CONNECTEDNESS: A DIMENSION OF SECURITY BUG SEVERITY ASSESSMENT TO MEASURE UNCERTAINTY

A PREPRINT

Chan Shue Long*
shue87655421@gmail.com

March 14, 2025

ABSTRACT

Current frameworks for evaluating security bug severity, such as the Common Vulnerability Scoring System (CVSS), prioritize the ratio of exploitability to impact. This paper argues that the above approach measures the "known knowns" but cannot properly measure the "known unknowns" especially when there exist multiple possible exploit paths and side effects, in which case uncertainty plays an important role. This paper introduces the concept of connectedness, which measures how strong a security bug is connected with different entities, and hence reflects the level of uncertainty of the actual impact and the possibility of novel exploit paths. This concept plays a critical but underappreciated role in severity assessments. This work demonstrates how connectedness influences the severity of security bugs.

Keywords Attack Surface · Cyber Risk Quantification · Philosophy of Cybersecurity · Uncertainty

1 Introduction

Risk and uncertainty analysis plays an important role in security bug severity assessment. Security professionals continuously seek better quantification of the importance of security bugs through analyzing the risk of it being exploited, since "we can not control what we can not measure." [Verendel, 2009]. However, today's severity assessment frameworks do not do well in measuring the uncertainty in the analysis of security bugs.

The practical difference between risk and uncertainty is "that in the former, the distribution of the outcome in a group of instances is known (either through calculation a priori or from statistics of past experience), while in the case of uncertainty this is not true" [Knight and Jones, 2002]. We can hereby define risk as the "known knowns" and uncertainty as the "unknown unknowns" as suggested by Simpson [Simpson, 2024]. For example, "this microservice has a 3 percent chance of melting down and being out of service" is a risk, while "I know that this feature is likely to go wrong, but I do not know how and how likely" is an uncertainty.

Today's severity assessment frameworks usually prioritize two dimensions: (1) the likelihood of a vulnerability being exploited and (2) the severity of its impact. The state-of-the-art scoring system CVSS measures them correspondingly with "exploitability metrics" and "impact metrics" [Mell et al., 2022][2]), and they are only capable of capturing the risk of a single exploit path. When there exist multiple possible exploit paths and multiple possible side effects, CVSS fails to address this situation and forces the user to use a single value (for each sub-item in the metrics) to represent the exploitability of multiple possible exploit paths and the impact of multiple possible side effects - in such cases, it fails to see a security bug as a set of possible security events and measure the effect of the induced uncertainty. In such situations, uncertainty plays an important role in severity analysis; hence, we need a way to measure the impact of uncertainty.

How could we do so? This paper will argue that connectedness is a good measure. First, we need to explain the meaning of "security bug as a set of possible security events", followed by the explanation of connectedness.

*homepage: https://katsuragicsl.github.io/

[2]also see `https://www.first.org/cvss/v4.0/specification-document`

2 Security bug as a set of possible security events

A security bug can have multiple possible exploit paths and multiple possible side effects. Let us consider XSS as an example. In earlier days when XSS prevention techniques were not mature and feasible libraries were not abundant, people used custom filters to sanitize the user input before reflecting it in the response, such as catching and removing dangerous elements in the input, HTML-encode the input, etc. However, these methods were not very effective, due to their bypasses of which many offensive security practitioners must be familiar with. So why was it hard to deter XSS? The main reason is there were many possible exploit paths, even if they were not known until someone published them: the user input reflected in the HTML of the response has connections with many different entities: it could be an HTML element; it could be an attribute of an existing element [3]; it could be inside an existing JavaScript snippet; it is also related to how HTML is parsed in different browsers and different libraries [4].

If we consider the impact of an XSS, we will also see multiple possible side effects, such as cookie stealing [5], phishing [6], cross-origin cookie leakage [7].

Hence, one should not see XSS as a single event, but as a set of possible events such as "the user input creates a dangerous HTML entity", "the user input creates an entity with a dangerous attribute", "the user input creates a dangerous attribute in an existing entity", "the user input creates an entity which confuses the HTML parsing logic", etc.

3 Connectedness

3.1 The more uncertainty the more potential risk

With hindsight, even if we lived in the early 2000s, we should have expected many possible ways to exploit XSS utilizing different objects related to the logic of how a webpage is rendered and how HTML is parsed. For example, even if we do not know that it is possible to exploit it by utilizing event handlers when the input is reflected in the `href` attribute of a `<a>` tag, we should still see it as a possible way. We do not know the actual way to exploit, but we do know that the behavior "user input getting reflected in the response" is connected with all components related to how a webpage is served, in particular the `<a>` tag and its `href` attribute; we just do not know exactly how and how likely it allows an exploit - this is the "known unknown". Once we find out an actual exploit, it becomes a "known known".

The more "known unknowns", the more uncertainty we have, and the more we should expect that some of them will one day turn into "known knowns". Equivalently, if a risk is potential negative security event, we can say "uncertainty" is potential risk.

If the problem of CVSS and our today's view of how severity of a security bug is evaluated is restricting us to only deal with the "known knowns", what should we do to measure the "known unknowns"?

Given a known unknown, we do not know how likely it is going to occur. However, we can estimate how strongly the current issue is related to them: how many different known unknowns are there? How closely are they related? The more and stronger connections it has, we say the higher connectedness it has.

To further illustrate the concept of connectedness, we study a few examples in the next section.

3.2 Examples

1. Input reflection

   Rated as informational usually (for example by Tenable[8]. An input reflection can be seen as "a (reflected) XSS that did not make it". It connects to everything that a reflected XSS connects to, hence it has the same known unknowns, except that none of them has turned into an actual risk. However, as discussed in 3.1, there are multiple potential risks.

2. Missing referrer policy

[3] such as the href attribute of an a tag

[4] for example see https://portswigger.net/research/bypassing-dompurify-again-with-mutation-xss

[5] which is usually the main concern.

[6] by disguising a phishing attempt as a genuine content served by a trusted domain, through its url.

[7] in case the cookie of the another site is scoped to parent domain, and the vulnerable site is another subdomain. This is hypothetical, but still a possible side effect, although not a strong one.

[8] `https://www.tenable.com/plugins/was/114135`

> Rated as informational usually (for example by Tenable[9]). By default the policy will be `strict-origin-when-cross-origin` when the referrer policy header is not set. Comparing to the above issue, the possible exploit paths and the possible side effects are much lesser - obviously it depends on the threat. For example we assume that "leaking" the referrer to the same origin is not problematic; we also do not consider browser errors in managing the same origin policy (since in that case we could not do much about it as a webapp engineer; and much worse things would happen). If we assume that "leaking" the referrer to the same origin is a problem, for example path A and path B are served by different microservices, and for some reasons we do not want the microservices know which requests are directed from another microservice, then this referrer leak would be a problem. In usual settings, there are not many ways one can manipulate this issue. Hence its level of connectedness is comparatively lower than the above issue.[10]

Both are usually treated as "informational" issues, inducing more or less the same level of risk, but they have very different levels of connectedness.

If a person had to prioritize the triage/fix of one over that of the other, the input reflection issue should be prioritized for having more potential risk.

3.3 Limitations

The examples above illustrated that one of the limitations of the analysis of connectedness is that it depends on the threat model of the system, particularly what threat scenarios we do not see as risks.

4 Conclusion

As Simpson [Simpson, 2024] mentioned, "we should question assumptions that quantitative solutions are necessarily superior to qualitative ones...; we should ask for confidence intervals, assumptions, limitations...it is imperative that lessons learnt...with regards to the relationships that exist between risk (known knowns), uncertainty (known unknowns) and unawareness (unknown unknowns) be leveraged to help ensure that the increasing quantification and modelling of cyber risk is put on somewhat more solid ground than is currently the case.", we should be aware that we currently lack a rigorous way to quantify the connectedness, not to mention its statistical attributes such as confidence intervals. Also, while it helps us to evaluate the "known unknowns", it does not help us to tackle the "unknown unknowns": our analysis of connectedness on a given security bug could also be faulty - it is possible for us to miss significant exploit paths and side effects, and mistakenly believe that a particular security bug is of low connectedness. However, "unknown unknowns" in severity analysis should not and could not be tackled with the methods of evaluation [11], but with something in a higher level of the abstraction ladder, such as continuous threat modelling, to (hopefully) reduce the "unknown unknowns".

However, it does not hurt the fact that connectedness helps assess the "known unknowns" and should be included in the severity analysis of security bugs.

What we should take away from this paper is that uncertainty can strongly affect the severity of a security bug, and since we cannot estimate the likelihood of a given known unknown, we could instead measure the connectedness to reflect the amount of uncertainty.

References

Vilhelm Verendel. Quantified security is a weak hypothesis: a critical survey of results and assumptions. In *Proceedings of the 2009 Workshop on New Security Paradigms Workshop*, NSPW '09, page 37–50, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605588452. doi:10.1145/1719030.1719036. URL `https://doi.org/10.1145/1719030.1719036`.

F.H. Knight and D.E. Jones. *Risk, Uncertainty and Profit*. Warriors (Washington, D.C.). Beard Books, 2002. ISBN 9781587981265. URL `https://books.google.com.hk/books?id=Im2dnQAACAAJ`.

[9] `https://www.tenable.com/plugins/was/98527`

[10] there are some possible side effects that were not mentioned, for example one could imagine that if there exists a bug in the server which causes DoS when a long referrer header is received, then the missing referrer policy might increasing an epsilon of chance of such DoS happens. But the connection between the missing referrer policy issue and this possible behavior is very weak (since the chance is small and such DoS could have been triggered by other means more efficiently), so it does not contribute much to the level of connectedness. I hope the reader is convinced that overall the input reflection issue still has higher level of connectedness.

[11] since no matter what evaluation method one uses, by definition there will always be "unknown unknowns".

Andrew Simpson. Into the unknown: the need to reframe risk analysis. *Journal of Cybersecurity*, 10(1):tyae022, 11 2024. ISSN 2057-2085. doi:10.1093/cybsec/tyae022. URL https://doi.org/10.1093/cybsec/tyae022.

Peter Mell, Jonathan Spring, Dave Dugal, Srividya Ananthakrishna, Francesco Casotto, Troy Fridley, Christopher Ganas, Arkadeep Kundu, Phillip Nordwall, Vijayamurugan Pushpanathan, Daniel Sommerfeld, Matt Tesauro, and Christopher Turner. Measuring the common vulnerability scoring system base score equation, 2022-11-15 05:11:00 2022. URL https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935413.