*** What definitions and perceptions do PoC developers hold regarding PoC exploits? ***(RQ2)*** Which technical, ethical, and economic factors influence PoC creation, validation, and reporting? ***(RQ3)*** How do PoC developers navigate dual-use decision points in disclosure and publication strategies?

We conducted semi-structured interviews with 16 PoC developers from diverse regions and backgrounds. Our EVT-informed analysis reveals PoC development as dynamic negotiation among technical confidence, value assessments, and costs within dual-use contexts. This framework enabled us to make the following contributions:

- We deliver the first comprehensive analysis of the PoC lifecycle from their developers' perspectives, spanning simple crashes to weaponized exploits, guided by motivational calculus beyond technical skill or ethics.
- We extend EVT to dual-use contexts, introducing dual-use moral reasoning, dynamic value assessment, and identity navigation constructs that explain how PoC developers navigate competing beneficial and harmful applications.
- We identify key factors influencing PoC development across five stages (vulnerability identification, development, validation, reporting, publication), demonstrating that personal motivation, vendor response, and past experience critically shape dual-use decisions.
- We reveal that offensive-minded versus defensive-minded practitioners exhibit distinct motivational profiles, significantly influencing development and disclosure practices through different risk tolerance and value prioritization.

2 Related Work

Studies on Security Practices and Hacker Motivations. A substantial body of research uses qualitative and interview-based methods to understand the practices, motivations, and challenges of security professionals, ethical hackers, and testers. Votipka et al. [46] compared white-hat hackers and software testers, highlighting differences in knowledge, access, and motivation, and showing how PoC exploits are used to validate vulnerabilities when communication with developers is absent. Chng et al. [20] provided a comprehensive framework of hacker types and strategies, while Piao et al. [39] revealed how bug bounty participants collaborate informally to enhance discovery. Alomar et al. [7] examined real-world disclosure practices, underscoring persistent challenges in vendor–researcher coordination.

Building on this, research has also investigated bug bounty programs and vulnerability disclosure channels. Studies highlight the economic incentives and dilemmas hackers face in navigating institutional programs [1,10,36]. Walshe and Simpson [48] assessed coordinated vulnerability disclosure (CVD) programs and proposed improvements to foster more trustful interactions. The *Psychology of Hackers* preprint [30] further examined the psychological drivers behind vulnerability discovery. More recently, Chaliasos et al. [17] evaluated automated tools in the decentralized finance (DeFi) ecosystem, finding that state-of-the-art analyzers would have prevented only a small fraction of high-profile attacks, highlighting gaps between tool capabilities and practitioner needs.

Ethnographic work has examined hacker communities as sites of identity and moral reasoning. Coleman's foundational study [21] and recent historical analyses [47] underscore how values shape hacking. Foundational typology work [14, 40] established psychological frameworks for hacker behavior, including tool developers. We build upon this by focusing on PoC development within contemporary disclosure ecosystems. These perspectives situate PoC exploit development as both a technical task and one embedded in contested moral economies and dual-use dilemmas.

PoC Exploits and Exploitation Timelines. Another line of work examines the lifecycle of PoC exploits and their weaponization in the wild. Nayak et al. [37] showed that some vulnerabilities are rapidly targeted while others remain dormant. Alrawi et al. [8] documented how IoT malware campaigns quickly integrate public exploits. More recent measurements confirm that adversaries may weaponize PoC within minutes of publication [22], underscoring the urgency of timely patching. Al Alsadi et al. [2, 3] provided large-scale analyses of exploit timelines, showing that while many exploits appear quickly, others surface years later or even prior to CVE publication date, complicating assumptions about disclosure. Related historical analyses [29,43] emphasize how exploit availability reshapes patching priorities and attacker decision-making.

Economics and Dual-Use Dilemmas. Economic perspectives on vulnerability exploitation highlight adversaries' cost–benefit reasoning. Allodi's work [5, 6] shows that attackers prioritize low-cost, high-impact vulnerabilities and adopt new exploits selectively. At the same time, dual-use dilemmas emerge as tools circulate across legitimate and malicious domains. Kang et al. [32] demonstrate how instruction-following LLMs amplify dual-use risks by enabling standard security attacks at scale. Silic [42] similarly analyzed the organizational

dilemmas of adopting dual-use open source security tools, showing both defensive benefits and heightened risks. Ethical scholarship on dual-use has a long history in the biological sciences, where debates on governance emphasize the tension between advancing knowledge and mitigating misuse [35]. Our study extends this dual-use lens to the socio-technical practices of PoC exploit developers, who often operate under pseudonymity and legal uncertainty, in contrast to more institutionalized bug bounty ecosystems.

Motivational Theories in Security Behavior. Motivation theories from psychology provide additional explanatory power for security behavior. Expectancy-Value Theory (EVT) [24, 49] posits that task engagement depends on expectations of success ("Can I succeed at this task?") and the subjective value of the task ("Is the task worth doing?"), which includes intrinsic, attainment, and utility values, as well as costs. EVT has been applied in security contexts such as password adoption [31] and mobile identity protection [4]. Chen et al. [19] further demonstrated its usefulness in phishing interventions, revealing how expectancy and value factors shape employee engagement. Other domains have also adopted EVT, such as studies on gender-related academic and occupational goals [38].

While EVT has been successfully applied to defensive security behaviors [4, 19, 31], no previous research has applied motivational frameworks to offensive security contexts such as PoC exploit development. We argue that EVT is well-suited to capture developers' decision-making, as it directly addresses *why* individuals take on tasks, *what* values they assign to them, and *how* costs shape engagement. This complements economic and ethical accounts of dual-use by foregrounding the situated motivations of actors.

Our study addresses this gap by extending EVT to capture dual-use decision-making, where the same activities produce both beneficial and harmful outcomes. We recognize that PoC developers have been studied as part of the broader hacker spectrum in prior work [20, 46]. Our contribution provides specific focus on this population within contemporary disclosure ecosystems and demonstrates how vendor interactions dynamically reshape developer motivations over time—a temporal dimension absent from prior motivational research in security contexts.

Finally, prior analyses of dual-use dilemmas at policy and organizational levels—such as work on economic analysis of dual-use research governance [23] and organizational study [42]—demonstrate the importance of expectancy–value reasoning beyond individual actors. Our study, by focusing on PoC exploit developers, fills a gap in connecting motivational theory to the dual-use challenges of PoC exploit circulation.

3 Methodology

We employed qualitative semi-structured interviews to explore the practices, motivations, and decision-making processes of PoC exploit developers. We recruited 16 participants from diverse geographic and professional backgrounds, piloted our interview protocol with psychology-trained collaborators, conducted remote interviews with rigorous verification procedures, and analyzed transcripts using hybrid inductive-deductive coding with strong inter-coder reliability.

3.1 Operational Definition of PoC Exploits

We define Proof-of-Concept (PoC) exploits as intentionally developed code demonstrating vulnerability exploitability, spanning simple crash demonstrations to fully weaponized exploits. This definition emphasizes deliberate exploit construction and excludes automated fuzzer outputs or raw crashes lacking purposeful development. Our focus is on exploits created through human decision-making processes, regardless of sophistication level. This operational definition aligns with how participants conceptualized their work and enabled examination of the full range of motivations and practices in PoC development.

3.2 Recruitment and Screening

We employed multi-channel recruitment to access this hard-to-reach population. We began by distributing a registration form via social media platforms, mailing lists, and direct professional outreach to collect participants' backgrounds and PoC development experience. Our inclusion criterion required prior experience developing at least one proof-of-concept exploit from start to finish. Beyond form-based recruitment, we utilized: (i) professional outreach via social media, mainly LinkedIn, to potential participants matching our criteria; (ii) security conference networking where the lead author identified and approached candidates; (iii) recommendations from academic colleagues; and (iv) snowball sampling where participants connected us to others in their networks. From 33 form responses plus additional professional contacts, 16 participants were selected after screening and eligibility confirmation. Personalized invitations detailed the study's purpose, scope, requirements, anonymity assurances, and emphasized voluntary participation with withdrawal rights at any time. Only one participant had a pre-existing work relationship with the research team; all others were identified through professional networking initiated for this study. Participants spanned North America, South America, Europe, Africa, and Asia, providing geographic diversity to assess whether motivational patterns reflected cultural specificity or shared institutional factors (CVD norms, legal frameworks). Thematic saturation was reached well within this sample size [18]. While professional network-based recruitment may introduce sampling bias toward communicative developers, this approach was necessary given the population's privacy concerns and legal sensitivities.

3.3 Pilot Interviews and Protocol Development

Three pilot interviews (one in-person, two remote) tested and refined our protocol. The lead researcher conducted these pilots with a second author—who has a background in psychology—present during the sessions to observe interview flow, evaluate question clarity, and ensure the environment encouraged open responses. Critically, the interview protocol evolved iteratively during pilot testing rather than being predetermined by EVT categories. Pilot sessions with structured feedback revealed that generic motivation questions needed reframing to capture specific PoC developer tensions. The refined protocol (see Appendix A) addresses EVT constructs implicitly through open-ended questions about confidence, benefits, risks, and outcome expectations, avoiding theoretical jargon that would artificially constrain responses. Post hoc application of EVT during analysis ensured the framework matched empirical patterns rather than predetermining them. This approach addressed concerns that EVT was imposed on rather than emerged from the data.

3.4 Interview Procedure

We conducted semi-structured interviews via video or audio conferencing, lasting 45-60 minutes (two extended to approximately two hours). The lead researcher conducted all interviews except one conducted in Spanish by a native-speaking co-author to accommodate participant preference. This transcript was translated using Argos Translate Python library [9], following established methodology [12]. The offline neural machine translation tool ran entirely on local resources to maintain GDPR compliance and data protection requirements. Two interviews included third-party interpreters at participants' request to facilitate communication. Interpreters were strictly limited to translation and clarification roles, with clear instructions ensuring only participant perspectives were captured. All participants provided informed consent prior to sessions, with study details reiterated at interview commencement. Our outreach and consent procedures emphasized that participation was entirely voluntary and that participants could withdraw at any time without consequence. Interviews followed a structured format: warm-up questions covering background, experience, and motivations (Questions 1-8), followed by core questions on PoC development processes (Questions 10-16). The complete interview protocol appears in Appendix A.

3.5 Verification and Credibility

To address self-enhancement bias, the tendency for participants to overstate their achievements or capabilities, we implemented converging verification checks while respecting anonymity and legal constraints. Many participants were corroborated through public records (CVE identifiers, vendor security advisories listing them as reporters). Others voluntarily demonstrated artifacts during interviews via screen sharing (triggering exploit scripts, debugger sessions showing exploitation, reproduction steps) or shared links to their public PoCs on GitHub or Exploit-DB aligning with their narratives. For participants bound by NDAs or ongoing vendor coordination, we relied on internal consistency of accounts and depth of technical detail provided. While these converging indicators enhance confidence in participants' accounts and reduce fabrication risk, they do not entirely eliminate self-enhancement bias or broader self-report limitations.

3.6 Data Processing and Analysis

The lead author manually transcribed all interviews verbatim using secure local software with meticulous accuracy review. Comprehensive anonymization removed identifying information including names, organizations, technical implementations, geographic indicators, and temporal markers preventing re-identification. Participants were invited to review their transcripts and quoted material for accuracy and comfort, reducing misinterpretation risk and enhancing validity. We imported anonymized transcripts into Atlas.ti [11] for qualitative analysis. Our hybrid inductive-deductive coding [26] enabled both data-driven code emergence and theory-informed EVT analysis. Two authors independently coded all 16 transcripts, iteratively refining the codebook through consensus meetings that reconciled discrepancies and refined definitions based on EVT constructs and emergent themes. To assess inter-coder reliability, we calculated Krippendorff's alpha [16], achieving strong consistency ($\alpha = 0.857$), exceeding accepted thresholds for qualitative security research [33]. This high reliability was facilitated by shared codebook use and consistent application of predefined coding themes (see Appendix C).

3.7 Participant Demographics

Our 16 participants included penetration testers, red and blue teamers, independent researchers, consultants, and in-house product security professionals; several held leadership positions (founders, CEOs). We recorded each participant's primary and secondary roles; years of experience (current role, overall security, PoC development); number of vulnerabilities discovered; and number of PoCs developed (as each participant defined PoCs). Additional self-reported data covered public PoC count, development frequency, skill level, and targeted software types. All participants identified as male despite targeted outreach to women in cybersecurity, reflecting broader gender disparities in the community [27]. Full demographic breakdown appears in Appendix B.

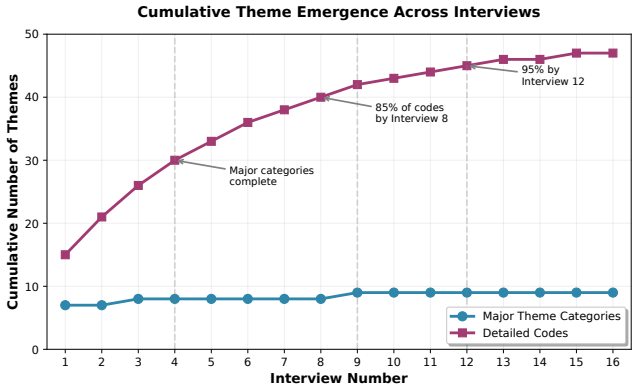

Figure 1: Cumulative theme emergence across 16 interviews.

3.8 Limitations

Our study has several limitations common to exploratory qualitative research [1, 7, 46]. First, participant recall may be incomplete or biased when retrospectively describing complex, expert-level tasks [46]. We mitigated this through hierarchical structuring in our interview protocol to prompt comprehensive responses [44]. Second, participants may have adjusted responses to influence how they were perceived (social desirability bias) or overstated their achievements (self-enhancement bias) [46]. While we created a non-judgmental atmosphere and implemented verification procedures (Section 3.5), we cannot entirely eliminate these biases. Third, explicit focus on PoC development may have attracted participants with particular experiences or interests, introducing selection bias. Additionally, individuals concerned about privacy or intellectual property may have opted not to participate. We partially mitigated this through diverse recruitment channels and screening for variation in experience and geography. Gender imbalance in our all-male sample, despite targeted outreach efforts, reflects broader cybersecurity community disparities [27]. Fourth, our sample of 16 participants demonstrates thematic saturation. Figure 1 shows cumulative theme emergence: all major thematic categories (blue circles) appeared by Interview 4, with 85% of detailed codes (purple squares) identified by Interview 8 and 95% by Interview 12. Interviews 13–16 yielded primarily refinements (e.g., explicit cultural stigmatization articulation) rather than new categories, providing empirical evidence of data sufficiency [28]. Finally, findings may not generalize beyond our sample, though our focused recruitment provides strong direction for future work in this niche practitioner group [46].

4 Understanding PoC Developer Perspectives and Decision-Making

We apply EVT to analyze our semi-structured interview data and reveal how PoC exploit developers navigate the inherent dual-use dilemma in their work. PoC development spans a continuum from simple crash proofs to fully automated, weaponized exploits. Higher sophistication—automation, weaponization, code-centricity—amplifies both security research value and misuse risk. Figure 2 presents our adapted EVT framework, highlighting dual-use decision points where developers balance competing motivations, integrating *Traditional EVT constructs* (yellow), *Extended EVT contributions* (pink), and *Contextual factors outside EVT* (blue).

We first examine how PoC developers define and perceive PoC exploits (RQ1), then identify the technical, ethical, economic, and contextual factors that shape PoC creation, validation, and reporting (RQ2), and finally explore how PoC developers negotiate dual-use decision points in their disclosure and publication strategies (RQ3). Notably, our EVT-based analysis reveals a phenomenon we term *motivational ambivalence*, where the very factors—such as intrinsic satisfaction from technical mastery—that drive thorough exploit development also heighten the potential for misuse (see Table 1).

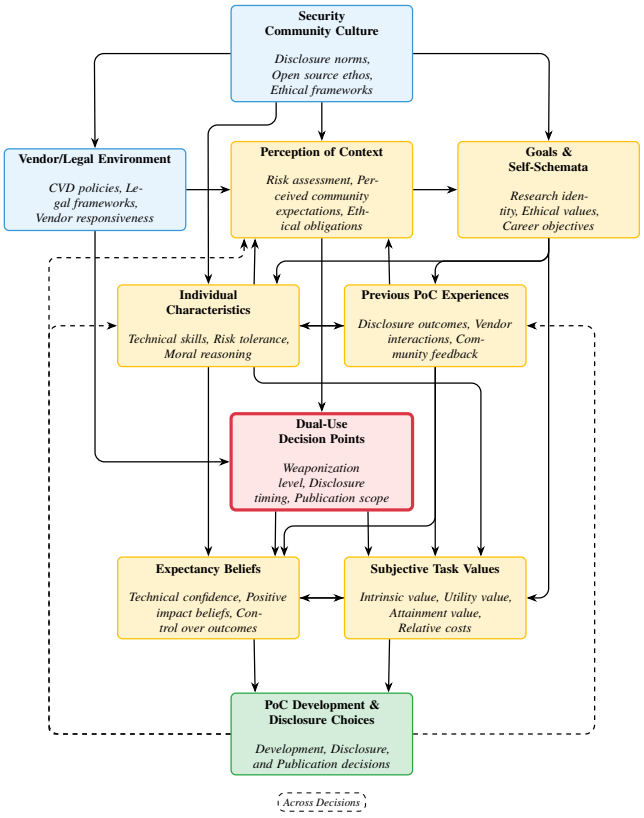

Figure 2: The Expectancy-Value Model for PoC Development Dual-Use Decisions. *(Adapted from [24])*

Table 1: Motivational Factors in PoC Development Dual-Use Decisions Based on EVT Analysis

EVT Component	Encouraging Dual-Use Risk	Discouraging Dual-Use Risk
Expectancy Beliefs	High technical confidence, Patch diffing success, AI-assisted development	Task difficulty uncertainty, Limited system knowledge, Tool/resource constraints
Intrinsic Value	Intellectual satisfaction, Technical mastery, Weaponization challenge	Preference for defensive applications, Lack of interest in harmful applications
Utility Value	Personal advancement, Financial gain, Skill demonstration, Security improvement	Low perceived benefit, Vendor assistance, Ecosystem benefits
Attainment Value	Technical mastery identity, Community recognition	Core ethical values, Professional identity
Cost Factors	Low perceived legal risk, Anonymity protection, Vendor unresponsiveness	High legal/reputational risk, Moral responsibility burden, Potential misuse guilt
Extended EVT:		
Moral Reasoning	Tool neutrality beliefs, Responsibility deflection	Strong moral framework, Harm prevention
Dynamic Value Assessment	Vendor unresponsiveness, disclosure/publication outcomes, and community feedback	Vendor responsiveness, Coordinated disclosure benefits
Identity Navigation	Personal gain, Financial gain	Ethical identity, Professional reputation
Outside EVT:		
Vendor Relationships	Unresponsive vendors, Poor communication	Responsive vendors, Strong CVD frameworks, Clear communication and timelines
Community Dynamics	Anonymous collaboration benefits, Knowledge sharing pressure	Trust erosion concerns, Community accountability
Cultural/Legal Context	Legal gray areas, Weak enforcement, Cultural acceptance	Strong legal frameworks, Cultural disclosure norms, Clear consequences

4.1 How PoC Developers Define and Perceive PoC Exploits (RQ1)

PoC developers vary widely in what they consider a valid PoC, ranging from simple crash demonstrations to fully weaponized exploit code. Their definitions reflect not only technical benchmarks but also personal standards for originality, sophistication, and intended use. Underlying these definitions are key motivational factors: beliefs about their own technical capabilities and likelihood of success, the intrinsic satisfaction of demonstrating mastery, the utility of PoC for security improvement, the role of PoC work in their professional identity, and the costs they associate with development and disclosure. Together, these factors shape how PoC developers define, prioritize, and engage with PoC exploit creation (highlighted in yellow in Figure 2).

4.1.1 The PoC Exploit Definition Spectrum

PoC developers' beliefs about their technical abilities and expectations of success shape their position on what a PoC exploit is and should do, influencing their development strategies and publication choices.

Perceived Technical Ability: Several PoC developers explicitly connected PoC creation to demonstrating technical understanding and validating their expertise. For instance, P12 mentioned *"because it proves that you understand the vulnerability. Ego plays a role here."* This reflects how self-concept of ability drives behavior choices where PoC developers with high technical confidence pursue challenging targets to validate and display expertise.

PoC developers demonstrated varying perspectives on legitimate PoC targets, with some insisting that valid PoC should exclusively target self-discovered or zero-day vulnerabilities. As P10 explained: *"I'd say it's when you find a vulnerability on your own, like a zero day,"* while P13 similarly stated: *"I think it's for the zero day vulnerability."* This exclusivity preference reflects a high self-concept of technical ability, where PoC developers prioritize original discovery over adaptation of known vulnerabilities. On the other hand, other PoC developers maintained that PoC can target known vulnerabilities discovered by others and still be considered valid as long as they demonstrate technical effectiveness, indicating different self-concept thresholds for meaningful technical contribution.

P8 similarly described preferences for *weaponized proof of concept* creation, indicating how a high self-concept of ability leads to more sophisticated development goals. The intrinsic satisfaction derived from technical mastery serves as a powerful motivator independent of external rewards.

PoC developers systematically adapt their approaches based on perceived expertise in different vulnerability domains. P12 explained their systematic approach: *"If [a vulnerability] is publicly published, I use Shodan to get the CVE. If that CVE has no PoC, I develop one."* This demonstrates how domain knowledge confidence (expectancy belief) drives efficient targeting strategies.

Task Difficulty Perceptions and Success Expectations: PoC developers demonstrated preferences for *known vulnerabilities* with or without existing PoC, viewing these as optimal targets balancing challenge with success probability. P2 described this approach as: *"So we look at the differences*

between the old version and the new version. And that often requires some reverse engineering skills. Based on what the vendor changed in the new version, we can deduce what the vulnerability was in the old version and write code to verify that those vulnerabilities still persist. In the new version, there might be a single line change that reflects the patch." This "patch diffing" approach used by reverse engineering silent patching reflects sophisticated task difficulty assessment—leveraging vendor patches as roadmaps reduces perceived difficulty while maintaining intellectual challenge. Such approaches maximize expectation of success while preserving intrinsic value through technical problem-solving.

In addition, pursuing existing vulnerabilities often leads to discovering new vulnerabilities. P4 explained: *"For example, in the case of a PDF reader, I noticed several vulnerabilities published recently, so I examined if one product or multiple products are affected. The larger the user base, the more likely you will find an easy target."*

For *unknown vulnerabilities*, PoC developers often adopt a trial-and-error approach. P12 mentions that he *"downloads, installs, and fuzzes the software,"*, while P15 tailors his methods based on access to physical devices, firmware, or mere specifications. Access to necessary tools and knowledge significantly affects PoC developers' willingness to pursue specific vulnerabilities, with deep system knowledge reducing perceived task difficulty and increasing expectation of success.

Prior experience guides both target selection and the choice of tools and programming languages for PoC development. A strong preference for Python is common due to its simplicity, extensive libraries, and compatibility with AI-driven methods. P11 noted: *"AI agents can work with Python extremely efficiently."* Similarly, P1 stated: *"Just use AI to make a good starting point."* PoC developers also utilize languages such as .NET, C#, bash, and Go, selecting the appropriate tools based on target specifics and operational requirements.

AI integration enhances expectancy beliefs by boosting PoC developers' technical confidence and perceived capability to handle complex vulnerabilities. The availability of AI-assisted development tools strengthens the self-concept of ability, as PoC developers feel more capable of creating sophisticated PoC regardless of their initial expertise level, which reduces effort costs. This technological augmentation of expectancy beliefs creates a feedback loop where increased confidence leads to more ambitious targeting and development approaches, while simultaneously reducing the perceived difficulty of moving toward more weaponized implementations.

PoC developer expectations for PoC sophistication varied significantly, with some requiring full automation and weaponization capabilities. P2 emphasized: *"Only consider it as a proof concept exploit if it's an automated code that anyone can run and get the same results."* This automation standard reflects high expectancy beliefs about technical outcomes, PoC developers confident in creating reliable, weaponized tools set higher success thresholds. P8 described the ideal as a "0-click" solution where "typing in an IP address" automatically yields complete system control. Such weaponization expectations demonstrate how a high self-concept of ability leads to more sophisticated development goals, potentially increasing dual-use risks through enhanced exploit capabilities. P6 described the evolution from basic demonstration to automated tool: initially demonstrating a vulnerability, but a robust PoC should *"eventually automate all steps so it can be reliably handed over to, for example, a red team or the vendor for remediation."*

4.1.2 Motivational Dynamics in PoC Development

PoC exploit development sits at a crossroad in cybersecurity, embodying a dual-use dilemma where the same tools that drive security research and remediation can also be weaponized for malicious purposes. This creates a complex motivational landscape for PoC developers, who must constantly balance technical ambition, values, ethical considerations, and risk management.

Intrinsic Value and Craftsmanship: PoC developers derive deep intrinsic satisfaction from crafting PoC exploits, emphasizing the paramount importance of executable code over abstract descriptions. As one participant (P8) succinctly put it, *"the most fun is to write a weaponized Proof of Concept."* This code-centric philosophy positions functional code as the ultimate demonstration of mastery. Reflecting this view, P12 stated: *"For me only the code counts... the perfect PoC is a code base."* Similarly, P7 highlighted that a PoC is *"any piece of code... tailored to exploit a vulnerability... especially if it is novel."*

This orientation motivates them to build PoC from scratch, particularly for new vulnerabilities, ensuring technical rigor and security integrity. P12 described their workflow for web exploits: *"I start with manual exploitation, then replicate it in Python, escalating to remote code execution."* The drive for ethical craftsmanship also manifests in efforts to minimize disruption: P5 embedded sleep commands into an SQL injection PoC to avoid crashing the system, and P3 praised *"thinking outside the box"* to achieve elegant yet non-disruptive demonstrations.

Utility Focus: Beyond personal fulfillment, many PoC developers prioritize utility — the benefit their work brings to cybersecurity. P2 framed this value as, *"...cybersecurity should be a fundamental right... PoC prove risks and motivate mitigation."* PoC developers often favor responsible disclosure to vendors or authorities. For instance, P13 supports full disclosure to national authorities, while P11 advocates delaying public release: *"...don't release... instantly... but eventually it's valuable."*

PoC are seen as tools for remediation. P1 emphasized, *"PoC are purely demonstrations to get things fixed,"* and P15 noted they help others *"reproduce the problem."* This approach balances knowledge sharing with reducing potential misuse risks.

Identity and Recognition: Disclosure decisions are strongly tied to complex professional identities and ethical considerations. Many PoC developers see themselves as responsible contributors rather than as seekers of notoriety. P11 expressed support for openness tempered by caution: *"I'm for openness, but... wait for fixes... to help defenders."* Cultural and regional contexts influence these decisions; for example, P16 noted that some prefer limited disclosure to avoid reputational damage.

Cost and Moral Challenges: PoC developers contend with significant costs, including legal risks, reputational concerns, and psychological burdens. P14 abstains from working on zero-day PoC due to legal restrictions. P1 related an experience involving misuse: *"We made a full exploit... it ended up in the wrong hands... several organizations got affected... I distanced myself after that."*

Concerns about irresponsible disclosure are common. P2 warned: *"...giving weapons to others who can't handle them responsibly is irresponsible."* Many take precautions to safeguard PoC, such as maintaining them locally with strict security measures. Economic incentives complicate decisions: P12 admitted to selling PoC in the past but now pursues responsible disclosure.

Takeaway 1

PoC development spans a continuum from simple crash demos to fully weaponized exploits, driven by complex motivations including technical mastery, ethical reflection, and risk negotiation. PoC developers' decisions are understood as dynamic calculations balancing their confidence (expectancy), perceived value (utility, intrinsic, attainment), and costs (legal, moral).

4.2 Factors Driving PoC Development Decisions (RQ2)

This section examines the key factors shaping PoC development decisions, extending EVT to capture the unique challenges of dual-use technologies. Central to this extension is the concept of *dual-use decision points* (highlighted in red in Figure 2), where PoC developers weigh competing technical, ethical, and contextual considerations.

Our extended EVT framework comprises three components: *Moral Reasoning*, where PoC developers navigate ethical reflections on tool neutrality and responsibility; *Dynamic Value Assessment*, reflecting how PoC developers continuously adjust their valuation of tasks based on vendor responsiveness, disclosure outcomes, and community feedback; and *Identity Navigation*, describing ongoing negotiation between competing professional identities in the face of dual-use tensions. Together, these elements illuminate the complex, context-sensitive calculus guiding PoC creation, validation, and disclosure decisions..

4.2.1 Moral Reasoning About Tool Neutrality

PoC developers grapple with moral responsibility stemming from the dual-use nature of their work, which introduces psychological costs beyond typical effort or time constraints. They often adopt a moral reasoning stance that views PoC as neutral tools—akin to knives—where the ethical implications depend on how others use them. This perspective helps PoC developers manage ethical tensions and sustain engagement despite potential misuse.

The philosophical reasoning about *tool neutrality* represents a distinct form of ethical reasoning: while a PoC exploit can be adapted for harmful purposes, the tool itself is inherently neutral, its effect determined by how it is employed, which was analogized by P9 as: *"Well, releasing an exploit have their risk. It's like a knife. It's like it's not good or bad; and a knife is just a tool. You can use the knife to feed your family, or you can use the knife to kill somebody else... So you the person who crafted the knife... oh, what people would do with my knife."*

4.2.2 PoC Dynamic Value Assessment

PoC development involves ongoing recalibration of risk and benefit evaluations, with PoC developers' subjective task values shifting dynamically in response to external influences such as vendor responsiveness, disclosure outcomes, and community feedback.

Several participants highlighted a preference for withholding public disclosure of PoC until vulnerabilities have been patched. For example, participant P10 expressed reluctance to share PoC prematurely: *"I'm not really in favor of publicly sharing the PoC unless I know the vulnerability has been patched. The people running the software were given a few months to patch, so I only publish after they have had ample time to update."*

On the other hand, participant P1 raised concerns about the limits of timing in mitigating risk, noting: *"If you look at the timelines when a proof of concept is published and when it gets abused, those timelines are pretty interesting. A patch does not mean everyone is protected immediately."* Some PoC developers practice controlled disclosure, delaying publication of weaponized PoC for extended periods. Participant P8 explained: *"If I have a weaponized proof of concept, I do not share it with the public right away. Maybe after a year or two, when everything is fixed, then I release it, often via my*

website."

Participant P12 provided a critical view of how vendor transparency impacts risk assessment: *"If there's a public exploit code... some vendors don't admit what the patch does. But if the vendor is fast and transparent, having public PoC balances the outcome — attackers' odds are about 50/50."* This dynamic process reflects how PoC developers continuously update their value calculations based on vendor behavior and disclosure context. Moreover, participants described a maturation from personal or opportunistic motives to more collaborative and career-oriented ones. P12 reflected on this evolution: *"Now I don't do that [selling PoC] anymore; I've grown and consider reporting publicly. Maybe it could even change my career path — like working for Facebook."*

4.2.3 Balancing Professional Identity with Ethical Responsibilities

While traditional EVT's attainment value assumes straightforward alignment between activities and identity, dual-use contexts create identity conflicts between roles as ethical researchers, technical experts, and security contributors, requiring active negotiation rather than simple alignment. The decision between supporting or discouraging PoC publication reflects different security philosophies and identity positions. P12 noted: *"Depends on your point of view. If you care about security, we should have a private place to disclose PoC. If you don't care about securing clients, every PoC should be public."*

Some PoC developers shifted their disclosure practices from *full to partial* when experiencing real or potential misuse of their PoC, representing identity adaptation in response to dual-use consequences. Even under responsible disclosure, PoC developers must decide whether to proceed with full or partial disclosure processes. P13 represents those who opt for full disclosure: *"I often always submit full disclosure. Yeah, because it is useful for the [National CVD], and also useful for the vendors to identify the vulnerabilities." "Yeah, I basically trust the [National CVD].. So yeah, I would like to share my PoC only with them, but I don't want to make it public."* Conversely, P4 represents those choosing partial disclosure: *"My real proof of concept code is something that I rarely share. And that's because my experience with a lot of responsible disclosure teams, within vendors, are not to be trusted completely. Like I've heard also from people around me that say, well, I submitted my code to just show them like the vulnerability, and then, out of the blue, a few days later, you see it online."*

Personal motivations can also play a critical role in shaping disclosure decisions. P12 recounted an early experience involving a cross-site scripting vulnerability in Facebook's chat feature, explaining that he deliberately chose not to disclose this vulnerability because it served his personal interests, using the exploit for about two months until Facebook eventually patched it. This example shows how personal gain in PoC development can influence a PoC developer's disclosure strategy.

Takeaway 2

Beyond individual skills and motivations, PoC developers engage in ongoing, complex negotiations shaped by ethical reasoning, external feedback, and identity tensions. Moral responsibility is often externalized via tool neutrality, while dynamic value assessments fluctuate with vendor responsiveness and community reactions. PoC developers continuously balance their professional identities—as innovators, collaborators, and guardians—within a shifting ecosystem marked by legal ambiguities and cultural complexities.

4.3 Contextual Forces Shaping PoC Developer Choices (RQ3)

Similar to [19], our analysis reveals important contextual factors that operate outside the conventional framework but significantly influence dual-use decision-making in PoC development decisions that don't fit neatly within traditional framework boundaries. These include security community culture with its disclosure norms and ethical frameworks, and vendor/legal environments encompassing CVD policies and regulatory constraints. These external factors shape the context within which individual motivational processes unfold (colored in blue in Figure 2).

4.3.1 Vendor Responsiveness & Communication

Vendor responsiveness creates dynamic feedback loops that modify future disclosure decisions in ways not captured by static EVT value assessments. A crucial aspect of the disclosure process is the tone of communication and responsiveness between PoC developers and vendors. Most of them believe that adopting a friendly and respectful tone not only fosters cooperative dialogue but also helps mitigate legal risks.

> *"I [think] definitely in your first messaging, saying something, giving them consequences for not replying is coming off very, you know, evil. To them it sounds like you're an enemy, whereas I more wanna make myself sound like a friend to them. So they can ask me questions, come back to me, or even order a real pen test if they want. But if you start saying, after 90 days, I will disclose this then, you kind of put the whole conversation in a negative tone, whereas it should be positive. And usually, I honestly couldn't care too much about actually disclosing stuff. What am I gonna do? I'm just gonna make it public. Threatening companies increases the chance of getting lawyers..., which is far worse than just being nice."* (P11)

However, the situation often changes when vendors fail to mirror that same level of communication, forcing PoC developers to adjust their disclosure to direct disclosure models. P11 recalled a case involving vulnerabilities in smart lights: *"I found the vulnerability in these [smart] lights. I tried to let them know and they said no, we're not interested. Go away."* In another instance, following an unresponsive disclosure to Microsoft, P12 explained: *"the first in my life I published to Microsoft. They didn't care. They didn't even disclose it. They didn't give me some reward or something. Nothing happened so I throw that script on the web. And 3 days later, we receive a patch through Windows update."* In both cases, PoC developers transitioned from responsible disclosure approach to direct publication to pressure vendors into remediation.

PoC developers increasingly rely on national frameworks—such as CERT programs—to report vulnerabilities. These delegations not only facilitate responsible disclosure but also reduce the administrative burden of direct vendor communication. Many participants report that these systems provide a structured and predictable timeline for vendor responses.

4.3.2 Community Trust and Anonymity Dynamics

PoC developer communities are characterized by anonymity and low trust, which shape unique social dynamics around information sharing and collaboration. Despite limited trust, collaboration remains vital, with PoC developers exchanging knowledge and jointly solving problems through platforms such as Discord, Telegram, bug bounty programs, and online forums. P11 exemplified this saying *"Yeah, in the bug bounty community, we do collaborate a lot and it's really fun to see. Definitely. If you're working on the same target, somebody will go. Hey, I have found this small issue here. I need this behavior to kind of chain it together and make it more impactful and then somebody else will go. I found that behavior on this on this website. Does that work together and then so yeah, that's quite fun to collaborate together in that way."*

Within this collaborative environment, a layered dual-use dilemma emerges. On one level, shared motives for knowledge exchange drive the community, as P12 explains: *"There are many collaboration channels... to share knowledge because sometimes they share the same goals you never know how many people hate Microsoft."* However, P12 also stated the open sharing of early-stage PoC can inadvertently enable unauthorized manipulation: *"Well, issue was where in the community I discussed that. I said I have command execution, not remote code execution. There is a command execution in Microsoft website, so one of the guys develop and had remote code exec on that server and he calls denial of service for Microsoft. It didn't last so much like two or three minutes, all servers were down."* The dilemma deepened when the collaborator chose not to contribute back: *"He succeeded in getting some remote code execution, and when I asked him to share the code and so he didn't share it back, the guy just logged out. Happens, yeah. You know, you just move on because everyone is anonymous."* The anonymous collaboration model doesn't fit traditional social influence categories within EVT. It creates different dynamics than conventional social contexts where EVT has been applied. This community paradox was described by P12 as: *"Zero trust you. Don't trust anyone in that community, but, as I said, you go to certain servers, certain channels where people share the same motives and goals. So trust is not a factor."*

4.3.3 Cultural & Legal Constraints on PoC Publication

Cultural and legal factors create constraints that operate outside traditional EVT motivational processes while significantly influencing decision-making outcomes. P16 highlighted that cultural and legal pressures may discourage public PoC publication, except where the impact is minimal and remediation does not require end-user action.

Other reasons PoC developers opt not to disclose vulnerabilities or PoC at all range between: issues deemed insignificant (P7), experiencing burnout (P1), someone else having published it (P7), wish to avoid reputational risk (P16), due to cultural or language communication challenges (P11). According to P2, social media can rapidly amplify risks *"looks for what's discussed on platforms like Twitter. And because people on Twitter can't shut up, so even though they have an NDA, they will post it on Twitter."* These cultural dynamics affect information flow and knowledge development in ways that indirectly influence EVT components while representing external constraints on PoC developer decision-making.

The PoC developers often bypass intermediary channels of disclosure by publicly, yet often anonymously, sharing both vulnerability details and PoC, often via online or underground platforms, as P12 noted: *"other PoC are like just thrown in the web, or ExploitDB, [or in] GitHub somewhere."* These platform choices reflect cultural norms and legal constraints that operate independently of individual motivational processes.

Takeaway 3

PoC exploit developers' decisions balance intrinsic satisfaction and utility with risks of misuse and legal costs. Vendor responsiveness critically influences disclosure strategies, fostering cooperation or prompting direct public release. Anonymous, low-trust communities enable collaboration but increase misuse risks. Cultural and legal contexts further complicate disclosure decisions. This dynamic interplay transforms PoC disclosure into a continuous negotiation between advancing security and managing dual-use risks—far beyond a simple responsible versus irresponsible dichotomy.

5 Discussion

Our findings fundamentally challenge the prevailing binary conceptualization of responsible versus irresponsible disclosure that dominates current cybersecurity discourse [7, 48]. Rather than operating within fixed ethical frameworks, PoC developers engage in sophisticated motivational negotiations that extend far beyond the technical-ethical dichotomy assumed by prior work [20, 46].

Where previous studies have primarily focused on observable behaviors and temporal patterns of exploit deployment [8, 37], our EVT-informed analysis reveals the underlying psychological mechanisms driving PoC development decisions. This represents a significant departure from economic models that assume rational cost-benefit calculations [5, 6] by demonstrating how expectancy beliefs, value assessments, and identity considerations create complex motivational landscapes that resist simple optimization frameworks.

Our extension of EVT through dual-use moral reasoning, dynamic value assessment, and identity navigation addresses a critical theoretical gap in cybersecurity research. While motivational theories have been successfully applied to defensive security behaviors [19, 31], this study provides the first systematic application to offensive security practices. The concept of motivational ambivalence, where the same factors that drive thorough exploit development also heighten misuse potential, offers a novel framework for understanding dual-use technology development that extends beyond existing governance models [23, 35].

Dual-Use Moral Reasoning as Responsibility Externalization. The insight of tool neutrality beliefs as a systematic form of moral reasoning highlighted by PoC developers in this study provides new theoretical insight into how practitioners manage psychological costs in dual-use contexts. Unlike previous work that has treated ethical considerations as static constraints [21], we demonstrate that those PoC developers actively construct moral frameworks that enable continued engagement despite potential harmful consequences. This process of responsibility externalization represents a previously undocumented psychological mechanism that may apply broadly across dual-use technology domains.

Dynamic Value Assessment and Contextual Adaptation. Traditional EVT assumes relatively stable value assessments, but our findings reveal that PoC developers continuously recalibrate their risk-benefit evaluations based on vendor responsiveness, disclosure outcomes, and community feedback. This dynamic process challenges static models of responsible disclosure [43] and suggests that governance approaches must accommodate ongoing negotiation rather than fixed rules. The identification of vendor behavior as the primary driver of this recalibration process provides empirical support for relationship-based rather than rule-based disclosure frameworks.

Identity Navigation in Professional Contexts. Our analysis reveals that dual-use decision-making involves continuous negotiation between competing professional identities—ethical researcher, technical expert, security contributor—rather than simple alignment with pre-existing values. This extends beyond existing work on hacker identity formation [21] by showing how professional contexts create identity conflicts that require active management rather than passive expression of fixed ethical commitments.

Challenging Economic and Rational Choice Models

Our findings directly contradict the assumptions underlying current economic models of vulnerability disclosure and exploitation. Allodi's work on attacker economics [5, 6] demonstrates that malicious actors prioritize low-effort, high-impact vulnerabilities through rational cost-benefit analysis. However, our PoC developer interviews reveal that PoC creation involves fundamentally different decision-making processes that cannot be reduced to simple economic optimization.

The concept of motivational ambivalence illustrates this divergence most clearly. Economic models would predict that rational actors would minimize dual-use risks to avoid potential costs, but our participants consistently described how intrinsic satisfaction from technical mastery drove them toward more sophisticated—and potentially more dangerous—exploit development. This pattern suggests that non-economic motivational factors may be more influential than previously recognized in shaping the dual-use technology landscape.

Similarly, our finding that vendor responsiveness is the primary determinant of disclosure decisions challenges economic models that assume stable preferences and consistent cost-benefit calculations. The dynamic nature of value assessment revealed in our data suggests that PoC developer behavior is fundamentally contextual and relationship-dependent rather than driven by fixed utility functions.

Implications for Vulnerability Governance and Policy

Our results have implications for current approaches to vulnerability governance and coordinated disclosure policy. Existing frameworks assume that clear rules and structured processes will produce predictable outcomes [48], but our findings suggest that individual motivational factors may be more influential than institutional constraints in determining actual behavior.

The Recognition Paradox and Incentive Misalignment. We identify a previously undocumented asymmetry in community recognition patterns: vulnerability discoverers receive

celebration while PoC developers face stigmatization, despite both activities representing dual-use research. This recognition paradox creates perverse incentives that may actually increase security risks by pushing PoC development underground or toward less responsible disclosure practices. Current bug bounty research has focused on economic incentives [1, 10], but our findings reveal that social recognition dynamics may be equally important in shaping behavior.

This misalignment suggests that governance frameworks focused solely on constraining harmful behavior may be counterproductive. Instead, policies that acknowledge and reward responsible PoC development alongside vulnerability discovery may be more effective in channeling dual-use activities toward beneficial outcomes.

Dynamic Governance for Dynamic Decisions. The finding that PoC developers continuously recalibrate their value assessments based on external feedback suggests that static governance approaches may be fundamentally inadequate for dual-use contexts. Unlike traditional risk management frameworks that assume stable risk profiles, dual-use technologies require governance systems that can adapt to changing motivational landscapes and contextual factors.

Our identification of vendor responsiveness as the primary driver of disclosure decisions provides specific guidance for improving governance outcomes. Rather than focusing primarily on rule development and enforcement, effective governance may require greater attention to relationship management and communication quality between stakeholders.

6 Recommendations

Based on our empirical findings, we propose specific interventions derived directly from our EVT-informed analysis of PoC developer decision-making. Each recommendation addresses specific motivational factors, contextual constraints, or community dynamics identified in our results.

For Vendors and Organizations

R1: Implement Structured Response Protocols: Our finding that vendor responsiveness is the primary determinant of disclosure decisions (exemplified by P12's Microsoft case where unresponsiveness led to immediate public disclosure) requires systematic changes to vendor communication practices. Organizations should establish 48-hour acknowledgment protocols, assign dedicated technical liaisons, and provide regular progress updates to PoC developers throughout the disclosure process.

R2: Deploy Recognition Programs: Our discovery that vulnerability discoverers receive celebration while PoC developers face stigmatization creates perverse incentives. Based on participants' descriptions of wanting professional recognition (P12's career advancement motivation), vendors should create formal acknowledgment programs that recognize responsible PoC development through technical advisory roles, security team consultation opportunities, and public recognition equivalent to vulnerability discovery programs.

R3: Establish Graduated Response Systems: Our identification of PoC development existing on a sophistication continuum from crash demonstrations to weaponized exploits requires differentiated organizational responses. Vendors should implement classification systems that distinguish between demonstration-level PoC (requiring standard disclosure timelines) and weaponized exploits (requiring expedited response protocols and enhanced PoC developer support).

For Policymakers and Standards Bodies

R4: Create Safe Harbor Provisions: Multiple participants (P14, P16) described avoiding PoC development due to legal uncertainty, directly impacting the cost component of our EVT analysis. Policymakers should establish explicit safe harbor protections for good-faith security research including PoC development, with clear criteria based on disclosure intent and target selection rather than tool sophistication.

R5: Support Dynamic Disclosure Frameworks: Our discovery that PoC developers continuously recalibrate risk-benefit evaluations based on vendor behavior requires policy frameworks that accommodate this dynamic process. Standards bodies should develop graduated disclosure models supporting time-delayed publishing (as practiced by P8), partial disclosure options (preferred by P4), and context-specific evaluation criteria rather than uniform disclosure timelines.

R6: Fund National CVD Infrastructure: Our finding that PoC developers increasingly rely on national CERT programs as trusted intermediaries (reported by multiple participants as reducing vendor communication burden) indicates the value of institutional infrastructure. Policymakers should expand funding for national vulnerability coordination centers that can serve as neutral intermediaries between PoC developers and vendors.

For Platforms and Technical Infrastructure

R7: Develop Sophisticated Content Management: Our identification of anonymous, low-trust community dynamics that enable valuable collaboration while increasing misuse risks requires technical infrastructure that can manage these tensions. Platforms should implement graduated sharing systems that allow vulnerability details sharing while controlling weaponized component access through community membership verification, temporal restrictions, and reputation-based access controls.

R8: Create Verified Researcher Networks: The "zero trust but shared goals" phenomenon identified in our community analysis suggests the need for reputation systems that preserve beneficial anonymity while providing accountability mechanisms. Platforms should develop verified researcher programs

that enable higher-trust collaboration through cryptographic identity systems and peer verification processes.

R9: Implement AI-Aware Risk Assessment: Our discovery that AI integration enhances PoC developers' expectancy beliefs and enables more sophisticated exploit development requires automated risk assessment systems. Platforms should invest in machine learning systems that can evaluate relative risk levels of PoC submissions and apply appropriate sharing restrictions based on sophistication metrics rather than requiring manual review.

For the Security Research Community

R10: Establish Dual-Use Decision Training: Our identification of ongoing identity negotiation between ethical research and technical mastery roles requires structured support for navigating these tensions. The community should develop training programs that teach dual-use decision-making frameworks, impact assessment methodologies, and ethical reflection processes rather than relying on informal trial-and-error learning described by participants.

R11: Create Collaborative Disclosure Support Networks: Our finding that vendor behavior creates dynamic feedback loops affecting future disclosure decisions suggests the need for community infrastructure that can buffer these relationships. Professional associations should establish intermediary organizations that can facilitate vendor-developer communication while providing accountability and support mechanisms for responsible disclosure processes.

R12: Develop Community Standards: Our documentation of differential stigmatization of PoC developers versus vulnerability discoverers requires systematic community norm changes. Security conferences, professional organizations, and community platforms should establish recognition standards that celebrate responsible exploit development as security contribution equal to vulnerability discovery, including dedicated presentation tracks and award categories.

For Academic and Industry Research

R13: Expand Motivational Research Applications: Our successful application of EVT to offensive security practices demonstrates the value of psychological frameworks in cybersecurity research. Researchers should explore additional motivational theories (self-determination theory, social cognitive theory) and their applicability to other security domains including malware analysis, threat intelligence, and security tool development.

R14: Conduct Longitudinal Studies of Dynamic Value Assessment: Our finding that vendor behavior shapes future disclosure decisions through dynamic value assessment requires longitudinal research designs that can capture these temporal effects. Researchers should design multi-wave studies that track how disclosure experiences influence subsequent PoC developer decision-making across different vendor types and disclosure contexts.

R15: Investigate Cross-Cultural Variations in Dual-Use Decision-Making: Our identification of cultural constraints as operating outside traditional EVT suggests the need for systematic cross-cultural research. International research collaborations should examine how different legal frameworks, cultural norms, and institutional contexts influence dual-use decision-making to inform more effective global cooperation on vulnerability governance.

7 Conclusion

In this study, we examined how PoC exploit developers navigate the dual-use dilemma inherent in exploit development. By applying Expectancy-Value Theory (EVT), we showed how PoC exploit developers' decisions are guided not merely by capability or ideology, but through dynamic assessments of their technical capabilities, expected outcomes, and multifaceted value considerations. Our analysis shows that the dual-use dilemma is not simply a binary choice between responsible and irresponsible disclosure, but rather a complex navigation of competing EVT components influenced by contextual factors specific to dual-use technologies. Expectancy beliefs about technical success and security impact, combined with subjective assessments of utility value, intrinsic satisfaction, identity alignment, and various cost considerations, systematically predict PoC developer behavior across the disclosure spectrum. The theoretical extensions we proposed—dual-use identity navigation, contextual cost amplification, and dynamic value assessment—expand EVT's applicability to cybersecurity contexts while identifying areas for future theoretical development. These insights offered practical guidance for improving coordinated vulnerability disclosure processes through EVT-informed interventions that address PoC developer motivations rather than simply imposing policy constraints. Our findings suggested that supporting responsible dual-use technology development requires addressing the full spectrum of PoC developer motivations—technical confidence, value perceptions, identity considerations, and cost concerns—rather than focusing solely on regulatory or technical controls. By understanding and supporting the complex motivational landscape revealed through our analysis, we can better harness the security benefits of dual-use technologies while minimizing their potential for harm.

Acknowledgments

This work is partly supported by the Dutch Research Council (NWO) under the RAPID project (Grant No. CS.007) and is partly funded by King Abdulaziz City for Science and Technology (KACST).

Ethical Considerations

This study received formal approval from the Institutional Review Board (IRB) of the authors' institution, ensuring adherence to ethical standards in human subjects research. Participant consent was informed and voluntary, with clear assurances of anonymity and withdrawal rights.

Stakeholders

We identify three primary stakeholder groups potentially impacted by this research.

Security Researchers. Our paper focuses on understanding the motivations and decision-making processes of PoC developers. To protect participant privacy and minimize any risk of harm, all data were anonymized prior to analysis. Researchers accessed only de-identified transcripts without any personally identifying information and agreed not to attempt re-identification. We conducted transcript reviews with participants to ensure accuracy and to confirm their comfort with the shared content. The artifacts released with this work exclude any sensitive or identifying information. Economic or reputational harms to participants are unlikely; if anything, the findings offer greater visibility into the motivations and challenges faced by PoC developers, potentially benefiting their recognition and policy engagement.

Software Vendors and Organizations. Vendors whose products are targeted by PoC exploits could experience indirect impacts through improved understanding of the exploit development ecosystem. This work aims to foster better coordination and communication between vendors and researchers by shedding light on vendor responsiveness as a key factor influencing PoC developer disclosure decisions. We minimized risks of misinterpretation by rigorous data validation, thorough review of all analysis outputs, and clear documentation of findings, including limitations.

End Users and Society. Although not directly studied, the ultimate beneficiaries of more secure software systems include end users whose safety depends on timely vulnerability remediation. By deepening understanding of PoC development and disclosure, this work contributes to improved vulnerability lifecycle management, which can reduce risks of exploitation and harm at scale.

Ethical Considerations and Impact

This study adheres to the ethical framework outlined by the Menlo Report, which adapts core Belmont Report principles to cybersecurity and ICT research. We have systematically applied these principles to balance stakeholder interests, minimize harm, and ensure research integrity.

Ethical Principles. *Respect for Persons:* All participants gave informed consent, understanding the study's purpose, voluntary nature, and their right to withdraw. Anonymity and data confidentiality were strictly maintained to respect autonomy and protect participants, especially given the sensitive dual-use context.

Beneficence: We maximized potential benefits by generating actionable insights to improve vulnerability disclosure practices, while minimizing risks through comprehensive anonymization and participant transcript review to avoid harmful exposure or misrepresentation.

Justice: Participant selection was fairly conducted across geographic regions and professional roles relevant to PoC development. Efforts were made to minimize sampling bias and transparently disclose limitations, ensuring equitable representation of the targeted community.

Respect for Law and Public Interest: The research complied with all applicable laws and institutional policies. We avoided publishing any sensitive exploit/PoC details to prevent misuse. Transparency is upheld through full disclosure of methodology and ethical precautions, fostering accountability and community trust.

Potential Harms Addressed. *Tangible Harms:* Risks of financial loss, reputational damage, or psychological distress to participants were mitigated through anonymization, secure data handling, and participant vetting of all quotes.

Human Rights Violations: Participants' rights to privacy and informed consent were prioritized. Data collection respected expectations of confidentiality and autonomy. No data collection or publication infringed on users' privacy or security.

Impact. This research advances understanding of the motivations and ethical navigation of PoC developers, describing the dynamics that influence disclosure strategies and dual-use risk management. By providing empirically grounded recommendations, it aims to foster more responsible vulnerability handling by vendors, researchers, and policymakers. This promotes a safer cybersecurity ecosystem, balancing innovation with risk mitigation. No new vulnerabilities were disclosed or created, ensuring no direct harmful outcomes. Instead, the study informs ethical policy development, trust-building measures, and improved communication frameworks essential for coordinated vulnerability disclosure. Despite inherent legal and professional risks to participants working in sensitive dual-use domains, our study was conducted because: (i) understanding PoC developer motivations and disclosure strategies is critical to enhancing coordinated vulnerability disclosure frameworks; (ii) empirical insights provide foundational knowledge to improve vendor responsiveness, reduce exploit weaponization risks, and foster productive security researcher-vendor collaboration; (iii) transparent, ethical engagement with participants and rigorous privacy safeguards mitigate risks while advancing community-benefiting knowledge; and (iv) our study explicitly avoids publishing exploit code or details that could facilitate malicious use, minimizing potential for harm.

Risk Mitigation. Special care was taken to anonymize

data and prevent any linkage between participants and their responses. The publication avoids release of exploit code or technical details that could facilitate weaponization. Our focus on human, motivational, and contextual aspects of PoC development minimizes potential misuse. Ethical tensions inherent in dual-use research are explicitly discussed, with careful reflection on responsibilities surrounding vulnerability disclosure and exploit publication.

Open Science

In accordance with USENIX Security's open science policy, we commit to providing transparency and facilitating the reproducibility of our research within ethical and legal boundaries. The research artifacts associated with this study include the interview protocol and the codebook, both of which are shared to enable critical review and replication of the study design. The full interview protocol, comprising all questions asked to participants along with examples of follow-up questions, is included in this appendix. Additionally, an excerpt of our codebook, illustrating the coding scheme with codes, definitions, and representative quotes, is provided to offer insight into our thematic analysis. Due to the sensitive nature of our research, particularly given the organizational ranks of our subjects and the small size of our participant pool, we are unable to share the raw transcripts of interviews, as well as anonymized transcripts. Sharing these documents poses a significant risk of de-anonymization, especially since many of the subjects are recognizable figures in technical communities and because technical details involved in this research could serve as unique identifiers or "fingerprints" capable of re-identifying participants. This concern is further supported by institutional review board (IRB) compliance and ethical considerations, which mandate the protection of participant privacy and confidentiality. Our adherence to open science principles is evidenced by the fact that the shared artifacts—the interview protocol and the codebook—permit rigorous evaluation of our study methodology and facilitate replicability by other researchers. These artifacts contain sufficient detail for understanding the study design and coding procedures, thereby serving as valuable resources for assessing the validity and reliability of our findings. Although we do not share the raw data, the transparency in our research process respects both the scientific community's values of openness and the imperative to protect participant privacy. Our approach exemplifies a balanced commitment to open science, allowing the community to scrutinize our research methods while safeguarding sensitive information that could compromise participant anonymity or pose legal risks associated with PoC exploit development work.

References

[1] Omer Akgul, Taha Eghtesad, Amit Elazari, Omprakash Gnawali, Jens Grossklags, Michelle L. Mazurek, Daniel Votipka, and Aron Laszka. Bug Hunters' perspectives on the challenges and benefits of the bug bounty ecosystem. In *32nd USENIX Security Symposium*, pages 2275–2291, Anaheim, CA, August 2023.

[2] Arwa Abdulkarim Al Alsadi, Kaichi Sameshima, Jakob Bleier, Katsunari Yoshioka, Martina Lindorfer, Michel van Eeten, and Carlos H. Gañán. No Spring Chicken: Quantifying the Lifespan of Exploits in IoT Malware Using Static and Dynamic Analysis. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22, page 309–321, New York, NY, USA, 2022.

[3] Arwa Abdulkarim Al Alsadi, Kaichi Sameshima, Katsunari Yoshioka, Michel Van Eeten, and Carlos Hernandez Gañán. Bin There, Target That: Analyzing the Target Selection of IoT Vulnerabilities in Malware Binaries. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '23, page 513–526, New York, NY, USA, 2023.

[4] Yasser Alhelaly, Gurpreet Dhillon, and Tiago Oliveira. When expectation fails and motivation prevails: the mediating role of awareness in bridging the expectancy-capability gap in mobile identity protection. *Computers & Security*, 134:103470, 2023.

[5] Luca Allodi. Economic Factors of Vulnerability Trade and Exploitation. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 1483–1499, New York, NY, USA, 2017.

[6] Luca Allodi, Fabio Massacci, and Julian Williams. The work-averse cyberattacker model: theory and evidence from two million attack signatures. *Risk Analysis*, 42(8):1623–1642, 2022.

[7] Noura Alomar, Primal Wijesekera, Edward Qiu, and Serge Egelman. "You've Got Your Nice List of Bugs, Now What?" Vulnerability Discovery and Management Processes in the Wild. In *16th Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 319–339, August 2020.

[8] Omar Alrawi, Charles Lever, Kevin Valakuzhy, Ryan Court, Kevin Snow, Fabian Monrose, and Manos Antonakakis. The Circle Of Life: A Large-Scale Study of The IoT Malware Lifecycle. In *30th USENIX Security Symposium*, pages 3505–3522, August 2021.

[9] Argos OpenTech. Argos Translate Package Index. Accessed: 2025. https://www.argosopentech.com/argospm/index/.

[10] Soodeh Atefi, Amutheezan Sivagnanam, Afiya Ayman, Jens Grossklags, and Aron Laszka. The Benefits of Vulnerability Discovery and Bug Bounty Programs: Case Studies of Chromium and Firefox. In *Proceedings of the ACM Web Conference 2023*, page 2209–2219, New York, NY, USA, 2023.

[11] ATLAS.ti. ATLAS.ti: The Qualitative Data Analysis & Research Software, 2025. https://atlasti.com/.

[12] Pierpaolo Basile, Elio Musacchio, Marco Polignano, Lucia Siciliani, Giuseppe Fiameni, and Giovanni Semeraro. LLaMAntino: LLaMA 2 Models for Effective Text Generation in the Italian Language. *arXiv preprint arXiv:2312.09993*, 2023. https://arxiv.org/abs/2312.09993.

[13] H Russell Bernard, Amber Wutich, and Gery W Ryan. *Analyzing qualitative data: Systematic approaches*. SAGE publications, 2016.

[14] John Van Beveren. A conceptual model of hacker development and motivations. In *Journal of E-Business*, volume 1, pages 1–9, 2001.

[15] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[16] Santiago Castro. Fast Krippendorff: Fast computation of Krippendorff's alpha agreement measure. GitHub repository; version 0.8.2, 2017. https://github.com/pln-fing-udelar/fast-krippendorff.

[17] Stefanos Chaliasos, Marcos Antonios Charalambous, Liyi Zhou, Rafaila Galanopoulou, Arthur Gervais, Dimitris Mitropoulos, and Benjamin Livshits. Smart Contract and DeFi Security Tools: Do They Meet the Needs of Practitioners? In *IEEE/ACM 46th International Conference on Software Engineering*, New York, NY, USA, 2024.

[18] Kathy Charmaz. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. Sage Publications, London; Thousand Oaks, Calif., 2006.

[19] Xiaowei Chen, Sophie Doublet, Anastasia Sergeeva, Gabriele Lenzini, Vincent Koenig, and Verena Distler. What Motivates and Discourages Employees in Phishing Interventions: An Exploration of Expectancy-Value Theory. In *20th Symposium on Usable Privacy and Security (SOUPS 2024)*, pages 487–506, Philadelphia, PA, August 2024.

[20] Samuel Chng, Han Yu Lu, Ayush Kumar, and David Yau. Hacker types, motivations and strategies: A comprehensive framework. *Computers in Human Behavior Reports*, 5:100167, 2022.

[21] Gabriella Coleman. *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton University Press, 2012.

[22] CyberPeace. Rapid Exploitation of Proof-of-Concept Exploits: A Growing Cybersecurity Threat, 2024. https://www.cyberpeace.org/resources/blogs/rapid-exploitation-of-proof-of-concept-exploits-a-growing-cybersecurity-threat.

[23] Thomas Douglas. *An Expected-Value Approach to the Dual-Use Problem*, pages 133–152. ANU Press, 2013.

[24] Jacquelynne S. Eccles and Allan Wigfield. From expectancy-value theory to situated expectancy-value theory: A developmental, social cognitive, and sociocultural perspective on motivation. *Contemporary Educational Psychology*, 61:101859, 2020.

[25] Soufian El Yadmani, Robin The, and Olga Gadyatskaya. SecurePoC: a helping hand to identify malicious CVE proof of concept exploits in GitHub. In *19th USENIX Conference on Offensive Technologies*, USA, 2025.

[26] Jennifer Fereday and Eimear Muir-Cochrane. Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods*, 5(1):80–92, 2006.

[27] Stuart Graham and Shelagh Kennedy. Women in Cybersecurity: A Manifesto for Today. *Journal of Cybersecurity Education, Research and Practice*, 2020(1):1–15, 2020.

[28] Greg Guest, Arwen Bunce, and Laura Johnson. How many interviews are enough? An experiment with data saturation and variability. *Field methods*, 18(1):59–82, 2006.

[29] Allen D. Householder, Jeff Chrabaszcz, Trent Novelly, David Warren, and Jonathan M. Spring. Historical Analysis of Exploit Availability Timelines. In *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*, August 2020.

[30] Adeel Javaid. Psychology of Hackers. *SSRN Electronic Journal*, 2013. https://doi.org/10.2139/ssrn.2342620.

[31] Jeffrey L Jenkins, Alexandra Durcikova, Grayson Ross, and Jay F Nunamaker Jr. Encouraging Users to Behave Securely: Examining the Influence of Technical, Managerial, and Educational Controls on Users' Secure

Behavior. In *ICIS 2010 - 31 International Conference on Information Systems*, 2010.

[32] Daniel Kang, Xuechen Li, Ion Stoica, Carlos Guestrin, Matei Zaharia, and Tatsunori Hashimoto. Exploiting Programmatic Behavior of LLMs: Dual-Use Through Standard Security Attacks. In *45th IEEE Symposium on Security and Privacy Workshops, SPW 2024*, pages 132–143, United States, 2024.

[33] Klaus Krippendorff. *Content analysis: An introduction to its methodology*. Sage publications, 2019.

[34] Richard A Krueger. *Focus groups: A practical guide for applied research*. Sage publications, 2014.

[35] Seumas Miller and Michael Selgelid. Ethical and Philosophical Consideration of the Dual-use Dilemma in the Biological Sciences. *Science and engineering ethics*, 13:523–80, 01 2008.

[36] Miftahul Munir. The Effectiveness of Bug Bounty Program for Technology Company Ecosystem. *Proceedings Series on Social Sciences & Humanities*, 21:85–88, Apr. 2025.

[37] Kartik Nayak, Daniel Marino, Petros Efstathopoulos, and Tudor Dumitraş. Some Vulnerabilities Are Different Than Others - Studying Vulnerabilities and Attack Surfaces in the Wild. In *International Symposium on Recent Advances in Intrusion Detection*, volume 8688 LNCS, pages 426–446. Springer Verlag, 2014.

[38] Jennifer Petersen and Janet Shibley Hyde. Chapter Two - Gender-Related Academic and Occupational Interests and Goals. In Lynn S. Liben and Rebecca S. Bigler, editors, *The Role of Gender in Educational Contexts and Outcomes*, volume 47 of *Advances in Child Development and Behavior*, pages 43–76. JAI, 2014.

[39] Yangheran Piao, Temima Hrle, Daniel Woods, and Ross Anderson. Study Club, Labor Union or Start-Up? Characterizing Teams and Collaboration in the Bug Bounty Ecosystem. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 20–20, Los Alamitos, CA, USA, May 2025.

[40] Marcus K. Rogers. A two-dimensional circumplex approach to the development of a hacker taxonomy. *Digital Investigation*, 3(2):97–102, 2006.

[41] Margarete Sandelowski. Whatever happened to qualitative description? *Research in nursing & health*, 23(4):334–340, 2000.

[42] Mario Silic. Dual-use open source security software in organizations - Dilemma: Help or hinder? *Computers & Security*, 39:386–395, November 2013.

[43] Jonathan M. Spring, Eric Nelson Hatleback, Art Manion, and Deana Shick. Prioritizing vulnerability response: A stakeholder-specific vulnerability categorization (version 1.0). In *Workshop on the Economics of Information Security*, 2020.

[44] Neville Stanton. Hierarchical task analysis: Developments, applications, and extensions. *Applied ergonomics*, 37:55–79, 02 2006.

[45] Bill Toulas. Hackers use PoC exploits in attacks 22 minutes after release. BleepingComputer, 2024. https://www.bleepingcomputer.com/news/security/hackers-use-poc-exploits-in-attacks-22-minutes-after-release/.

[46] Daniel Votipka, Rock Stevens, Elissa Redmiles, Jeremy Hu, and Michelle Mazurek. Hackers vs. Testers: A Comparison of Software Vulnerability Discovery Processes. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 374–391, 2018.

[47] VulnCheck. A Peek into the Last Decade of Vulnerability Exploitation. Online report, 2024. https://www.vulncheck.com/blog/state-of-exploitation-a-decade.

[48] T. Walshe and A.C. Simpson. Coordinated Vulnerability Disclosure programme effectiveness: Issues and recommendations. *Computers & Security*, 123(C), December 2022.

[49] Allan Wigfield and Jacquelynne S. Eccles. Expectancy–Value Theory of Achievement Motivation. *Contemporary Educational Psychology*, 25(1):68–81, 2000.

A Interview Protocol

This section outlines the semi-structured interview protocol used in our study. The protocol is organized into four thematic areas: (1) Background and Experience, (2) Motivation, (3) PoC Exploit Development Process, and (4) Disclosure, Publication, and Potential Misuse.

Background and Experience

1. What is your main job?
   - 1.1. How many years have you been in this position?
2. How does your job relate to PoC exploit development?
   - 2.1. If it is not directly related, how would you describe your involvement?
3. How many years have you been involved in developing PoC exploits?
4. How frequently do you work on developing PoC exploits?
5. On a scale of 1–5 (1 = beginner, 5 = expert), how would you rate your PoC exploit development skills?
6. How many PoC exploits have you developed (publicly disclosed or not)?
   - 6.1. How many of them are public?
   - 6.2. Where can they be found?
   - 6.3. What do you count or consider as a PoC exploit?
7. For which types of software do you develop PoC?
   - 7.1. Why do you focus on those types?
   - 7.2. Can you briefly describe the last three PoC you developed (or any three you recall)?

Motivation

1. What motivates you to develop PoC exploits for vulnerabilities?

PoC Exploit Development Process

1. Can you walk me through your typical process for developing a PoC exploit?
   - 1.1. How do you identify or search for vulnerabilities?
   - 1.2. What factors do you prioritize when deciding to develop a PoC?
   - 1.3. Do you collaborate with others? Why or why not?
     - 1.3.1. If yes, what role do you usually play in the collaboration?
   - 1.4. Why do you follow this particular process?
   - 1.5. Have you ever used alternative methods? If so, what made you switch?
2. Have you developed PoC for known vulnerabilities discovered by others that lacked a PoC? Why?
3. Have you developed PoC for vulnerabilities that already had existing ones? Why?
4. What programming languages or tools do you typically use, and why?
5. How do you validate or determine that your PoC exploit is successful?

PoC Disclosure, Publication, and Potential Misuse

1. After developing a PoC, what steps do you take for reporting it?
   - 1.1. If disclosing, to whom do you disclose?
     - 1.1.1. What guidelines or principles do you follow to ensure responsible disclosure?
   - 1.2. Do you usually report full or partial disclosure? Why?

     Full disclosure: Sharing both vulnerability details and the complete PoC.

     Partial disclosure: Sharing only vulnerability details.
   - 1.3. What are your reasons for making PoC public?
     - 1.3.1. Where do you typically publish them?
2. Are there cases where you chose not to disclose or publish a PoC you developed? Why?
3. What specific impacts or responses have you observed from the community or vendors regarding your PoC?

B Interview Participants Demographics

Table 2 provides a detailed overview of the individuals who contributed to our study on PoC exploit development. It summarizes key characteristics such as geographic distribution, professional roles, experience levels, and frequency of PoC development among the 16 participants. This demographic context is essential for understanding the diversity and representativeness of our qualitative sample, highlighting factors that may influence perspectives on PoC creation, disclosure, and the broader dual-use dilemma explored in the main study.

Table 2: Summary of participant roles and background.

ID	Role	Sec. Role[1]	Owner	Yrs Job	Yrs Sec.	Yrs PoC	#Vuln.	#PoC (public)	Platform	Freq.	Skill[2]	Software Types	Recruitment[3]
P1	Pentester	F, V, BB	-	4.6	-	9	6–7	10–12 (2–3)	-	unclear	3.5	Web, Desktop, Electron, IoT	F, S
P2	Cybersec R&D Eng.	F, V, BB	-	1	4	-	-	300 (0)	-	unclear	3	Web, Network mgmt, IoT, PC, HD	C
P3	Owner		✓	9	16	11	-	15–20 (>3)	GitHub	3–4/mo	3–4	Web, Mobile, band &IoT devices, Network & ICS protocols	S
P4	Blue teamer	V, BB	✓	3	4–5	-	-	7–8	-	2–3/yr	3.0	PDF, Network mgmt	M
P5	Pentester (red teamer)	V	-	4	-	2	-	7 (1)	-	Monthly	2.5	mgmt System	S
P6	CTO	-	✓	8	-	17	-	30 – 40 (5–10)	-	5–10/yr	2	Industrial switches, SSH vuln, Ransomware exfiltration tool vuln	C
P7	Cybersec consultant– Pentester (red teamer)	-	-	2	-	2–3/mo	12	12 (0)	-	2–3/we	2–3	Hypervisor software	F
P8	Pentester	V, BB	✓	10		6–7	> 40	50–60	YouTube, others	10–20/yr	3	Any SW or Sys but mainly Web	S
P9	Hacker (Pentester)	Company PT	-	3	3	11	unclear	10 (0)	-	Monthly	3	Web, Mobile apps	C
P10	Hacker (Red Team)	V	✓	1	3	10	6–7	10–12 (2–3)	-	2–3/mo	3.0	Web, Local apps	F, C
P11	Lead Pentester	V, BB	-	2	6	8	35 (only in 2024)	100s (40–50)	LinkedIn	-	4.0	Web, AI apps	F, C
P12	Pentester	BB	-	-	4	5	3	<50 (36)	ExploitDB	Monthly	Web: 3, Bin: 4	Web, Binaries, Mobile	C
P13	Sec. Researcher	-	-	4	4	19	20	20 (0)	-	3/yr	4	IoT, Web UI	C
P14	Hacker (Pentester)	V	-	18	50	-	unclear	> 100 (>1)	Conferences, Youtube, others	unclear	5	Systems (Mainly Windows)	S
P15	Pentester	-	-	6–7	6–7	6–7	2	2,000 (0)	-	2–3/mo	4-5	Embedded sys, IoT (consumer&industrial)	C
P16	Security analyst& reverse engineer	-	-	12	12	12	10	100s (0)	-	Daily	4-5	Web, Embedded sys, IoT	C

[1] Roles – F: Freelancer, V: Voluntary security researcher/tutor, BB: Bug bounty.
[2] Skill scale: 1 = Beginner, 5 = Expert.
[3] Recruitment method – C: Personal Contact, M: Social Media, F: Form, S: Snowball.

C Codebook

Table 3 presents the final thematic codebook derived from a hybrid inductive–deductive analysis of anonymized interview transcripts [15, 26], with high inter-coder reliability (Krippendorff's $\alpha = 0.857$).

Table 3: Interview Codebook.

Code	Description	Code	Description
Background and Demographics			
role-primary	Main job role of the participant	no-vuln-discovered	Number of vulnerabilities discovered
role-secondary	Additional roles beyond primary job	no-poc-developed	Total number of PoC developed (public and private)
job-involve-poc	Job involves PoC exploit development	no-poc-public	Number of public PoC
job-no-poc	Job does not involve PoC exploit development	freq-poc	Frequency of PoC development
years-job	Years in current position	poc-lasts	Most recent PoC developed
years-security	Years in security field	poc-platforms	Platforms used to share PoC
years-poc	Years developing PoC	software-targets	Types of software targeted
skill-level	Self-assessed PoC development skill level (1–5)		
PoC Perceptions and Definitions			
poc-def-code-centric	Code-based only	poc-def-spectrum	Continuum from crash to full exploit
poc-def-demo	Simple crash demonstrations	poc-def-any	Any form (text, video, code)
poc-def-weaponized	Fully weaponized exploits	poc-def-0day	Only newly-discovered or zero-day vulnerabilities
Motivational Factors			
mot-ego	Prove technical and demonstrate expertise	mot-fun-challenge	Fun and game-like activity
mot-curiosity	Intellectual and technical challenges	mot-civic-duty	Responsibility as a good digital citizen
mot-demonstrate	Showcase technical abilities	mot-vendor-pressure	Pressure vendors for faster patching
mot-recog	Community and professional reputation	mot-education	Improve security community skills
mot-sec-improv	Improve cybersecurity for society	mot-personal-protection	Protect personal accounts and data
mot-protection	Safeguard organizational systems and colleagues	mot-financial	Seek monetary gain or career advancement
Development Tools and Methods			
tool-python	Use Python for PoC development	approach-known-vulns-no-poc	Disclosed vulns without existing exploits
tool-ai-assisted	Use AI tools for code assistance	approach-unknown-vulns	Newly discovered or zero-day vulnerabilities
tool-other-languages	Use non-Python languages	approach-known-vulns-poc	Disclosed vulns with existing exploits
tool-shodan	Use Shodan for target discovery	approach-fuzzing	Automated vulnerability discovery
validation-local	Test in isolated local environments	approach-poc-adaptation	Validate existing public PoC
validation-physical	Test on physical devices	approach-poc-scratch	Develop PoC from scratch
validation-systematic	Structured multi-environment validation	approach-poc-scratch-reason	Avoid hidden malicious intent
approach-patch-diffing	Identify vulns via patch diffing		
Workflow and Process			
method-systematic	Structured, repeatable processes	work-both	Alternate between solo and collaborative work
method-opportunistic	Resource- and context-driven approaches	work-style-reasons	Reasons for work style choices
work-solo	Independent PoC development	work-collaborative-role	Roles when collaborating
work-collaborative	Collaborative PoC development		
Disclosure Strategies & Reporting Channels			
channel-cvd-national	National coordinated vulnerability disclosure programs	disclosure-delayed	Delay until patching
channel-vendor-direct	Direct reporting to vendors or organizations	disclosure-immediate	Immediate publication to pressure vendors
channel-internal-client	Internal or client-only disclosure	disclosure-conditional	Conditional public disclosure
channel-public-platforms	Public disclosure platforms	disclosure-reasons	Reasons for disclosure strategy
channel-restricted	Private or underground channels	disclosure-reasons-impact	Scope and severity of potential misuse
disclosure-full	Share vulnerability details and full PoC	disclosure-reasons-legal	Legal concerns
disclosure-partial	Share details without PoC code	disclosure-reasons-moral	Moral burden of potential harm
disclosure-none	No public disclosure	disclosure-reasons-reput	Reputational concerns
disclosure-delayed	Delay until patching	disclosure-reasons-oppo	Opportunity costs
disclosure-immediate	Immediate publication for pressure	disclosure-reasons-legal-supportive	Permissive legal context
Publication Practices			
full-poc-public	Publicly available full PoC	publication-monetization	Sale through private markets
selective-poc-community	Restricted community sharing	community-trust-high	High trust in community
publication-kowdg-share	Community knowledge sharing	community-trust-low	Skepticism toward community
publication-pressure-tactic	Pressure vendors via publication	community-anonymous	Preference for anonymity
publication-recognition	Community recognition		
Experience and Learning			
exp-positive-disclosure	Beneficial disclosure outcomes	exp-misuse-witnessed	Witnessed PoC misuse
exp-negative-disclosure	Harmful disclosure outcomes	learning-community-driven	Community-based learning
exp-vendor-cooperation	Positive vendor interactions	learning-self-directed	Independent skill acquisition
exp-vendor-hostility	Negative vendor interactions		
Identity and Values			
identity-ethical	Responsible security researcher identity	values-prevention	Prevent misuse over transparency
identity-community	Contributor to collective security	values-open-source	Commitment to open knowledge sharing
identity-expert	Identity centered on technical expertise	values-neutrality	PoC as neutral tools
identity-conflict	Tension between professional identities		