A Historical and Statistical Study of the Software Vulnerability Landscape

Assane Gueye

assaneg@andrew.cmu.edu

Assistant Teaching Professor

Carnegie Mellon University Africa

Kigali, Rwanda

Carnegie Mellon University Africa

Peter Mell

peter.mell@nist.gov

Senior Computer Scientist

National Institute of Standards and Technology

Gaithersburg, MD-USA

NIST National Institute of Standards and Technology

The Seventh International Conference on Advances and Trends in Software Engineering

SOFTENG 2021

April 18, 2021 to April 22, 2021 - Porto, Portugal

IARIA

Who Am I?

Assane Gueye

ASSISTANT TEACHING PROFESSOR

Department(s): CMU-Africa

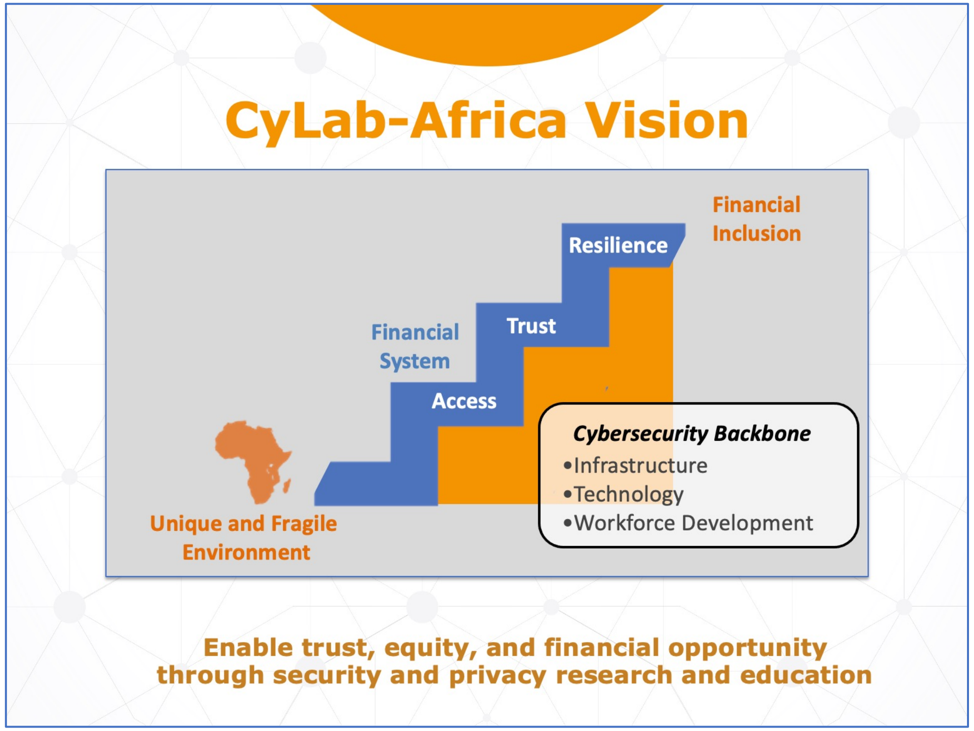
Email: assaneg@andrew.cmu.edu

Assane Gueye joined Carnegie Mellon University Africa on **August 1st, 2020**. Prior to joining CMU Africa, he was a faculty member at the **ICT Department at the University Alioune Diop of Bambey, Senegal**, where he also leads the research group "Technologies de l'Information et de la Communication pour le Développement" (TIC4Dev). Gueye also holds a guest researcher position with the **National Institute for Standards and Technology, Gaithersburg, Maryland, USA**.

Assane completed his **Ph.D. in electrical engineering and computer science from UC Berkeley** in March 2011. He holds a **Master's degree in communication systems engineering from Ecole Polytechnique Fédérale de Lausanne**, Switzerland.

His research focuses in two main areas**: performance evaluation and security of large-scale communication systems**, and **information and communication technologies for development (ICT4D)**. Assane is a Fellow of the Next Einstein Forum (Class of 2016). In 2019 he was nominated as a member of the European Alliance for Innovation (EAI) inaugural Fellow Class.

Research Activities



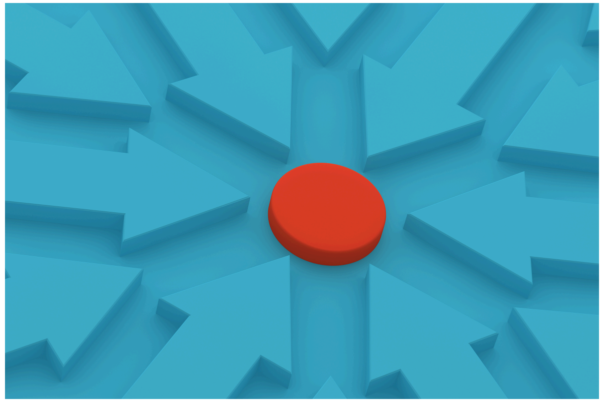
https://www.africa.engineering.cmu.edu/research/cylab.html

Motivation

Understanding the landscape of software vulnerabilities is key for developing effective security solutions.

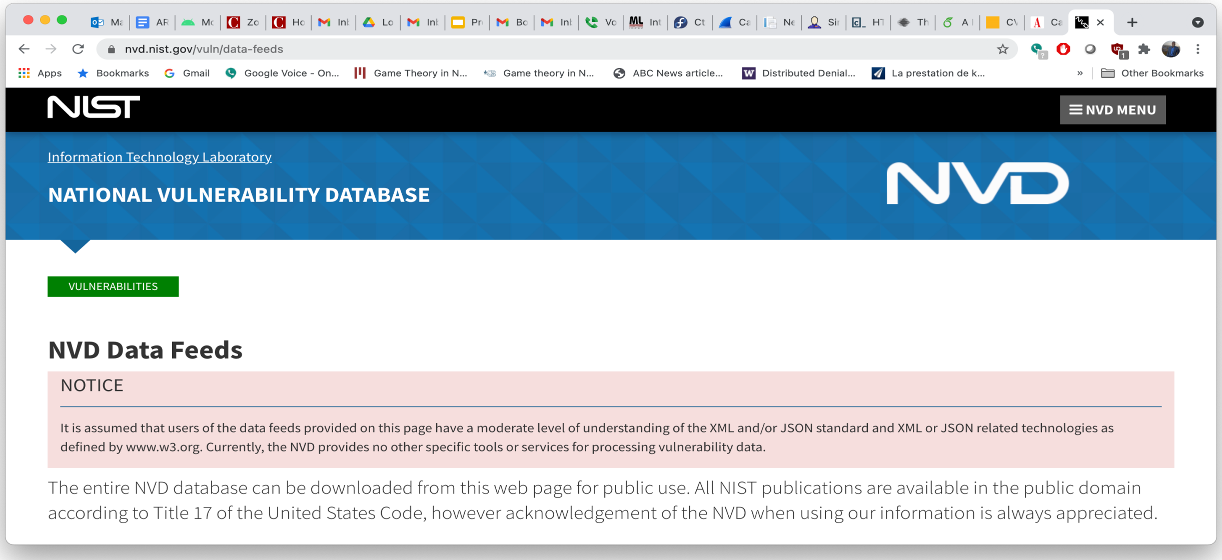
If the most significant of these types can be identified, developers of programming languages, software, and security tools can focus on preventing them

➔diminish the quantity and severity of newly discovered vulnerabilities

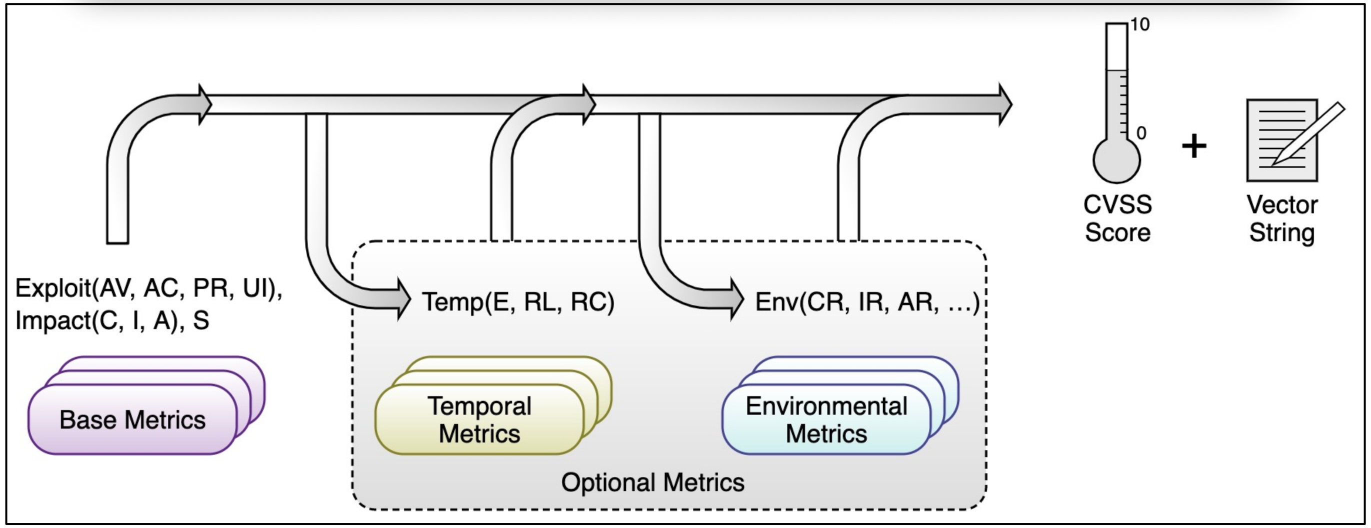


Approach (1)

Common Vulnerabilities Scoring System (CVSS) Dataset



CVSS v3 Metrics	Metric Values
Attack Vector (AV)	Network (N), Adjacent (A), Local (L), Physical (P)
Attack Complexity (AC)	Low (L), High (H)
Privileges Required (PR)	None (N), Low (L), High (H)
User Interaction (UI)	None (N), Required (R)
Scope (S)	Unchanged (U), Changed (C)
Confidentiality (C)	High (H), Low (L), None (N)
Integrity (I)	High (H), Low (L), None (N)
Availability (A)	High (H), Low (L), None (N)



CVSS:3.1. /AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:N

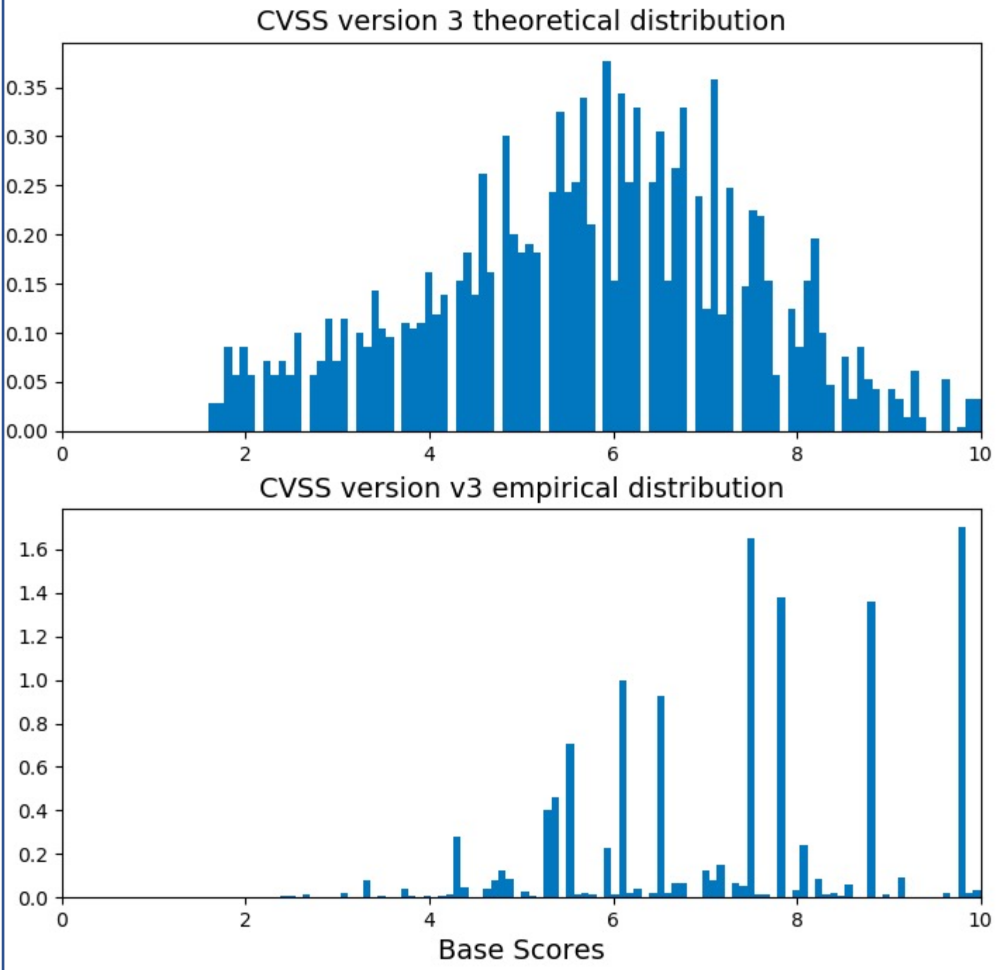
Approach (2)

- **Experiments**
  - Score (numerical) distributions
  - Metric values distributions
  - Relative rankings of the most frequent metric values
  - The most prevalent patterns of co-occurrence of the metric values

Results and Analysis (1)

- **Score Distribution**

CVSS:3.1.   /AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:N



Producing Numerical Score

Metric	Metric Value	Numerical Value
Attack Vector / Modified Attack Vector	Network	0.85
	Adjacent	0.62
	Local	0.55
	Physical	0.2

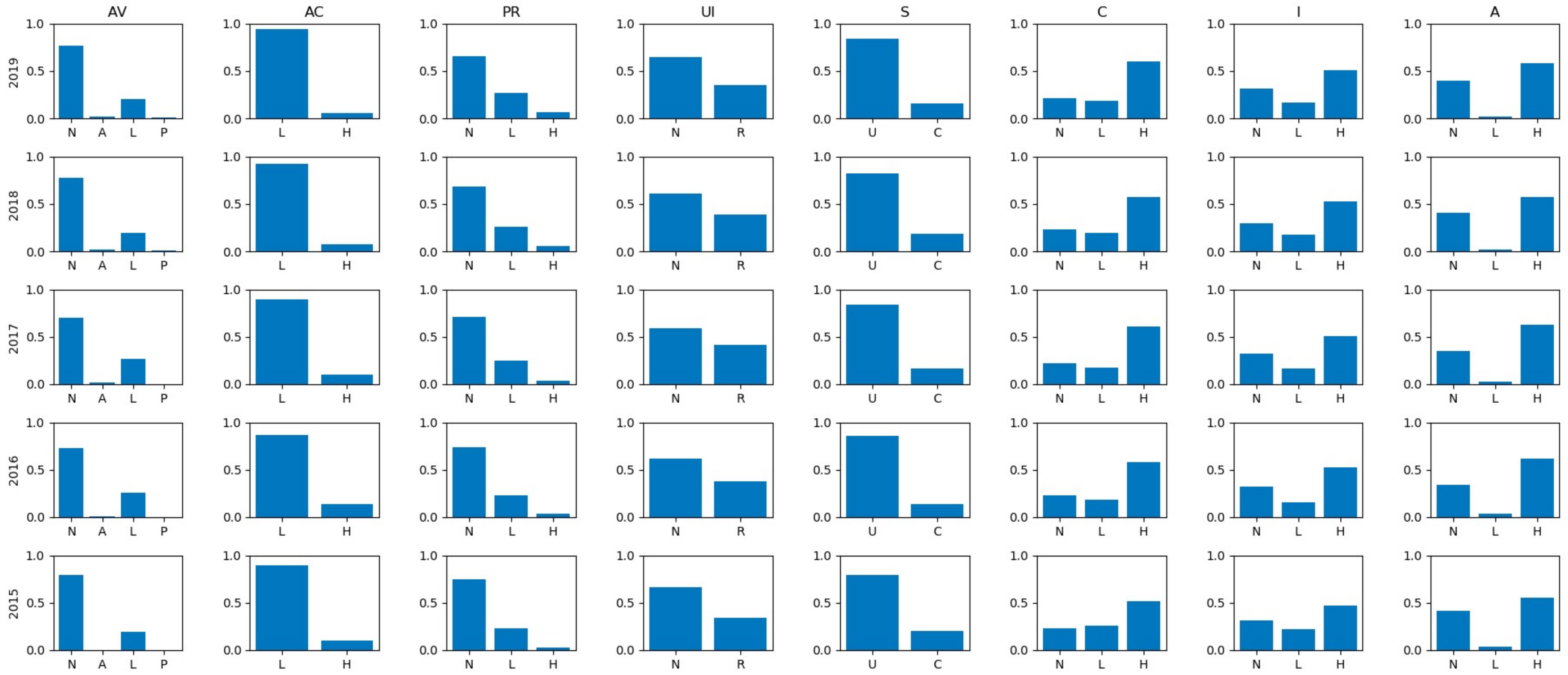
ISS =	1 - [(1 - Confidentiality) × (1 - Integrity) × (1 - Availability)]
Impact =	
If Scope is Unchanged	6.42 × ISS
If Scope is Changed	7.52 × (ISS - 0.029) - 3.25 × (ISS - 0.02)^{15}
Exploitability =	8.22 × AttackVector × AttackComplexity ×
	PrivilegesRequired × UserInteraction
BaseScore =	
If Impact \<= 0	0, *else*
If Scope is Unchanged	Roundup (Minimum [(Impact + Exploitability), 10])
If Scope is Changed	Roundup (Minimum [1.08 × (Impact + Exploitability), 10])

Some Insights:

Predominance of certain vectors (groupings of vulnerability characteristics) in the real world!

Results and Analysis (2.1)

- **Metric Values Distribution**



Results and Analysis (2.2)

- **Metric Values Distribution**

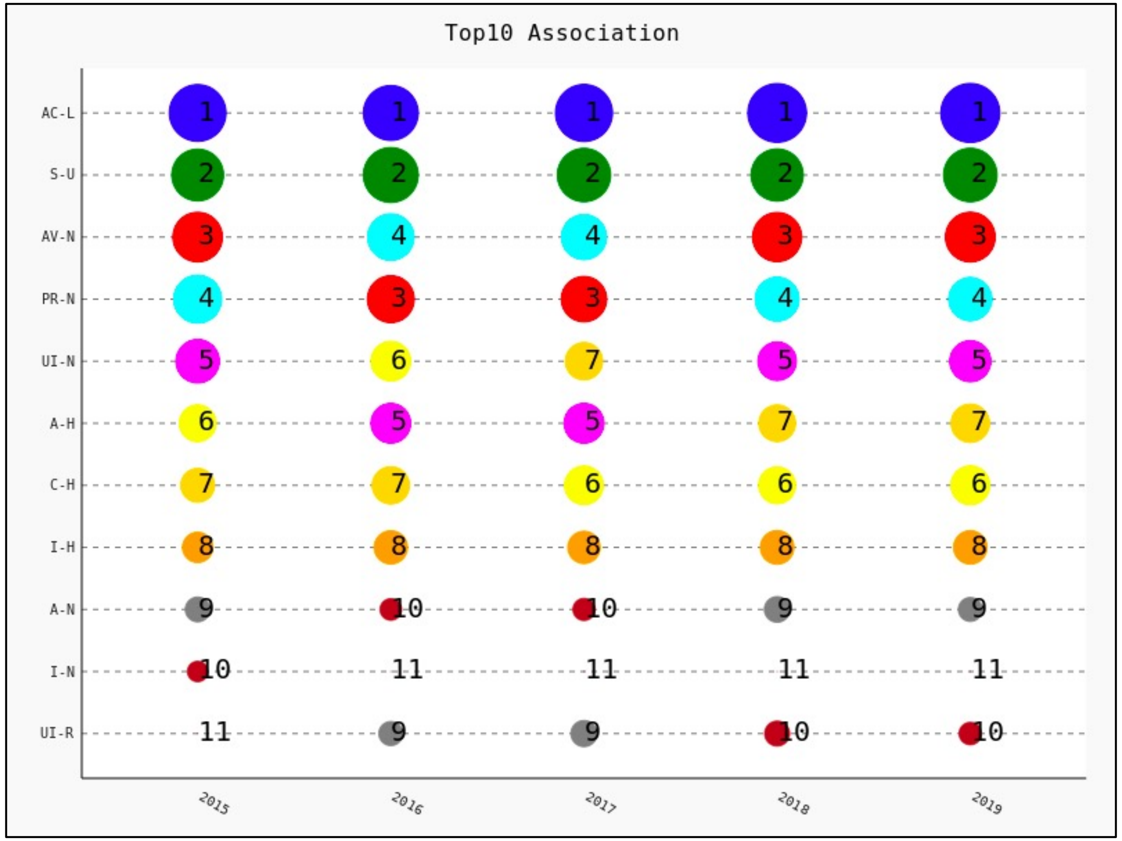
Metric	Values
Attack Vector (AV)	**Mostly network (N)**, some local (L)
Attack Complexity (AC)	**Low (L)**
Privilege Required (PR)	**Mostly none (N)**, sometime low (L)
User Interaction (UI)	**Dominantly not required (N)**
Scope (S)	**Unchanged (U)**
Confidentiality (C)	Dominated by high (H)
Integrity (I)	Dominated by high (H)
Availability (A)	Dominated by high (H)

Some Insights:

Some metrics values have dominated the landscape!

Results and Analysis (3)

▪ Metrics Values Ranking (Top 10 over the years)



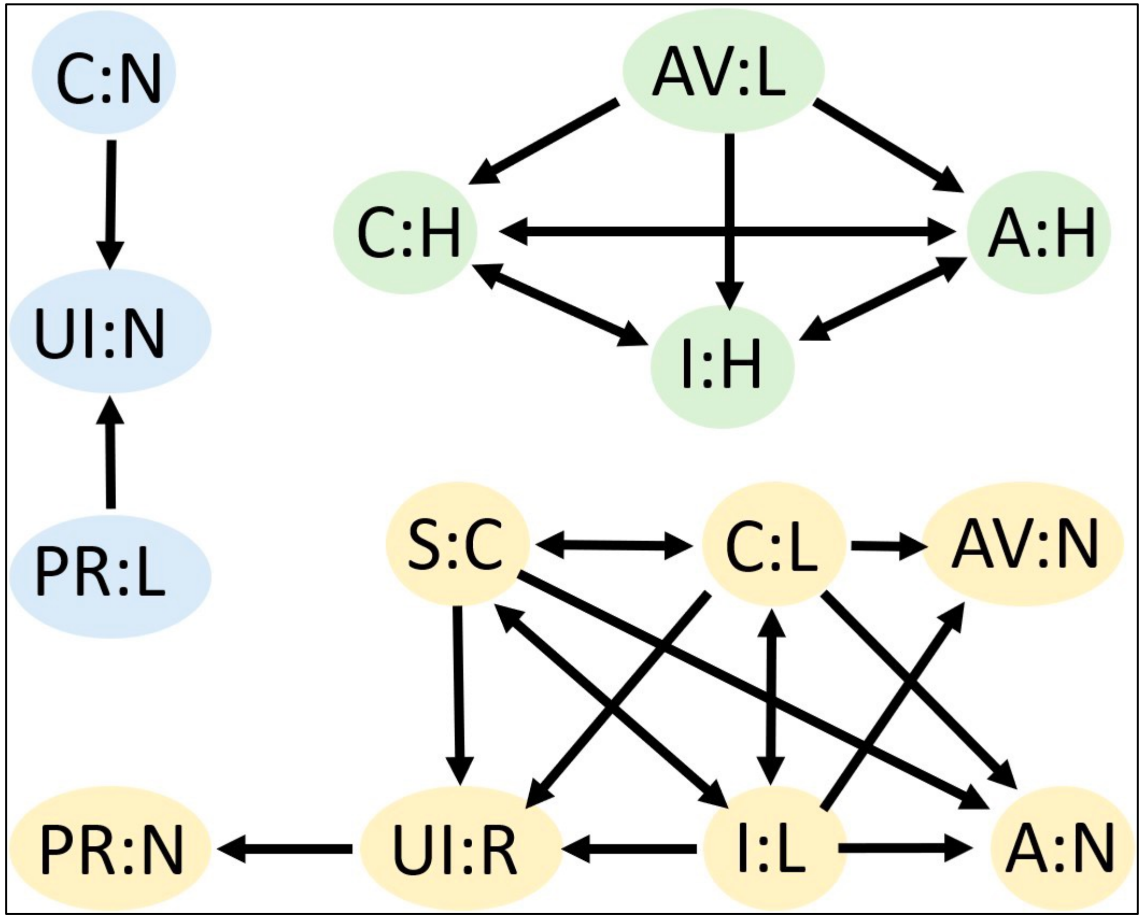
The size of the circle is proportional to the number of times that metric value appeared in a score in that year.

Some Insights:

- Same top 10 values appeared from 2016 to 2019 (confirming domination by some values)
- Metrics values ranked almost the same over the years
  - Top 2 are constant and in the same order over the time period
  - Top 4 and the bottom 4 (including the 11th appended value) are also constant

Results and Analysis (4)

- **Associations**



Some insights:

- Impact metrics (C:H), (I:H), and (A:H) form a clique. Whenever one of the metrics is highly impacted the others are also highly impacted.
- (S:C), (C:L) and (I:L) form a clique.
  When clique values are true:
  - AV is likely to be network (AV:N),
  - A is likely not impacted (A:N),
  - User interaction required (UI:R).

  When (UI:R), no privileged (PR:N) is needed.
- When C is not impacted (C:N) or PR is low (PR:L) UI is likely not needed (UI:N)

Discussion/Conclusion

Observations:

- Vulnerability landscape constantly dominated by a few vulnerability types
- Overwhelming majority of software vulnerabilities exploitable over the network
- Most vulnerabilities requiring no/low sophistication to be exploited
- No spill-over effect for attacks

Conclusion:

- As a community, we have **not been successful fixing** what seems to be **the most prevalent software vulnerabilities**
- Either:
  - We are **incapable of fixing them**
  - We are **focusing on the wrong ones** (i.e., our security metrics are flawed)
- In either case **we need to “stop and think”**: about the ways we are developing software and/or the methods we use to identify vulnerabilities

Thank you!

Contact: assaneg@andrew.cmu.edu