

SESSION ID: TECH-F01

Vulnerability Management Nirvana: A Study in Predicting Exploitability

Kymberlee Price

Senior Director of Operations
Bugcrowd
@Kym_Possible

Michael Roytman

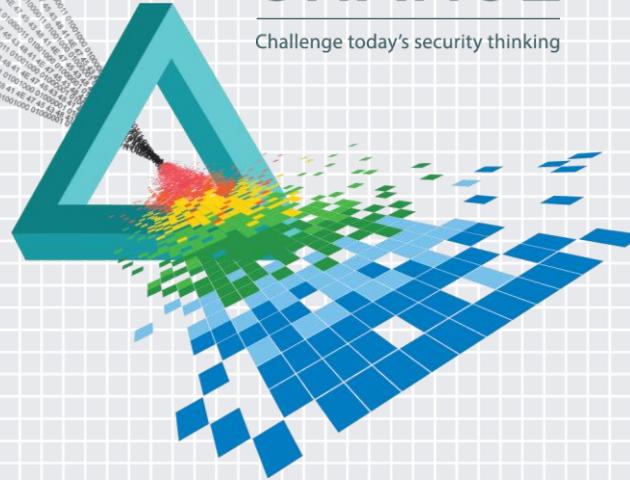
Senior Data Scientist
Risk I/O
@MRoytman

David F. Severski

Mgr., Information Security Program
Seattle Children's Hospital
@DSeverski

CHANGE

Challenge today's security thinking

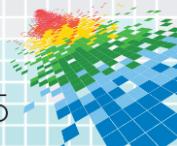


Why Prioritization Matters

- ◆ You have 150 vulnerabilities open with CVSS 6.8 and above
- ◆ Your inbound new vulnerabilities average 15 dev tasks per week, from both internal and external sources
- ◆ What do you fix first?

TOP PRIORITIES

1. _____
1. _____
1. _____
1. _____
1. _____



Historical Research: Prioritizing Product Vulnerabilities

- ◆ Sources of vulnerabilities
 - ◆ Internal Security Research Group
 - ◆ External Security Researchers
 - ◆ Third Party Libraries/OSS Disclosures
- ◆ Developers would prioritize on CVSS v2 Base Score
- ◆ Limitations in CVSS became apparent

Historical Research: Prioritizing Product Vulnerabilities

- ◆ Applied custom criteria for extended CVSS fields
- ◆ Weighted extended CVSS fields to adjust base CVSS
- ◆ Defined priority bands with SLA for remediation
- ◆ Automated the priority calculations – the only manual requirement was for the CVSS score to be entered when the bug was logged, which was part of existing SOPs

Our Path towards Nirvana

- ◆ Using CVSS for prioritization today
- ◆ Alternative prioritization models
- ◆ Our research
- ◆ Comparative data & results
- ◆ Conclusions & How to Apply

RSA® Conference 2015

San Francisco | April 20-24 | Moscone Center

CURRENT VULNERABILITY PRIORITIZATION MODELS



CVSSv2: The Tool We Have

- ◆ Open industry standard
- ◆ Maintained and regularly updated
- ◆ Modular – base, temporal, and environmental components
- ◆ Objective... mostly
- ◆ False negatives 
- ◆ Base score is non-predictive single point in time measure
- ◆ Few companies use update Temporal or Environmental scores
- ◆ No indication of historical attack patterns
- ◆ Misused as prioritization tool, designed to convey vulnerability characteristics

The Common Vulnerability Scoring System (CVSS) provides an open framework for communicating the characteristics and impacts of IT vulnerabilities.

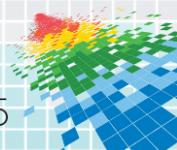
CVSSv3: The Tool We Need?

- ◆ Changes Authentication to Privileges Required
- ◆ Removes “medium” option for Access Complexity
- ◆ Adds new dimensions to measure User Interaction and Scope
 - ◆ User Interaction used to be considered part of Attack Complexity
 - ◆ Scope documentation is confusing

Components run within a scope that authorizes the actions they can perform and the resources they can access. An example of an authorization scope is the user list and the privileges granted to users of an operating system. A separate authorization scope could be contained within a database application that runs on the operating system. If a successful exploit only impacts resources within the scope of the vulnerable component, then Scope is Unchanged. If a successful exploit impacts resources outside the scope of the vulnerable component, then Scope is Changed.

CVSSv3: The Tool We Need?

- ◆ Still a non-predictive single point in time measure
- ◆ Temporal & Environmental fields now impact Base Score
 - ◆ Base score is worst possible outcome – completing temporal and environmental fields only has potential to lower base score
- ◆ Framework transition pain
 - ◆ Retooling your systems for new format
 - ◆ No standardization to compare v2 vulns to v3 vulns
- ◆ Still designed for communication of vulnerability characteristics



Exploit Index: Is This Nirvana?

- ◆ Exploitability Index (Microsoft, 2008)
 - ◆ Intended to provide customers with more granularity to improve risk assessment and patch prioritization
 - ◆ Determining exploitability is heavily dependent on human researchers, creating scale and skill limitations

July 2013, 5 years after Exploit Index launch:

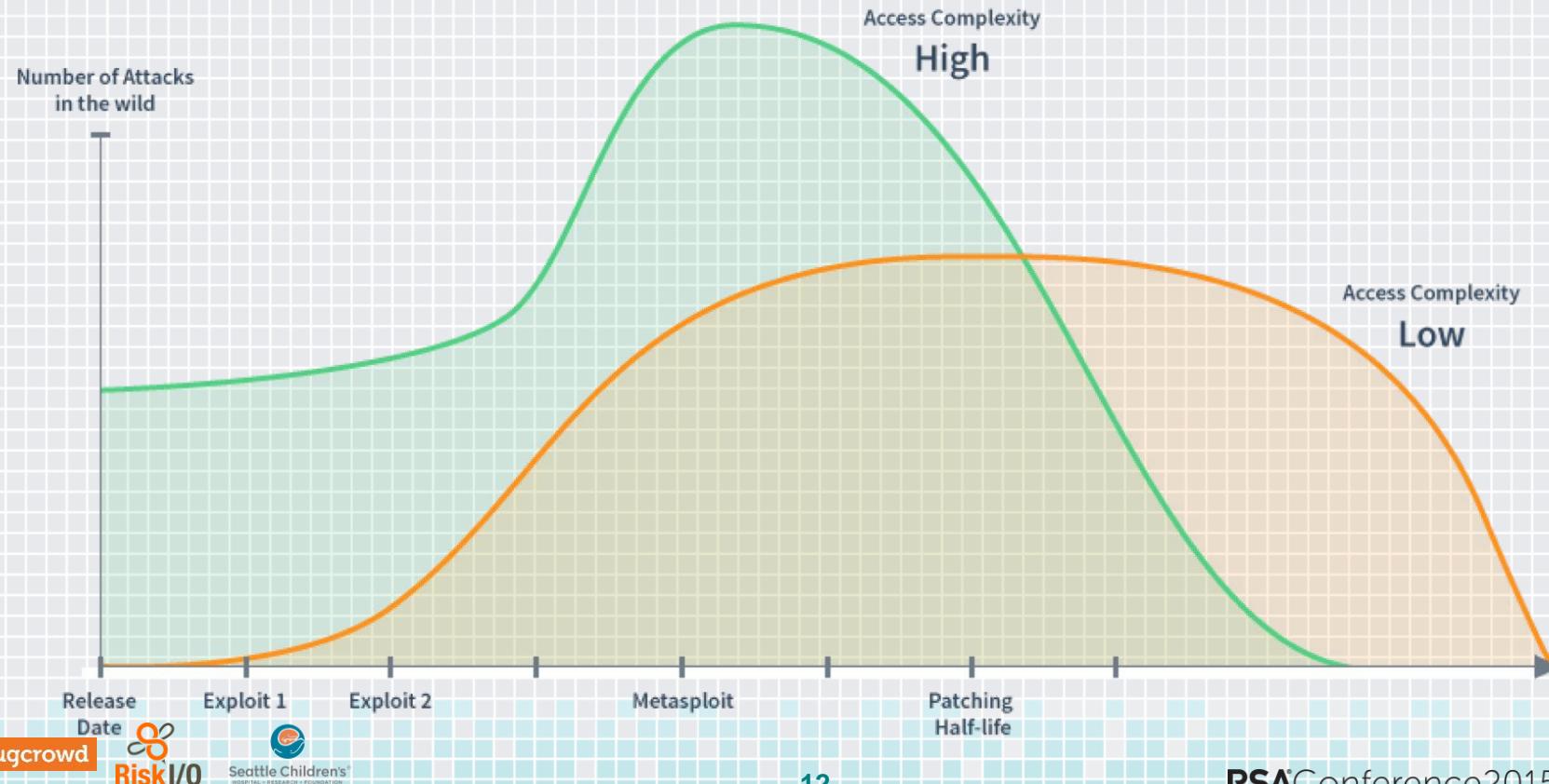
- ◆ “While no exploit surfaced for a vulnerability within 30 days of security bulletin release, it does not mean that the vulnerability could not have been exploited researchers or attackers may just have been prioritizing other vulnerabilities instead”²

What Else Is There?

- ◆ Indicators of Badness
 - ◆ Exploit Presence in Metasploit
 - ◆ Exploit Presence in Canvas
 - ◆ Known Public vs Private Exploits
 - ◆ Attack Vectors
- ◆ That's a lot of threat intelligence feeds to monitor and investigate in real time on every vulnerability you've got logged.



One Model to Rule Them All



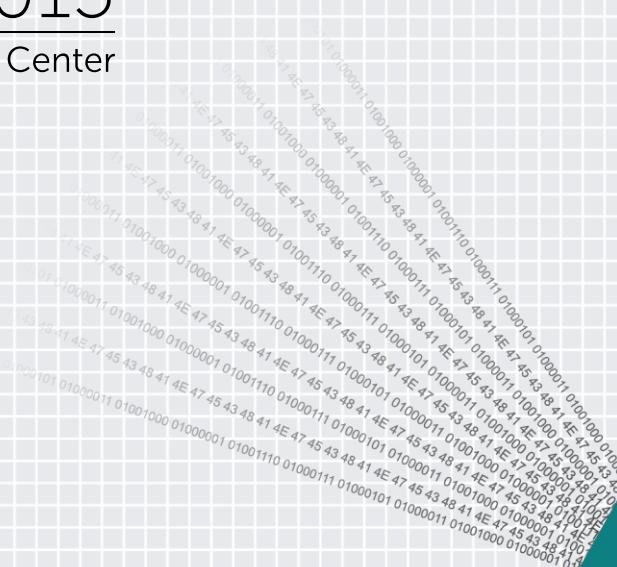
One Model to Rule Them All

- ◆ Time Series Data
 - ◆ Release Date
 - ◆ First Exploit Released
 - ◆ Weaponized Exploit Released (Metasploit)
 - ◆ Average Patch Time (Qualys Half-Life)
- ◆ Attack Data
 - ◆ Attacks Detected
 - ◆ Successful Attacks Detected
 - ◆ Impactful Breaches Detected
- ◆ Categorical Data
 - ◆ Complexity
 - ◆ Access Vector
 - ◆ Impact

RSA® Conference 2015

San Francisco | April 20-24 | Moscone Center

NIRVANA RESEARCH



Historical Research: Predicting Exploitability – Big Data Time

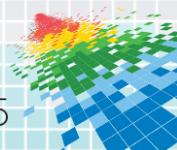
50,000,000 Live Vulnerabilities



1,500,000 Assets



2,000 Organizations



Historical Research: Predicting Exploitability – Big Data Time

#RSAC

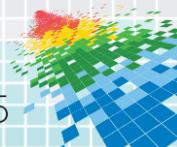
150,000,000 Breaches

Baseline ALLTHEthings!!

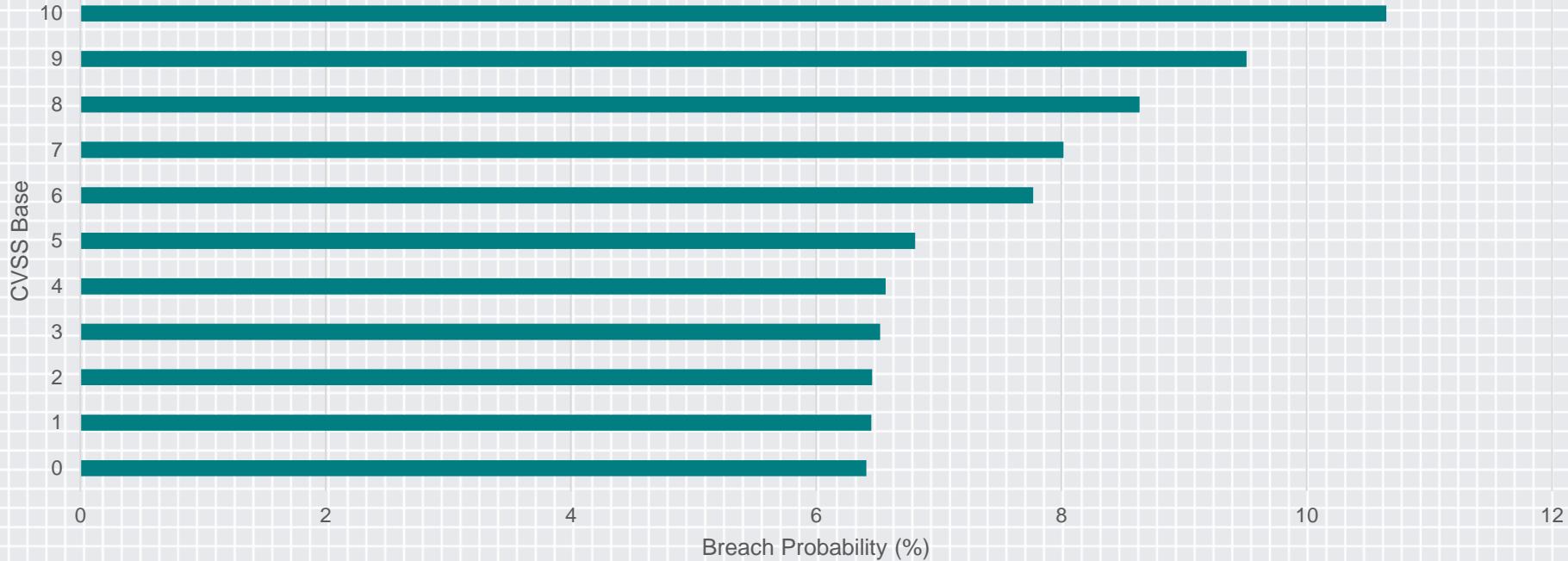
- ◆ Probability
(You Will Be Breached On A Particular Open Vulnerability)?

$$\frac{\text{(Open Vulnerabilities | Breaches Occured on Their CVE)}}{\text{Total Open Vulnerabilities}}$$

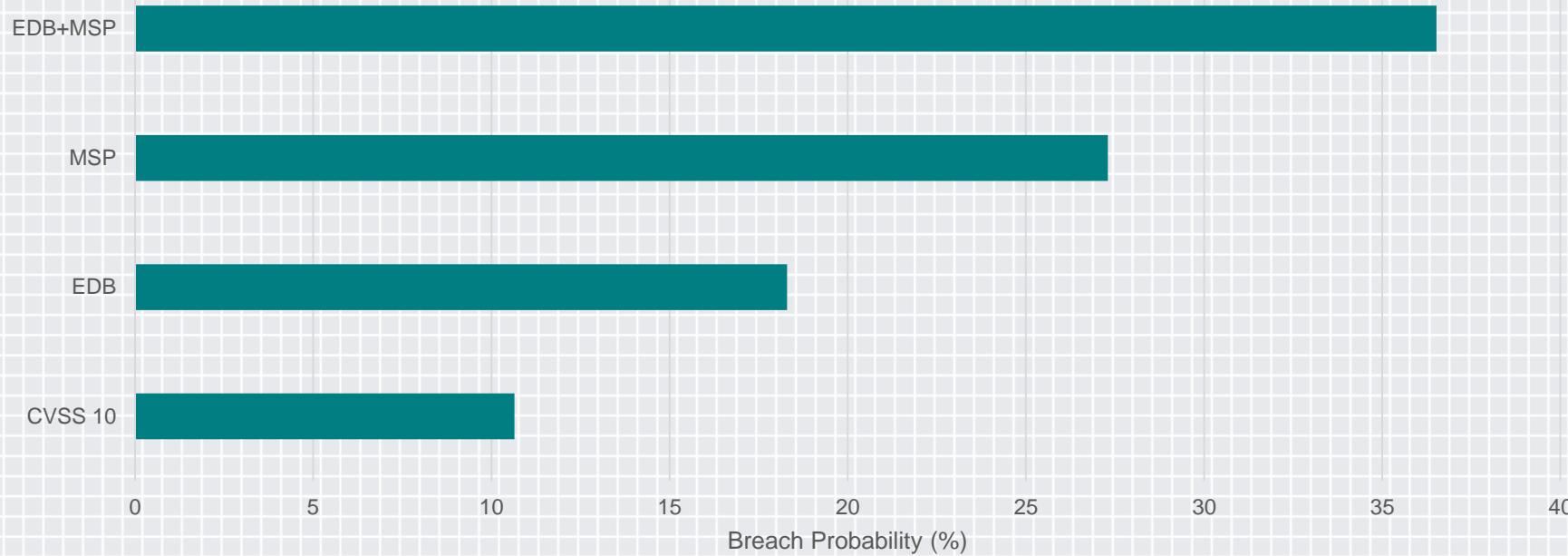
6%

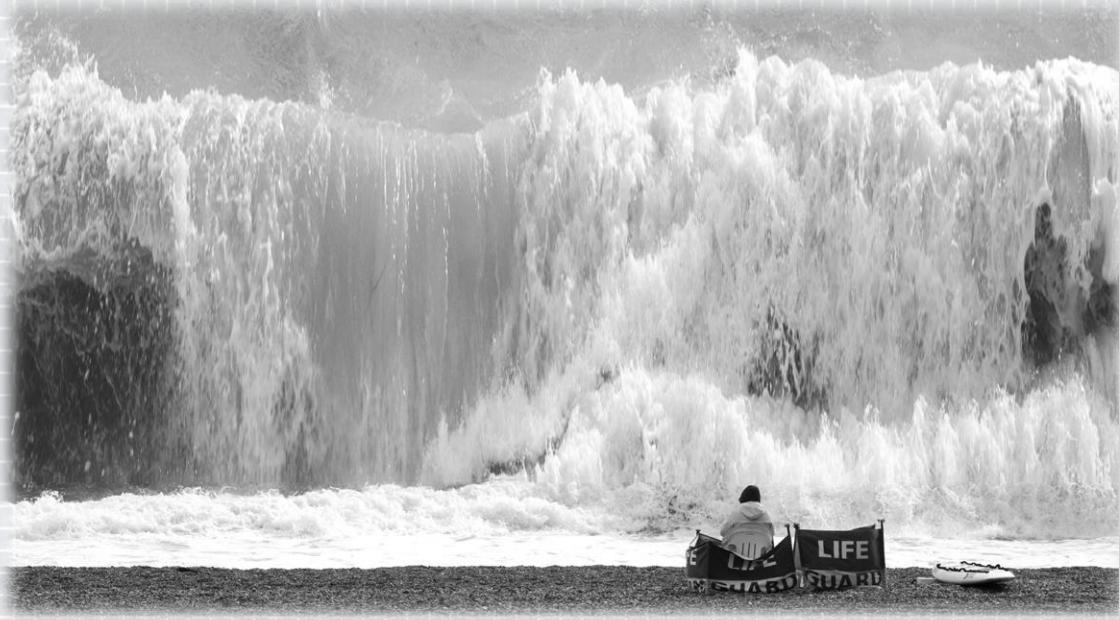


Probability a Vulnerability Having CVSS > X Has Observed Breaches



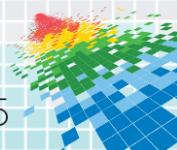
Probability a Vulnerability Having Property X Has Observed Breaches





VulnPryer: A Survival Strategy for Vulnerability Management

Triaging and answering the question – “What should enterprise defenders fix first?”



Design Requirements

- ◆ Use commonly accessible data
- ◆ Customizable for our threat scenarios
- ◆ Easy to produce
- ◆ Must adjust to changing information

We Have the Data...We Can Rebuild It

National Vulnerability
Database

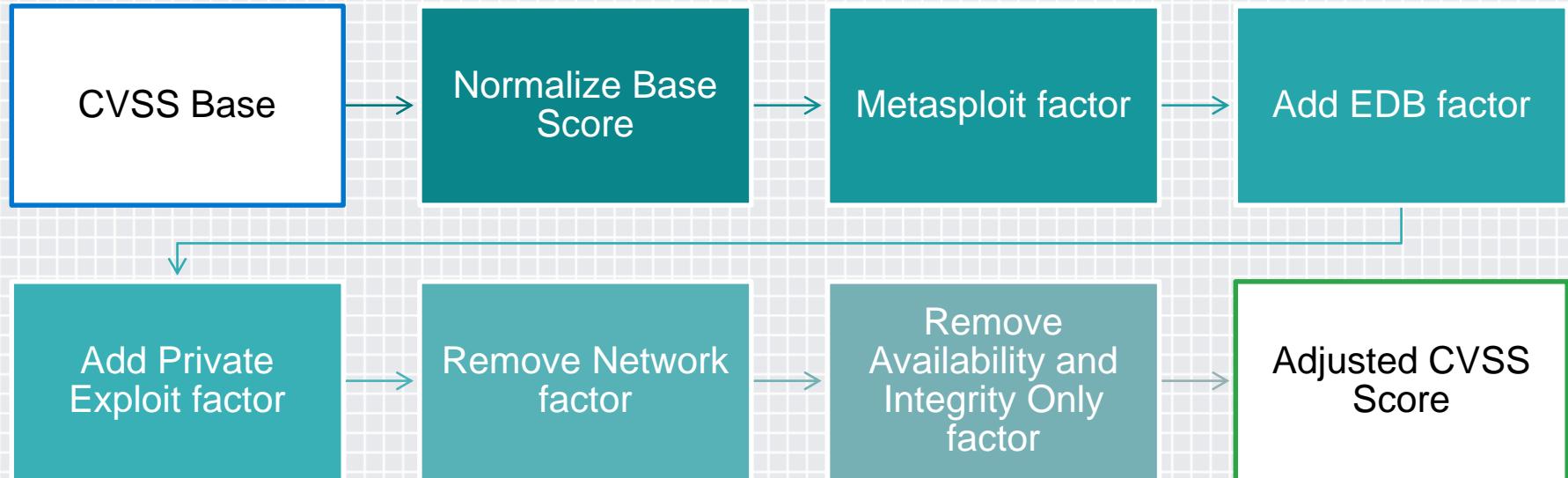
Network Security Posture
Analysis

Sources of Data

Commercial Vulnerability
Feeds

Internal Asset Valuations

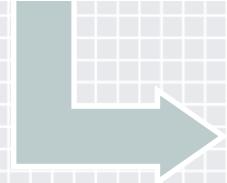
VulnPryer Flow



Automate ALLTHEthings!!

Daily Generation of
Vulnerability
Reference Library

- AWS Data Pipeline
- Risk Based Security VulnDB
- VulnPryer



Network Analysis
with Customized
Severities

- RedSeal
- Combine Asset Values with Actual Network Configuration

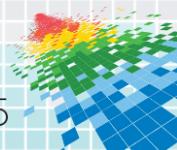
Internal Reporting
and Prioritization

- Internal Dashboards
- Tableau and R for Presentation

Fix **this**
before
that

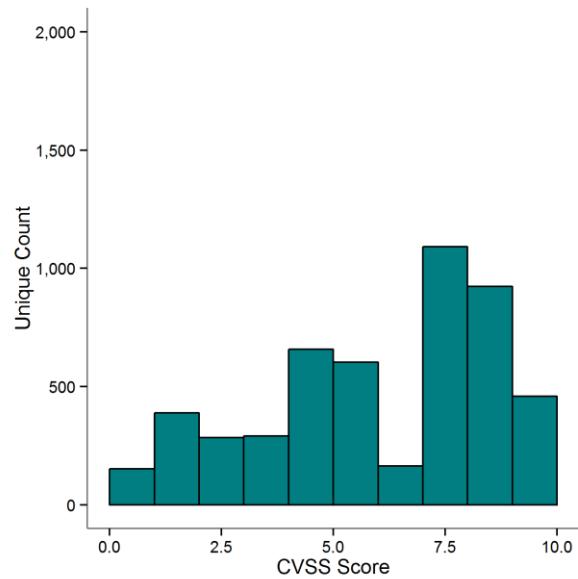
A Few Individual Results

- ◆ Mean adjustment: -1.7 (24% decrease)
- ◆ Maximum increase: 3.3 (112% increase)
- ◆ Maximum decrease: 5.6 (99% decrease)
- ◆ CVE-2014-0160 (Heartbleed) rescored from 5.0 to 8.3
- ◆ 4% of vulnerabilities reduced to CVSS 0
- ◆ Approximately 18% reduction in network based risk scores

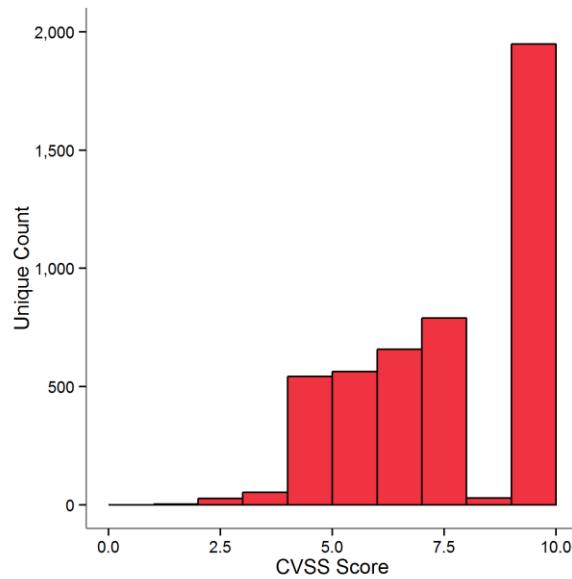


Results Over Our Specific Population

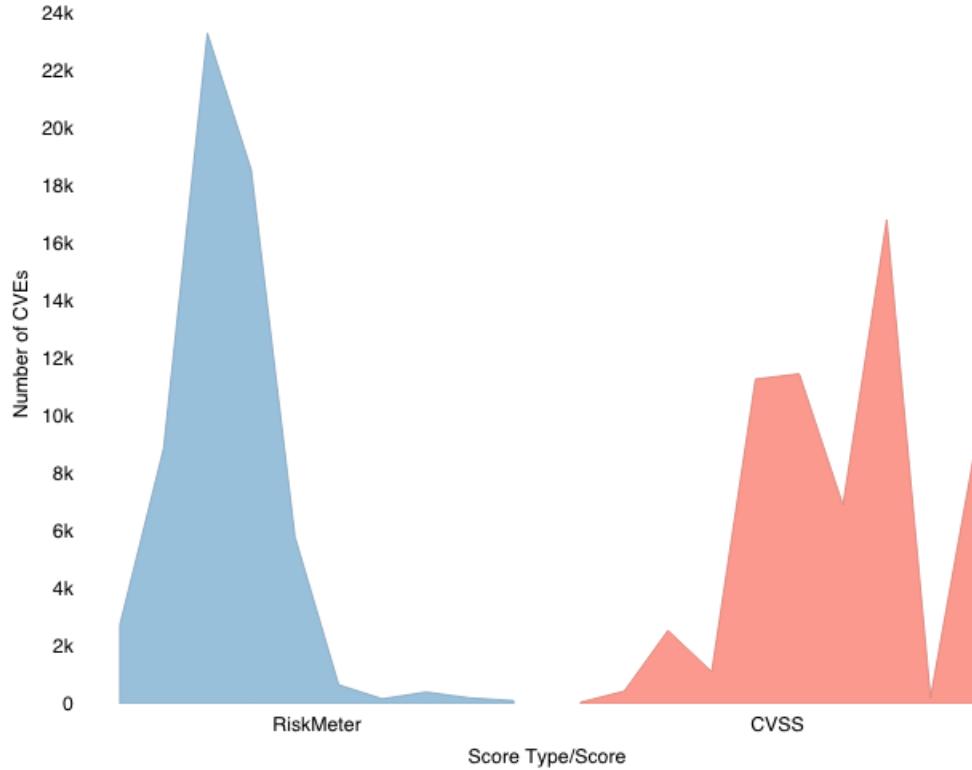
Vulnpryer-adjusted CVSS



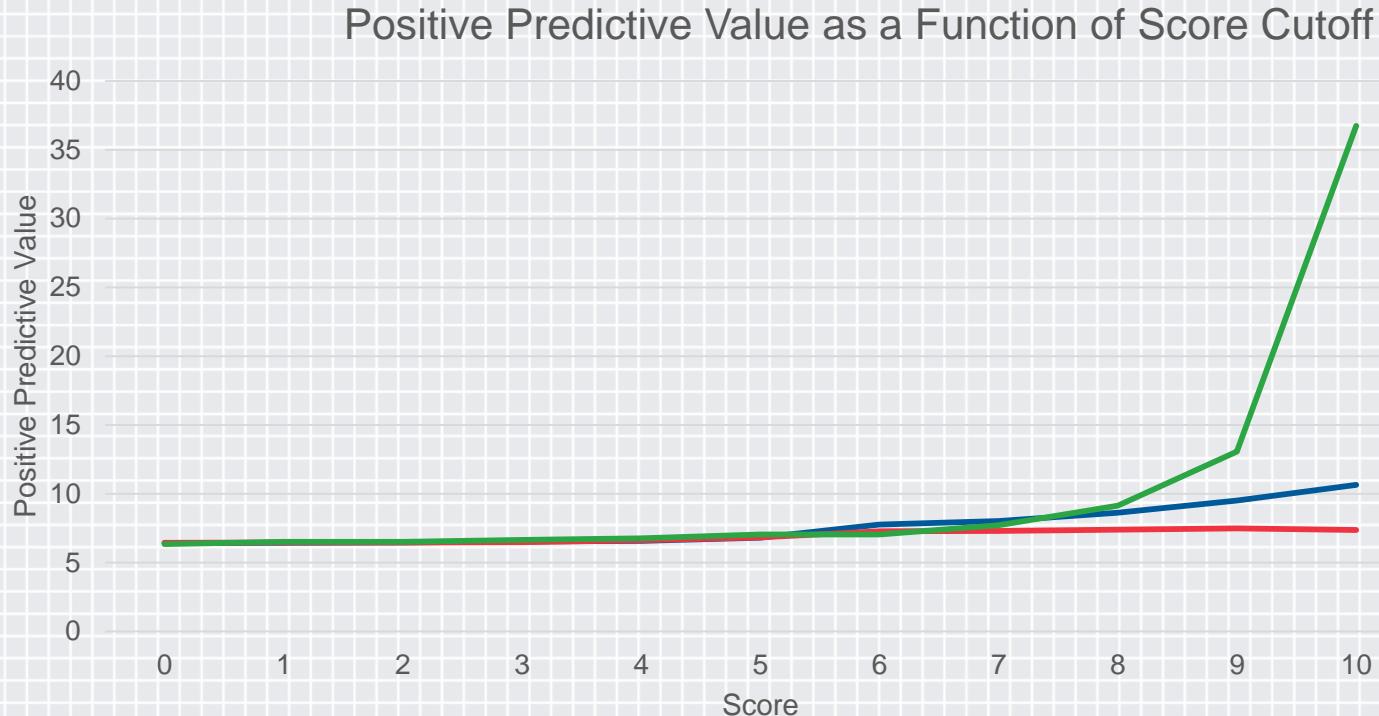
Base CVSS



Results: Comparison with Risk I/O RiskMeter



Nirvana Research: Results



Future Directions towards Nirvana

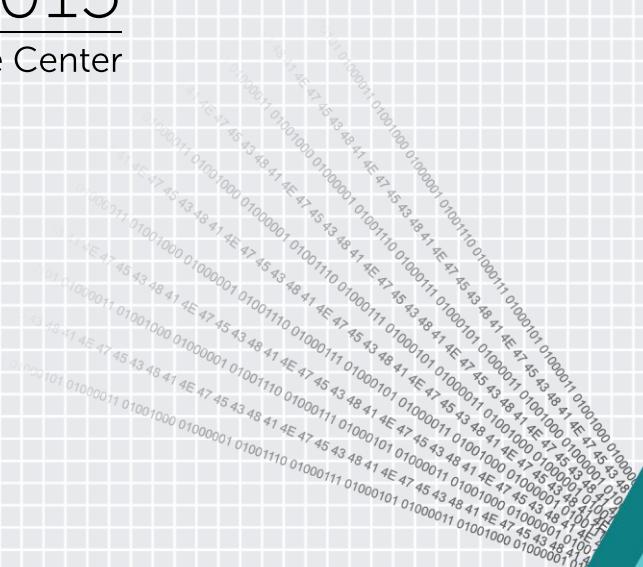
- ◆ There's always another layer...
- ◆ Performance Optimization
- ◆ Reporting
 - ◆ Alerting on Changes to Scores
 - ◆ Sample Reporting Templates (Tableau and/or R)
- ◆ More Flexible Formula Changes
- ◆ Generalize the Pythonic Framework for Other Use Cases
- ◆ Analysis of Nirvana model in product security environment



RSA® Conference 2015

San Francisco | April 20-24 | Moscone Center

CONCLUSIONS & HOW TO APPLY



Conclusions

- ◆ Base CVSS as sole criteria has serious known limitations
- ◆ Readily available tools and data sources can be used to help you focus on what matters to **you and your organization**
- ◆ It is both **possible** and **practical** to stay abreast of changing vulnerability risk to drive timely resource allocation decisions

Searching for Nirvana at Home

- ◆ Easy to get started, just add
 - ◆ Database of vulnerability features
 - ◆ Your vulnerabilities
 - ◆ A little bit of code
- ◆ Code provided for you!
 - ◆ VulnPryer in both Python and R versions
 - ◆ Fully functional example automatic deployment via Chef and AWS



RSA® Conference 2015

San Francisco | April 20-24 | Moscone Center

SESSION ID: TECH-F01

QUESTIONS

Kymberlee Price

Senior Director of Operations
Bugcrowd
@Kym_Possible

Michael Roytman

Senior Data Scientist
Risk I/O
@MRoytman

David F. Severski

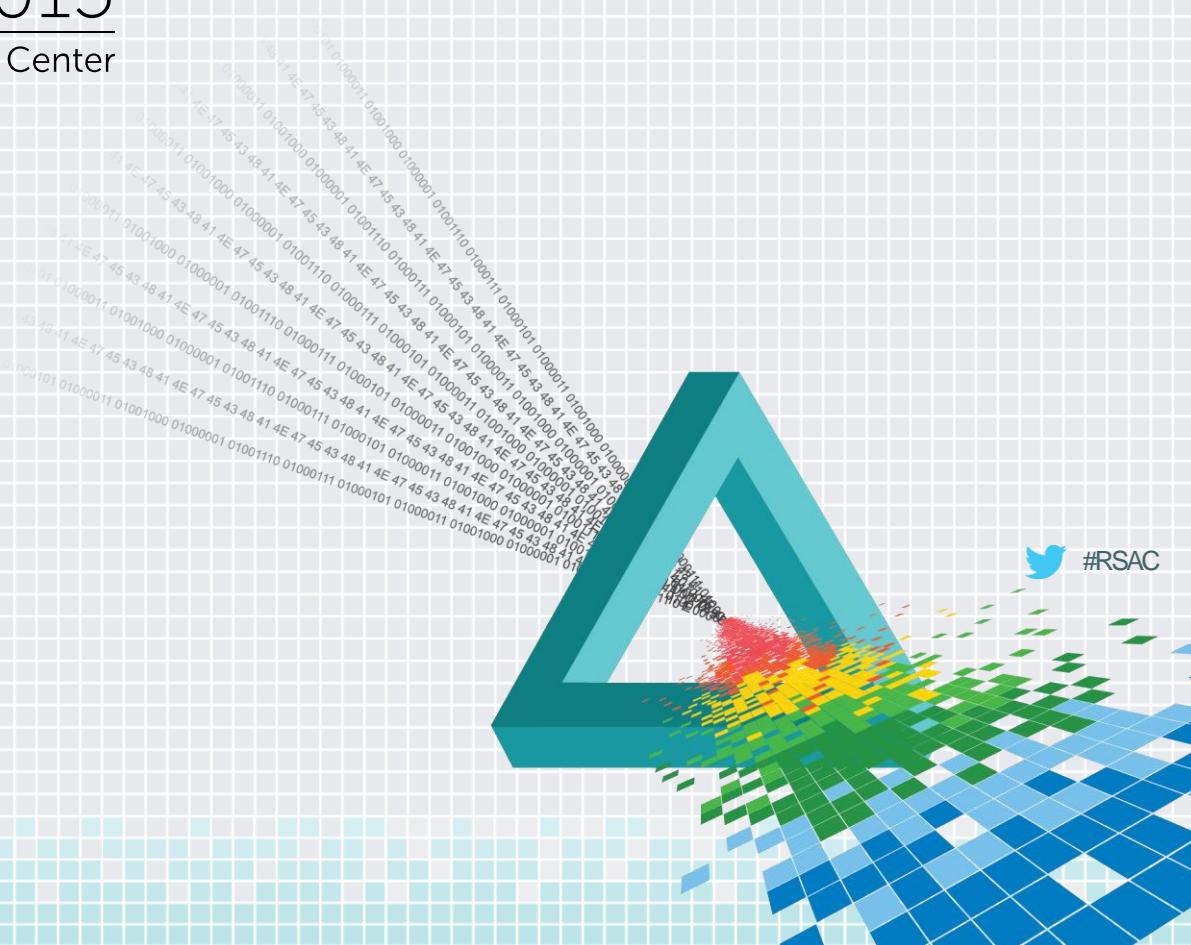
Information Security Program Mgr.
Seattle Children's Hospital
@DSeverski



RSA® Conference 2015

San Francisco | April 20-24 | Moscone Center

REFERENCES



References – Tools and Code

- ◆ VulnPryer
 - ◆ Python Code – <https://github.com/SCH-CISM/VulnPryer>
 - ◆ R Version – <https://github.com/SCH-CISM/vulnpryr>
 - ◆ AWS Automation Code – <https://github.com/SCH-CISM/sch-vulnpryer-orchestration>
- ◆ Vendors
 - ◆ RedSeal – Network Security Posture Analysis
 - ◆ Risk Based Security – Vulnerability Database
 - ◆ Risk I/O – Prioritization as a service
- ◆ Tools
 - ◆ Tableau
 - ◆ Chart.io

References – Additional Reading

- ◆ ¹ Immunity, Inc. White Paper: A Bounds Check on the Microsoft Exploitability Index
<http://download.microsoft.com/download/3/E/B/3EBDB81C-DF2F-470B-8A64-981DC8D9265C/A%20Bounds%20Check%20on%20the%20Microsoft%20Exploitability%20Index%20-%20final.pdf>
- ◆ ² Exploitability/Priority Index Rating Systems (Approaches, Value, and Limitations) <https://www.riskbasedsecurity.com/reports/RBS-ExploitabilityRatings-2013.pdf>

