Risk metrics for vulnerabilities exploited in the wild

Luca Allodi

University of Trento, Italy.

$name.$surname@unitn.it

Outline

- Introduction
  - Approaches to estimate system risk
  - The CVSS score
  - Result: guidelines
- Vulnerability landscapes
  - The good guys
  - Most bad guys
  - Our baseline: data
  - Reality on attacks, according to the data
- Observational analysis of CVSS scores
  - CVSS distributions
  - Map of vulnerabilities, exploits and CVSS scores: CVSS not good
- What makes the CVSS so inaccurate?
  - Inspection of CVSS subscore distributions
  - Case controlled study: CVSS as a test for exploitation
  - A bit of Bayes
  - Relative diminishment in risk with vulnerability patching
- Conclusions

Introduction

What is a vulnerability

- *A weakness of an asset or group of assets that can be exploited by one or more threats*
- *A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy*
- *A weakness in design, implementation, operation or internal control*
- …
- *Some even speak of "probability of being attacked"..*

What is a vulnerability

- All very general definitions
  - Software, Design, Architecture, …
- We are interested in software vulnerabilities
- Still, a sw vulnerability may mean many things:
  - A security bug is there, nobody knows about it
  - The vulnerability is disclosed
  - A proof-of-concept exploit exists
  - The bad guys are actually attacking it

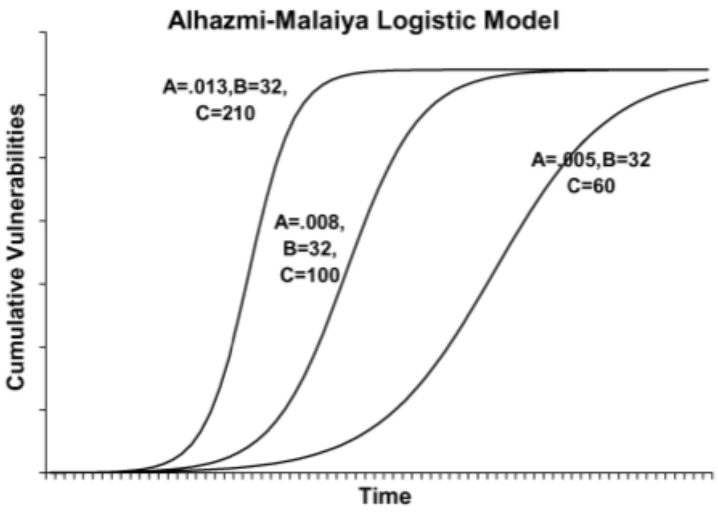
→ Different levels of risk

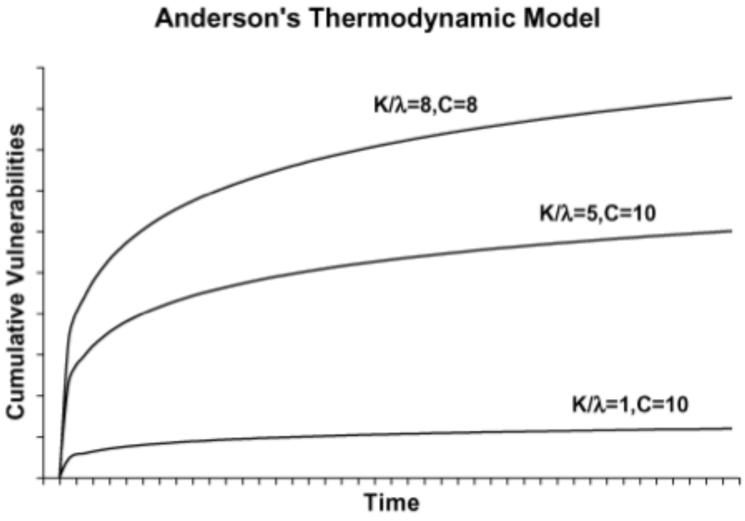
With that in mind..

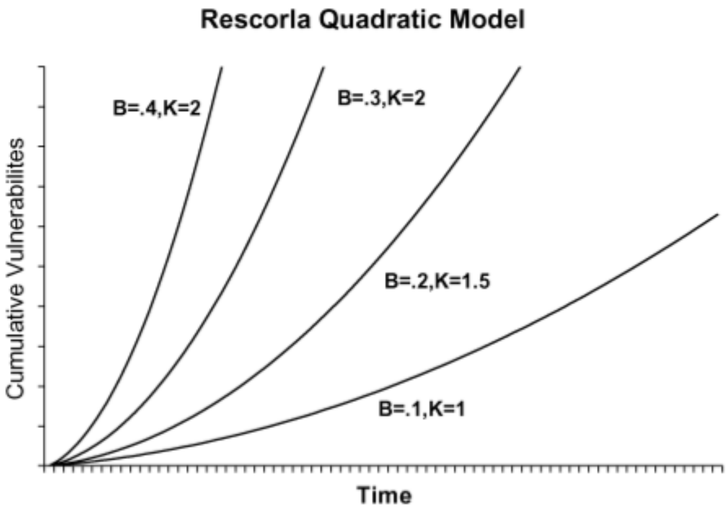
- Say that we decided what a vulnerability is
- How do we measure **how much trouble are we in?**
  - Vulnerability Discovery Models
  - Attack Surfaces
  - Attack Graphs

With that in mind.. VDMs

- Vulnerability Discovery Models
- Estimate at a certain time *t* how many vulnerabilities you may expect to have in your software at time *t+n*

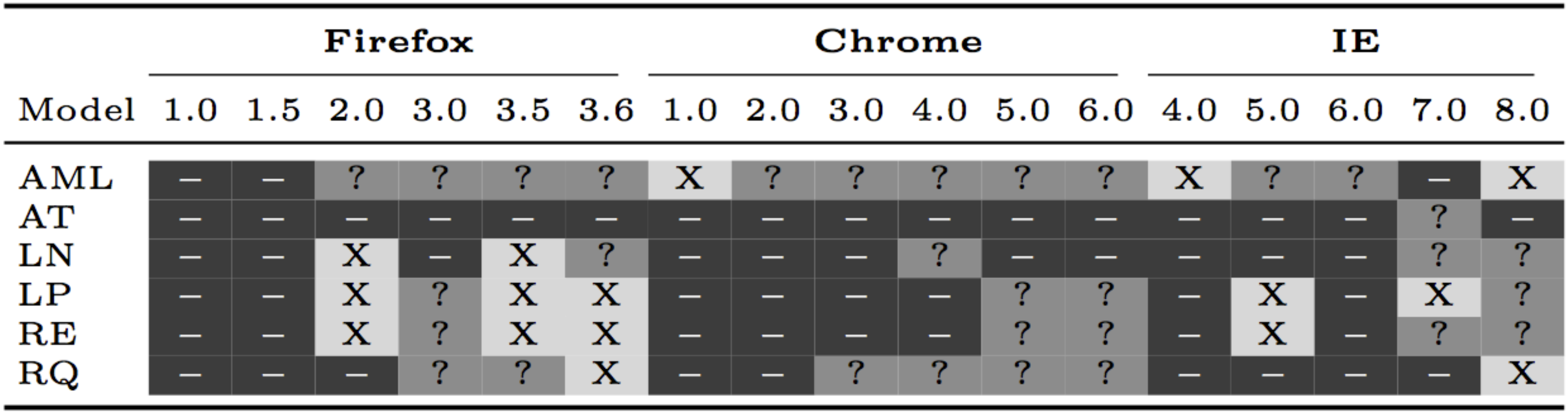






With that in mind.. VDMs

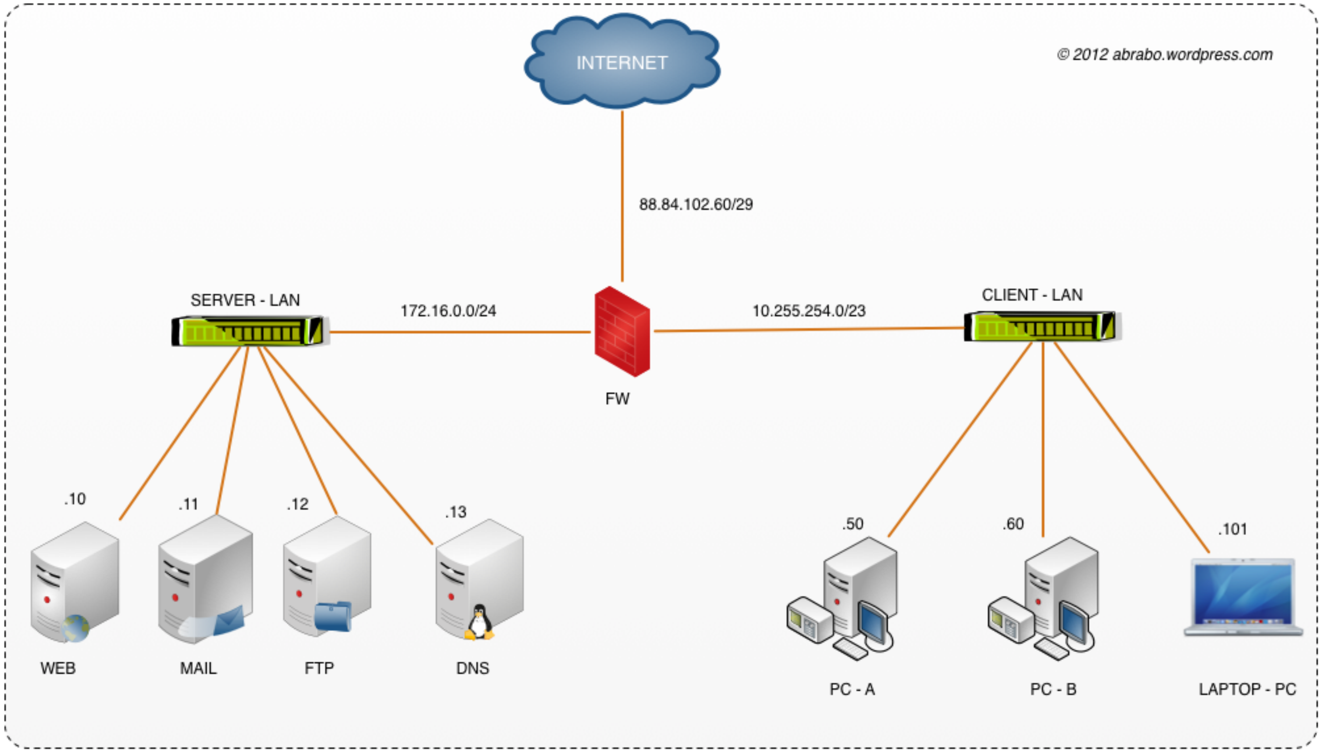
- Bottom line: Count no. of vulns
- Also, they do not really work (at least for browsers)
  - X = works (p>=0.95)
  - ? = Cannot assess if it works (0.05<p<0.95)
  - - = Does not work (p=<0.05)



	Firefox						Chrome						IE				
Model	1.0	1.5	2.0	3.0	3.5	3.6	1.0	2.0	3.0	4.0	5.0	6.0	4.0	5.0	6.0	7.0	8.0
AML	–	–	?	?	?	?	X	?	?	?	?	?	X	?	?	–	X
AT	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	?	–
LN	–	–	X	–	X	?	–	–	–	?	–	–	–	–	–	?	?
LP	–	–	X	?	X	X	–	–	–	–	?	?	–	X	–	X	?
RE	–	–	X	?	X	X	–	–	–	–	?	?	–	X	–	?	?
RQ	–	–	–	?	?	X	–	–	?	?	?	?	–	–	–	–	X

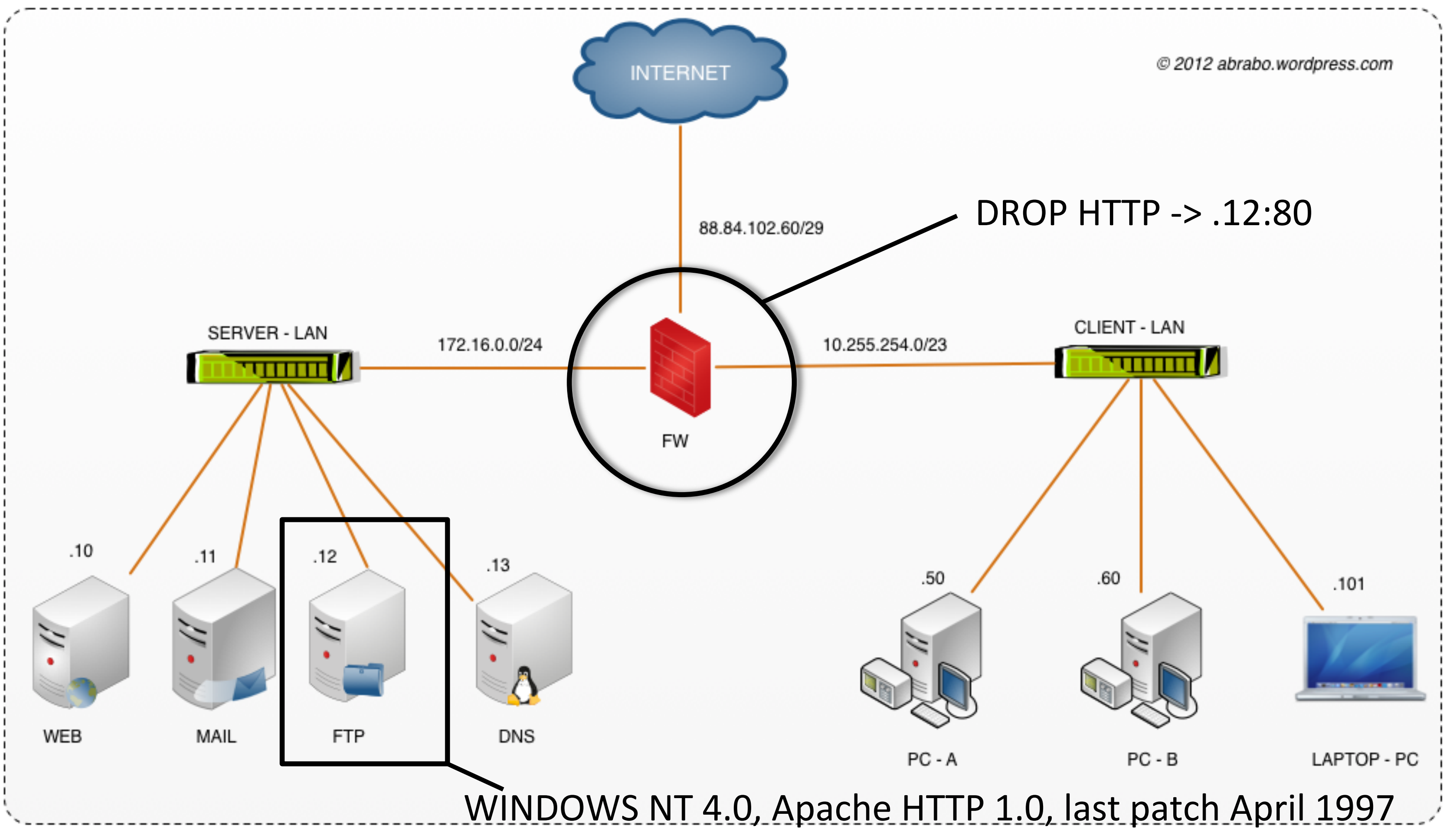
With that in mind.. Attack Surfaces

- Change the definition of vulnerability
- Vulnerability is not the *technicality* by itself
  - It needs to be exposed to represent risk
- For example:



© 2012 abrabo.wordpress.com

With that in mind.. Attack Surfaces



DROP HTTP -> .12:80

WINDOWS NT 4.0, Apache HTTP 1.0, last patch April 1997

Who Cares?

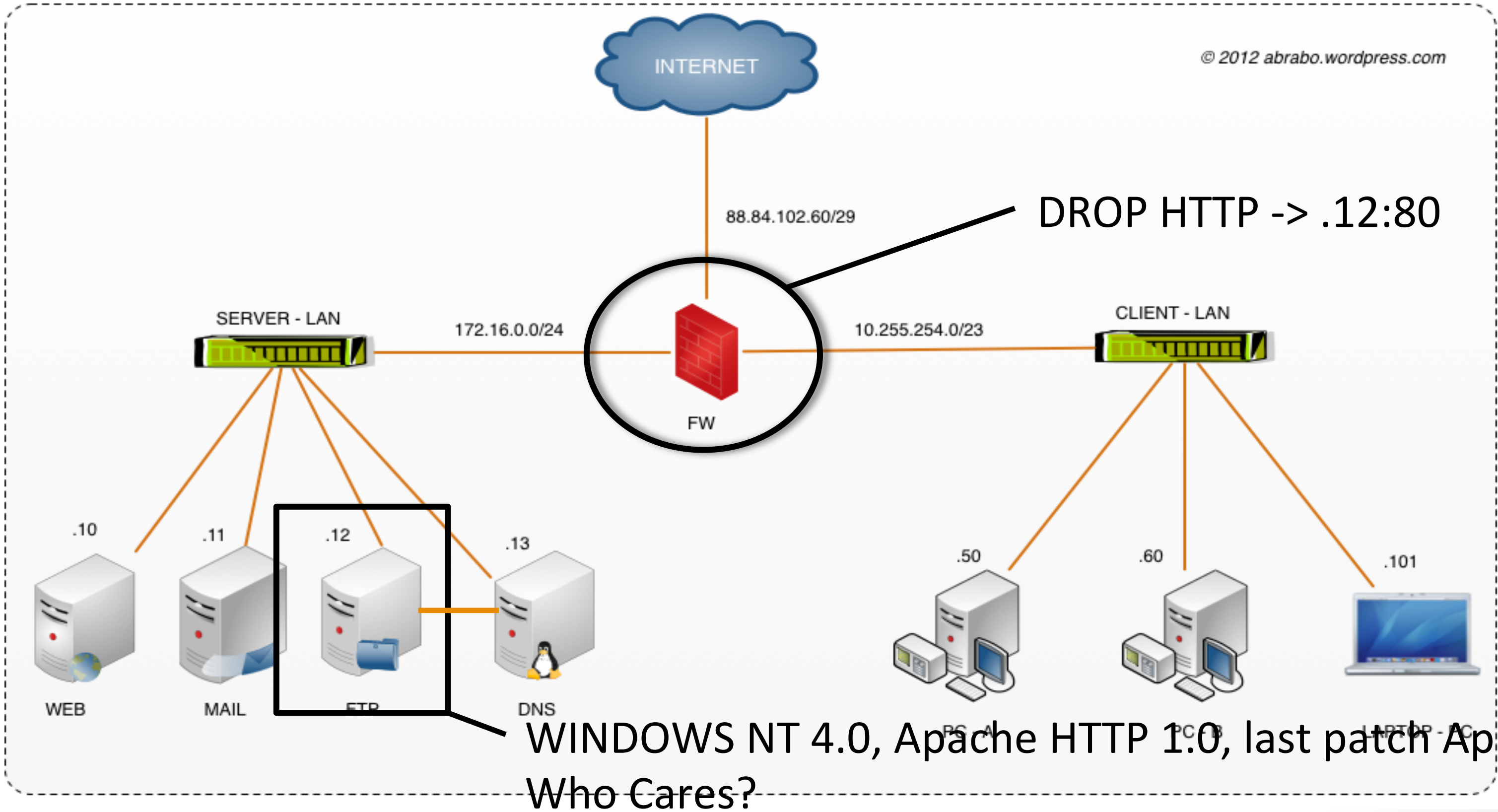
With that in mind.. Attack Surfaces

- They change the definition of vulnerability
- Identify a subset of “vulnerabilities” that are a threat to you

- Bottom line: Count no. of vulns

With that in mind.. Attack Graph

- Assume that some vulnerabilities can be exploited only *after* others (e.g. unreachable)



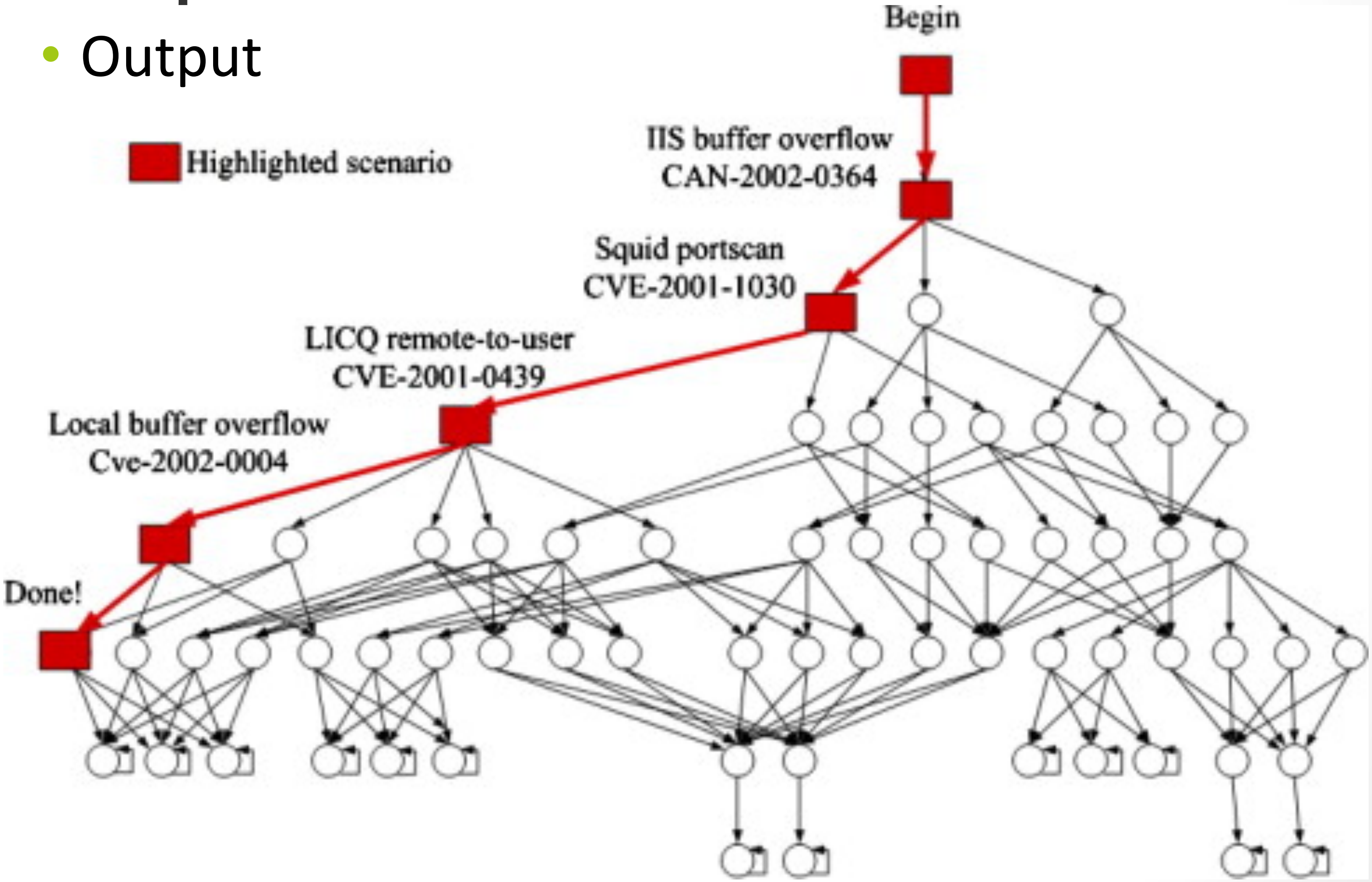
© 2012 abrabo.wordpress.com

DROP HTTP -> .12:80

WINDOWS NT 4.0, Apache HTTP 1.0, last patch April 1997
Who Cares?

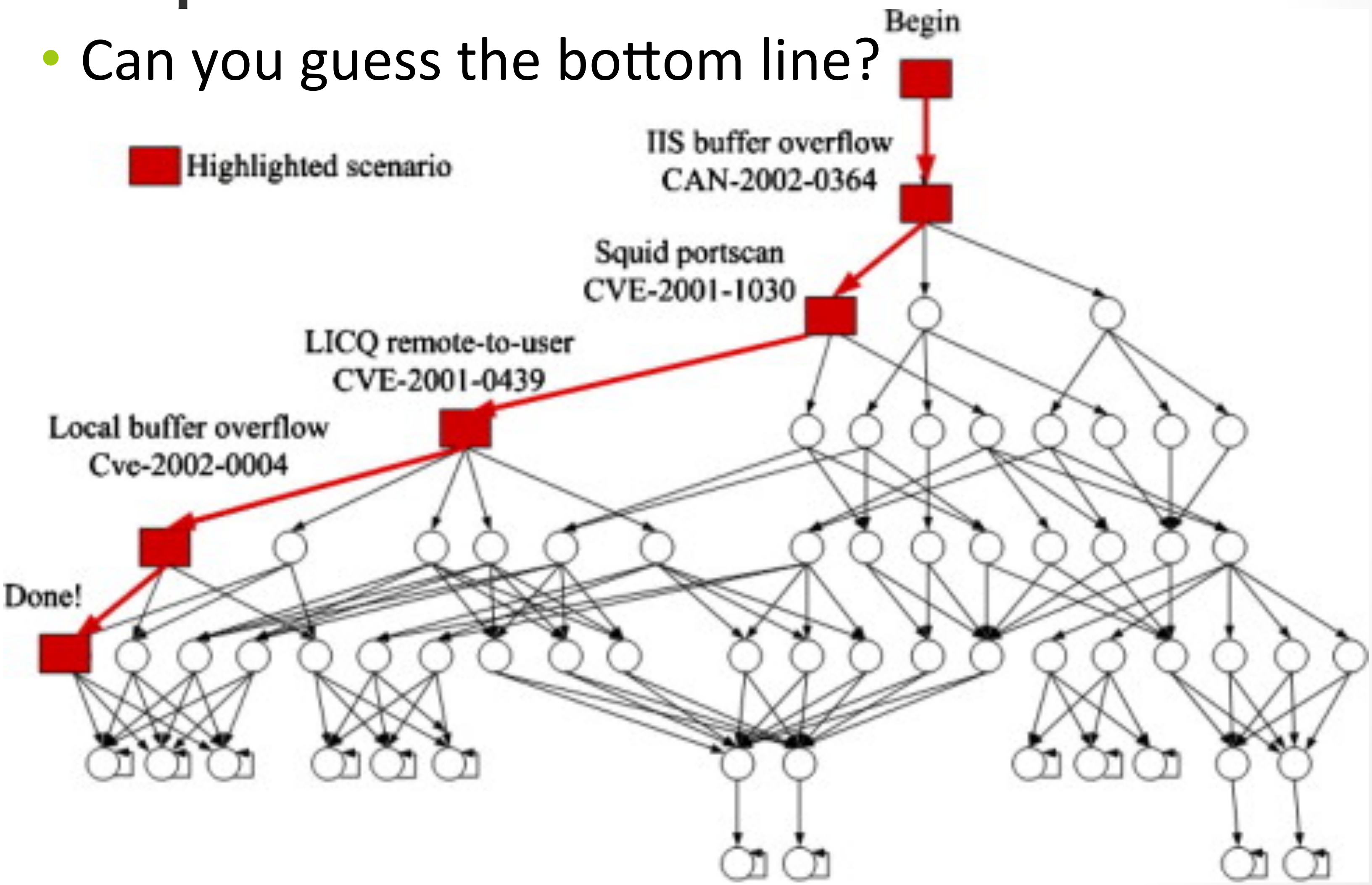
With that in mind.. Attack Graphs

- Output



With that in mind.. Attack Graphs

- Can you guess the bottom line?



This is typical in IT security

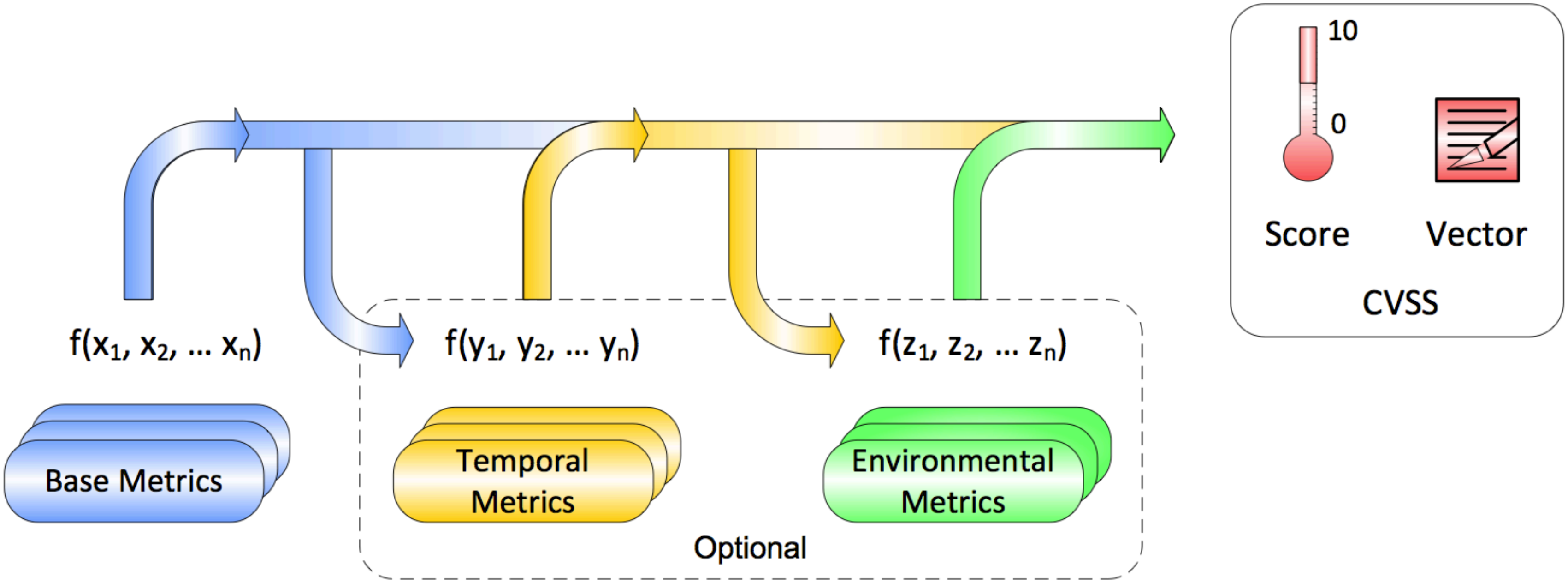
- Schneier:
  - Security is as strong as the weakest link
- Dolev's Model of the attacker (Crypto)
  - Very powerful, can do anything, can see anything
- Variations to these models exist
  - E.g. honest but curious
- Still, they all say the same:

If a vulnerability is there, sooner or later somebody will attack it

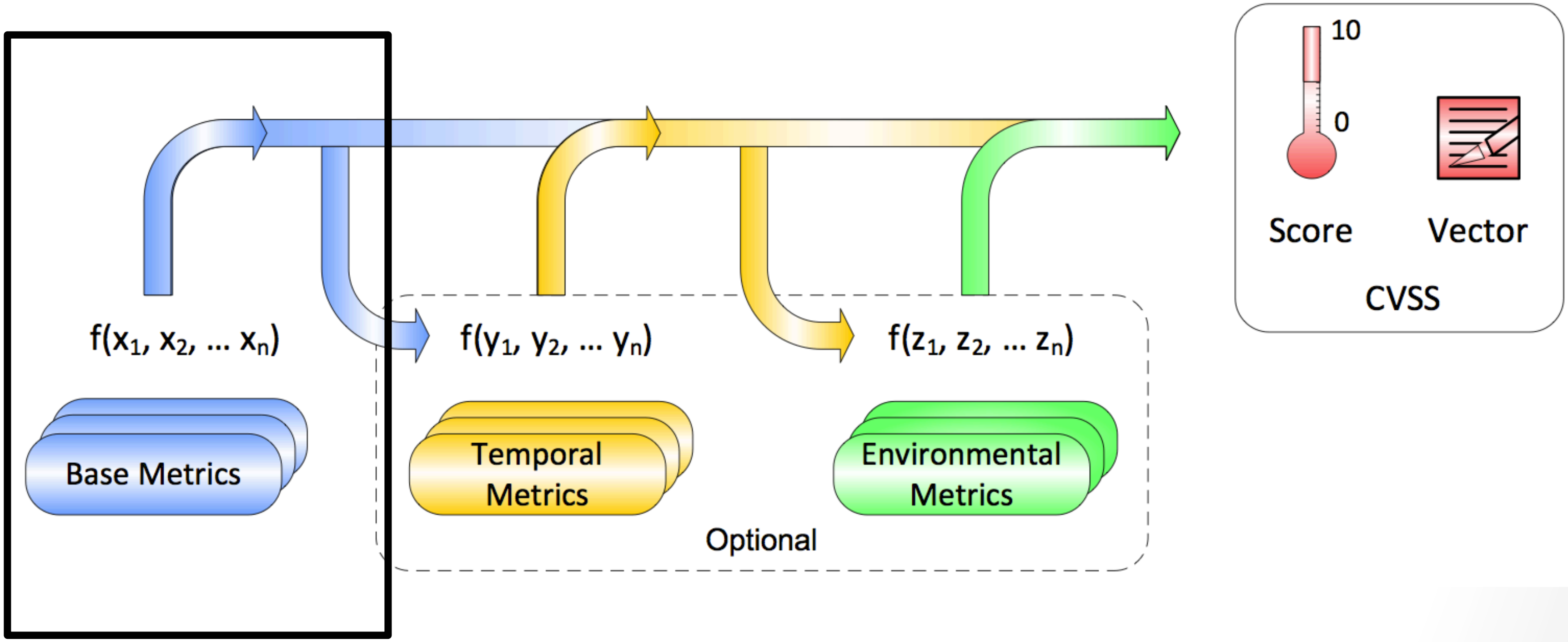
We are almost there

- Vulnerabilities are not all the same
- We need a metric to characterize them
- NIST CVSS Score
    - Identifies a number of technical characteristics of the vulnerability
    - Assign a "criticality score" to each characteristic
    - The function returns a "risk score" for the vulnerability
        - Classic risk function: Risk = Impact x Likelihood

CVSS Score


10
0
Score
Vector
CVSS
f(x1, x2, … xn)
f(y1, y2, … yn)
f(z1, z2, … zn)
Base Metrics
Temporal Metrics
Environmental Metrics
Optional

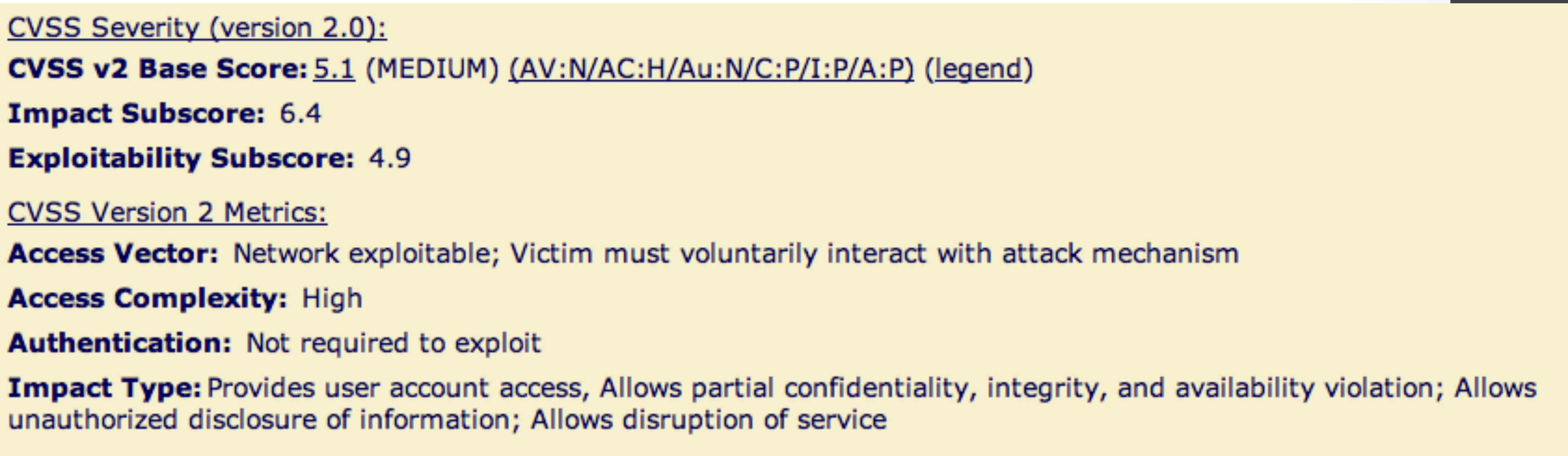
CVSS Score


f(x_1, x_2, … x_n)
Base Metrics
f(y_1, y_2, … y_n)
Temporal Metrics
f(z_1, z_2, … z_n)
Environmental Metrics
Optional
10
0
Score
Vector
CVSS

CVSS Score: Base Metric

- Impact x Likelihood
- Each variable computed on the basis of three expert assessments
- Impact:
  - Confidentiality (Complete, Partial, None)
  - Integrity
  - Availability
- Exploitability:
  - Access Vector (Network, Adjacent Net., Local)
  - Access Complexity (high, med, low)
  - Authentication (..)

CVSS Score: Base Metric



CVSS Severity (version 2.0):

CVSS v2 Base Score: 5.1 (MEDIUM) (AV:N/AC:H/Au:N/C:P/I:P/A:P) (legend)

Impact Subscore: 6.4

Exploitability Subscore: 4.9

CVSS Version 2 Metrics:

Access Vector: Network exploitable; Victim must voluntarily interact with attack mechanism

Access Complexity: High

Authentication: Not required to exploit

Impact Type: Provides user account access, Allows partial confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service

Vulnerabilities guidelines

- US Government SCAP Protocol for vulnerability remediation [Scarfone 2010]

> *“Organizations should use CVSS base scores to assist in prioritizing the remediation of known security-related software flaws based on the relative severity of the flaws.”*

Vulnerabilities guidelines

- US Government SCAP Protocol for vulnerability remediation [Scarfone 2010]

> *"Organizations should use CVSS base scores to assist in prioritizing the remediation of known security-related software flaws based on the relative severity of the flaws."*

➡ bother with every software vulnerability, use CVSS to prioritize your work

Don’t cite me on that (they said)

- *“My job is **the** professional nightmare: if everything goes well, I am not doing anything. If something goes badly wrong, I get fired.”* – Security Manager of big Italian player in sw industry
- *“Just acknowledging there is a bug costs hundreds of euros”* – Representative of EU leader in sw management
- *“You are crazy if you think I’ll install all the patches”* – IT Admin of big US telecommunication company

Vulnerabilities: research question

- What the CIO would like to know
  - If I follow SCAP or equivalent guidelines, how much will my final risk decrease?

- A clear value proposition:
  - if we fix high CVSS vulns we decrease risk by +43%
  - if we fix all medium CVSS only raises to +48%
    - → +5% more is not worth the extra money, maybe even +43% is not worth

Vulnerabilities: landscapes

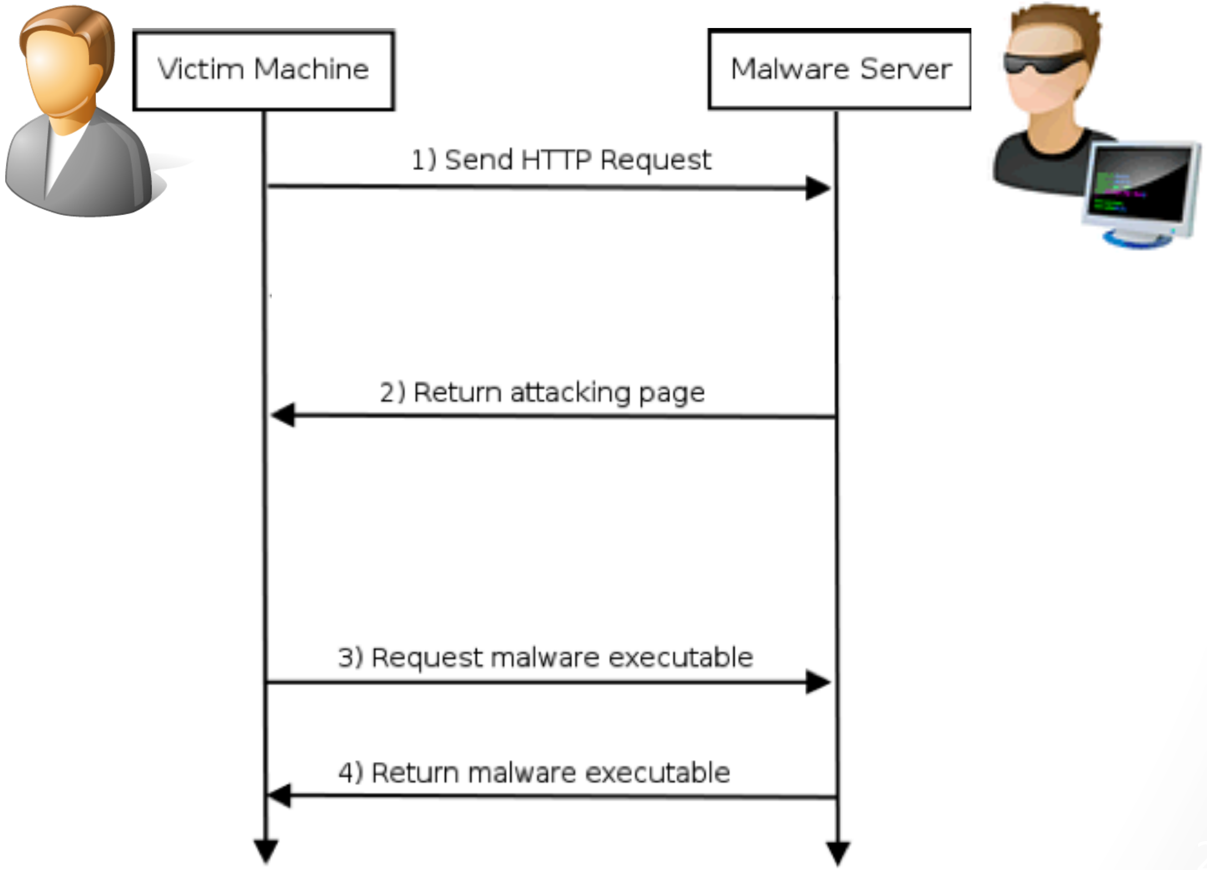
Vulnerabilities: the good guys

- Databases for vulnerabilities:
  - Lots of Vulnerabilities are published daily
  - NVD runs at 50K
  - CVSS scoring system is now drafting V.3
- Databases for exploits:
  - Vendors' "Bounty programs"
  - iDefender, TippingPoint acquisition program
  - "Responsible Disclosure" debate
- Analysis of complete protection against a powerful adversary
  - Classic model of the attacker [Dolev, Schneier…]

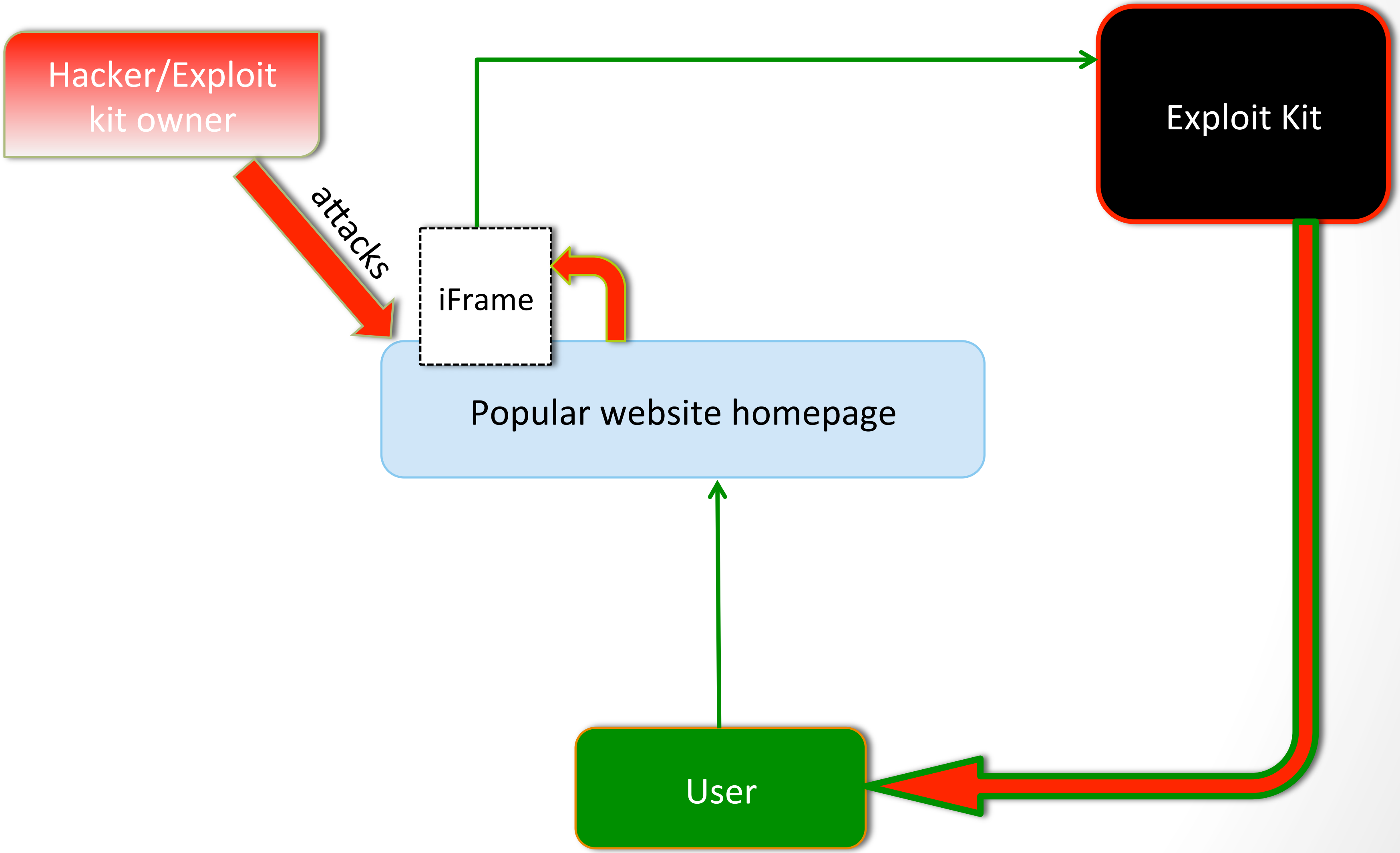
➢ Fix all vulnerabilities or die

Vulnerabilities: most bad guys

- Automated web attacks represent 2/3 of final threat for users [Google 2011],[Grier 2012]

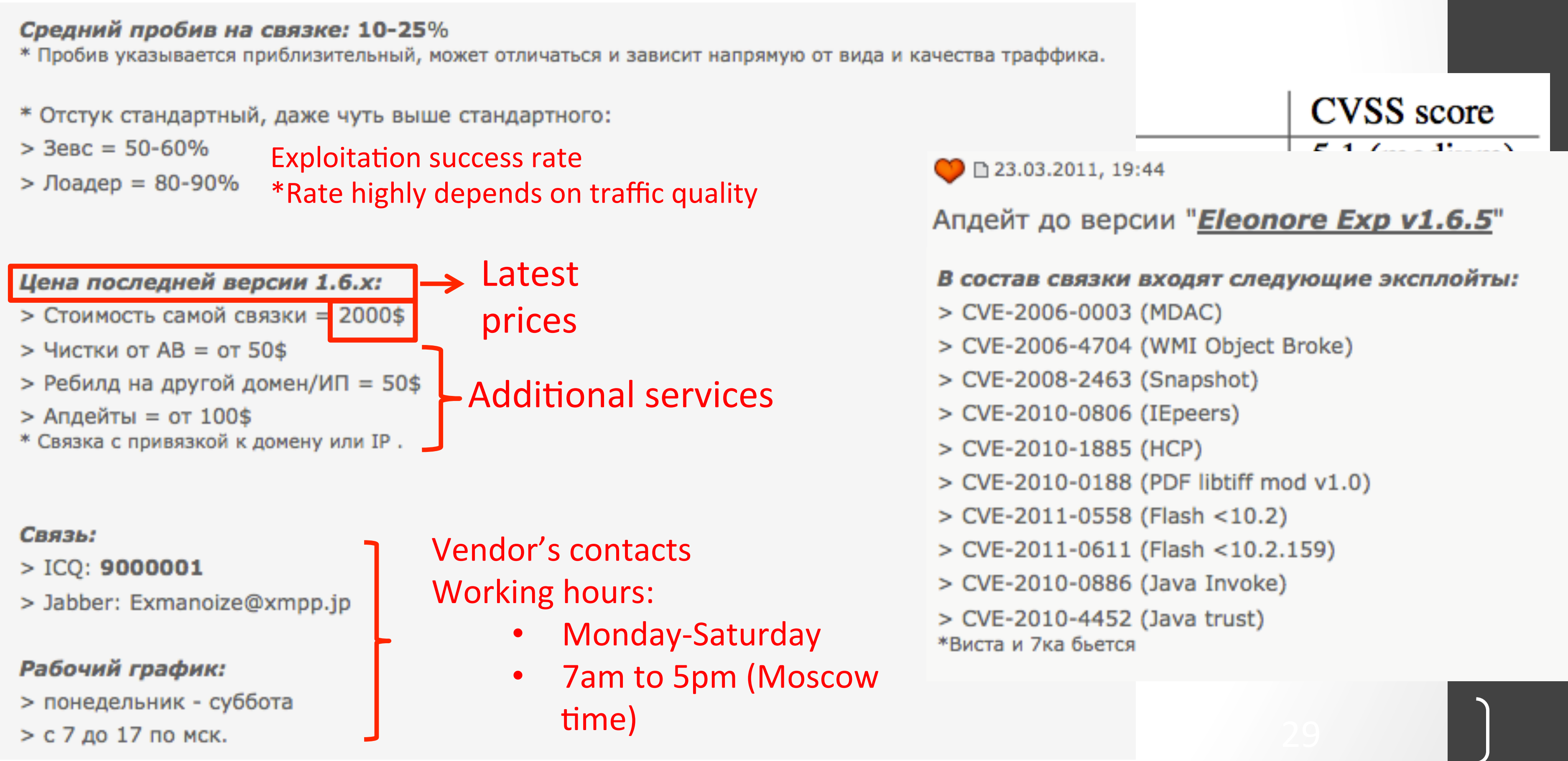


Vulnerabilities: most bad guys



Vulnerabilities: most bad guys

- Automated web attacks represent 2/3 of final threat for users [Google 2011],[Grier 2012]



Средний пробив на связке: 10-25%
* Пробив указывается приблизительный, может отличаться и зависит напрямую от вида и качества траффика.

* Отстук стандартный, даже чуть выше стандартного:
> Зевс = 50-60%
> Лоадер = 80-90%

Exploitation success rate
*Rate highly depends on traffic quality

Цена последней версии 1.6.x:
> Стоимость самой связки = 2000$
> Чистки от АВ = от 50$
> Ребилд на другой домен/ИП = 50$
> Апдейты = от 100$
* Связка с привязкой к домену или IP .

Latest prices

Additional services

Связь:
> ICQ: **9000001**
> Jabber: Exmanoize@xmpp.jp

Рабочий график:
> понедельник - суббота
> с 7 до 17 по мск.

Vendor's contacts
Working hours:

- Monday-Saturday
- 7am to 5pm (Moscow time)

CVSS score

23.03.2011, 19:44

Апдейт до версии "***Eleonore Exp v1.6.5***"

В состав связки входят следующие эксплойты:
> CVE-2006-0003 (MDAC)
> CVE-2006-4704 (WMI Object Broke)
> CVE-2008-2463 (Snapshot)
> CVE-2010-0806 (IEpeers)
> CVE-2010-1885 (HCP)
> CVE-2010-0188 (PDF libtiff mod v1.0)
> CVE-2011-0558 (Flash <10.2)
> CVE-2011-0611 (Flash <10.2.159)
> CVE-2010-0886 (Java Invoke)
> CVE-2010-4452 (Java trust)
*Виста и 7ка бьется

Vulnerabilities: our baseline

- **NVD**
  - The universe of vulnerabilities
- **EXPLOIT-DB**
  - Exploits published by security researchers
- **EKITS** (The black markets)
  - 1.5 years of study of the black markets
  - Automated monitoring of exploit kits and new CVEs
  - 90+ exploit kits from the black markets
- **SYM**
  - Vulnerabilities actually exploited in the wild
  - Browser/Plugins 14% – Server 22% – App. 24%
  - Solaris, MacOs, Linux and others are included

dataset	volume
NVD	49.624
EDB	8.189
EKITS	126
SYM	1.289

Reality so far

- The “Classic” Attacker Model looks wrong
  - Few exploited vulnerabilities
  - Big chunk of risk from a bunch of vulnerabilities
  - ~~Fix all vulnerabilities or die~~ → waste of money?
- But CIO can’t wait:
  - Use a Security Configuration Management Product!
  - 30+ products: Microsoft, Dell, HP, VMWare, McAfee, Symantec etc..
  - Based on CVSS (Common Vuln. Scoring System)

Observational analysis of CVSS scores

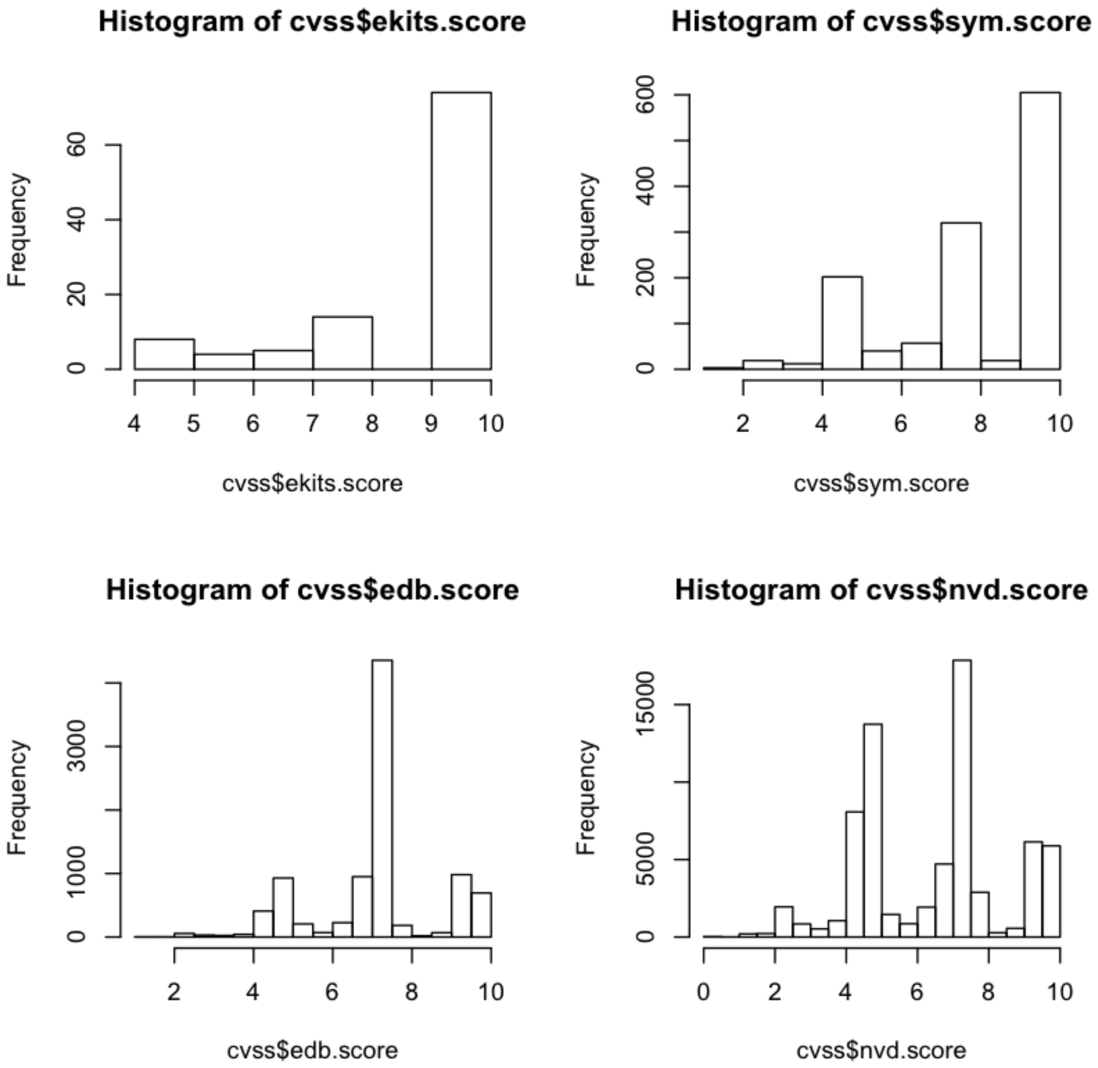
CVSS Study

- Remember: the SCAP protocol tells you: take a dataset of vulnerabilities, order vulnerabilities by CVSS.

- We therefore look at:

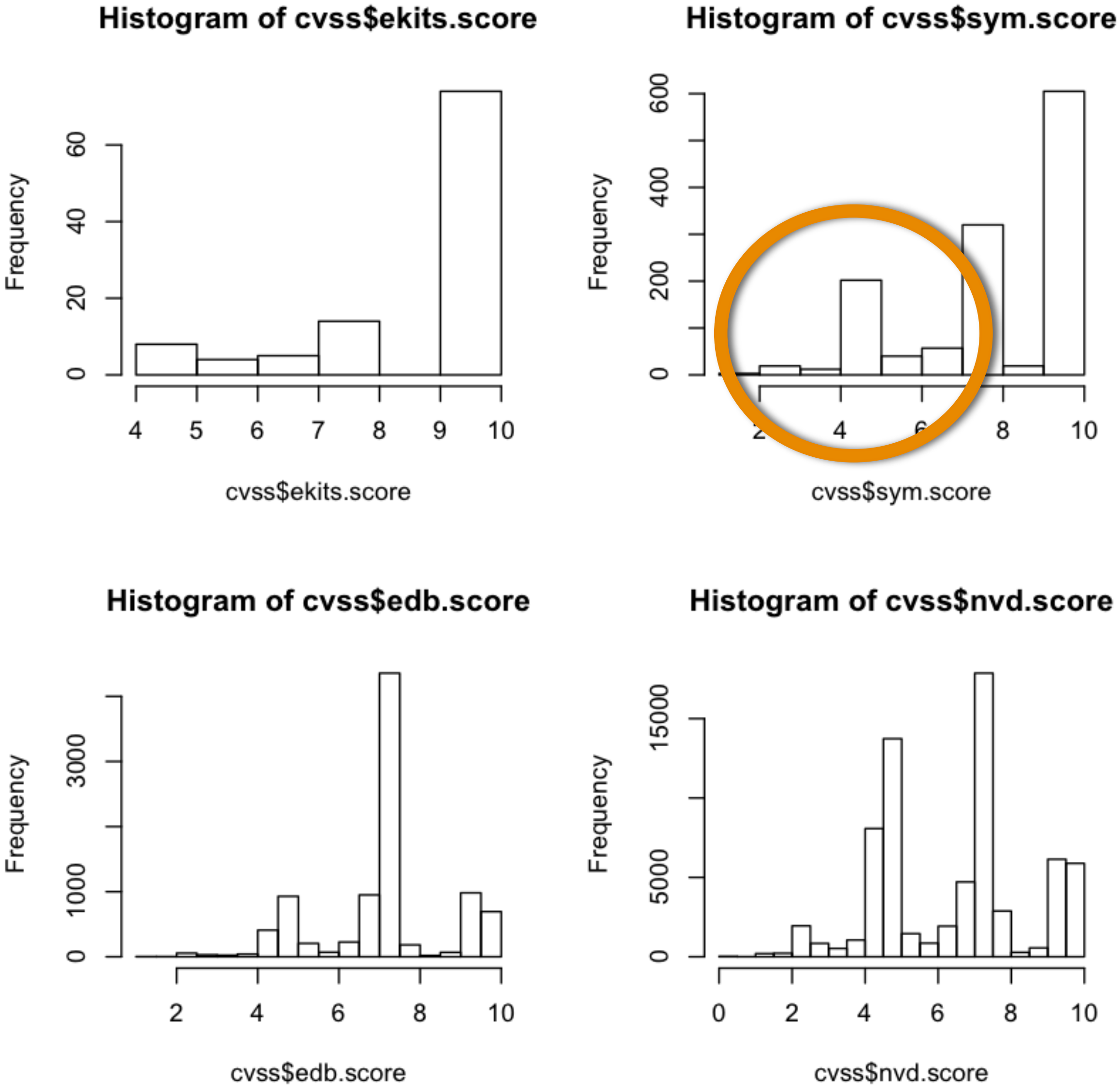
1. Distribution of CVSS scores per dataset
   - Are datasets different in terms of type of vulnerabilities?
2. VENN diagram of datasets and scores
   - Are datasets interesting in terms of attacks actually delivered by the bad guys?

CVSS Distribution: HIST



- LOW: CVSS <6
- MEDIUM: 6<CVSS<9
- HIGH: CVSS > 9

CVSS Distribution: HIST



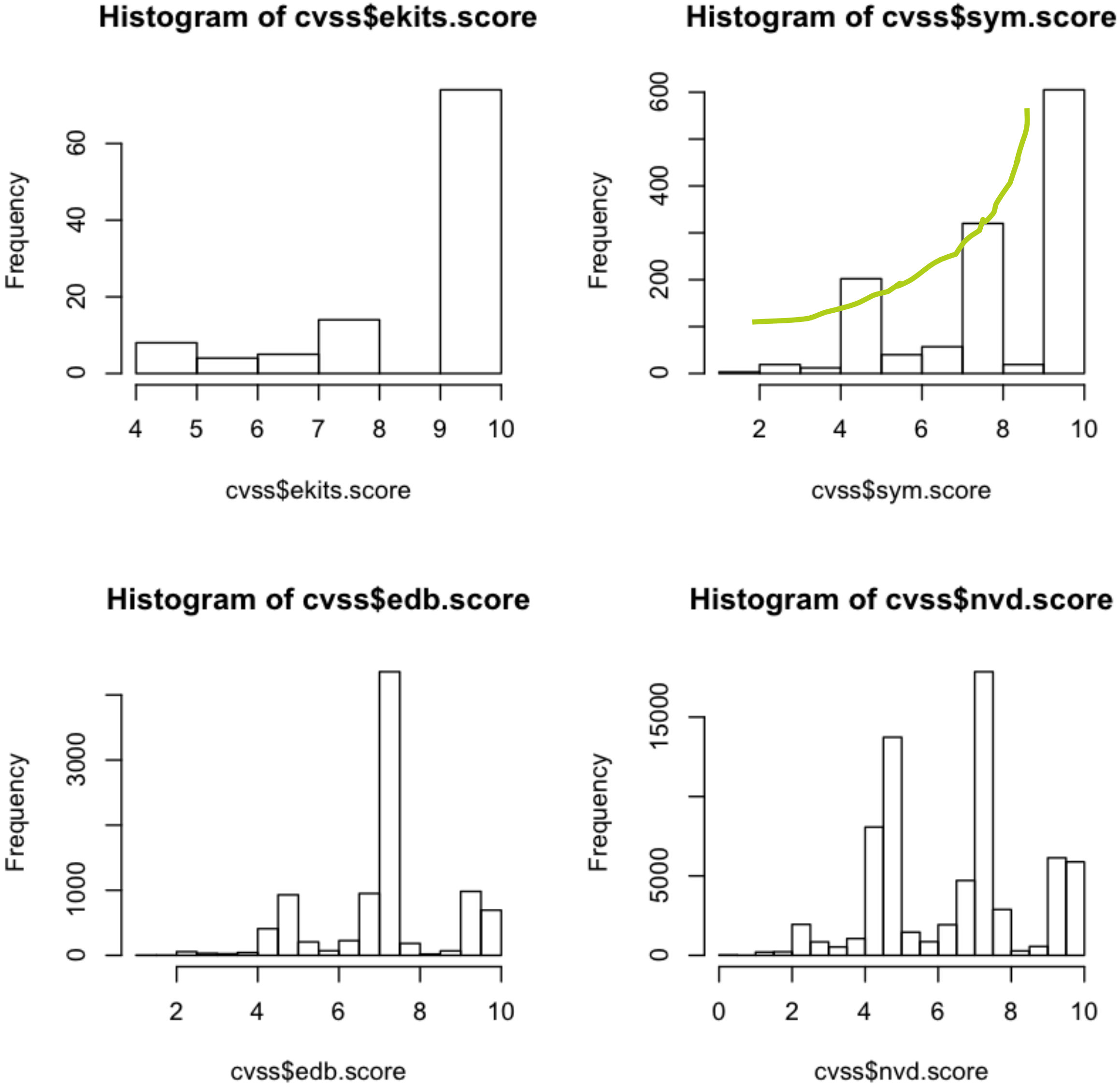
Histogram of cvss$ekits.score

Histogram of cvss$sym.score

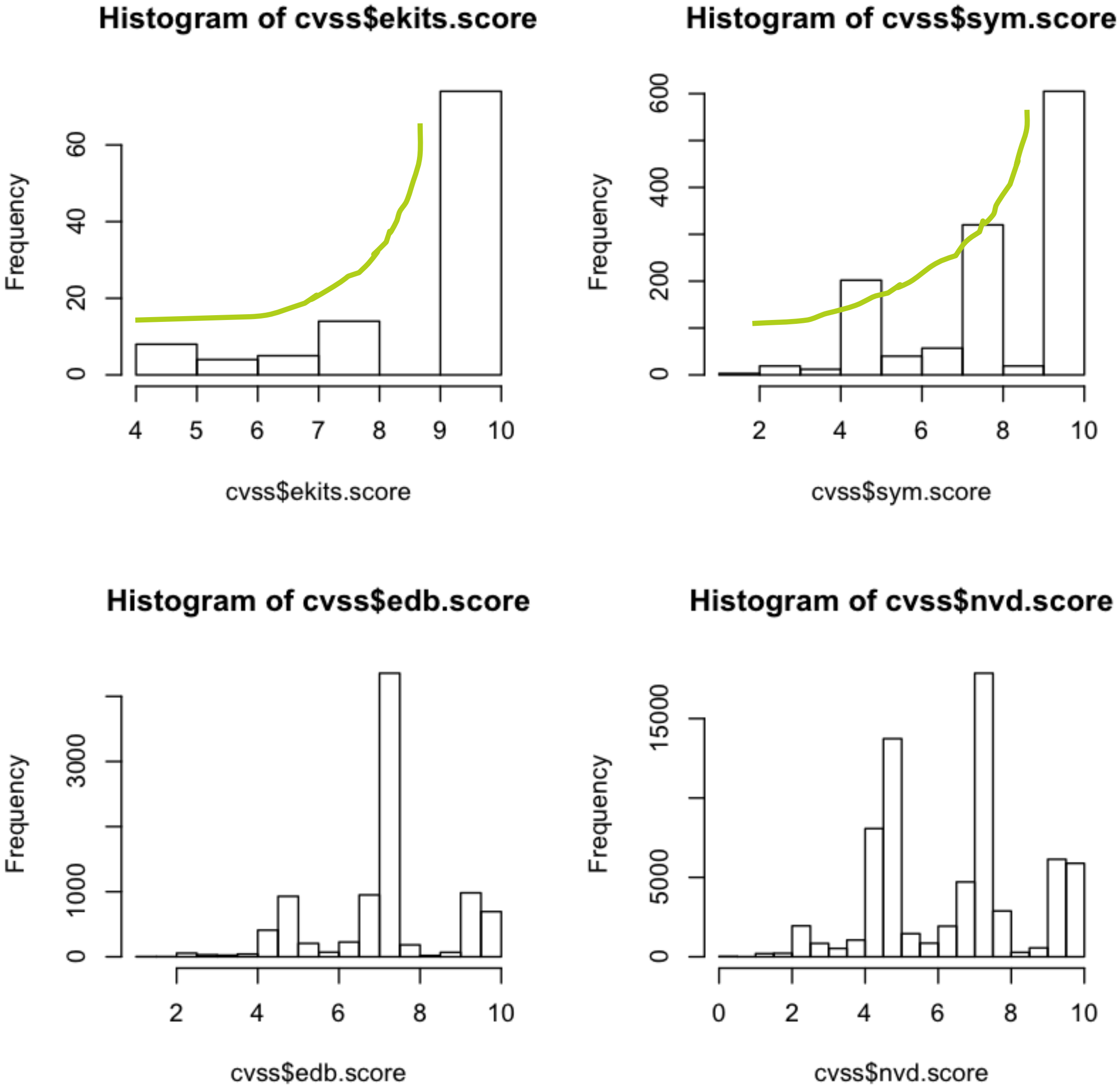
Histogram of cvss$edb.score

Histogram of cvss$nvd.score

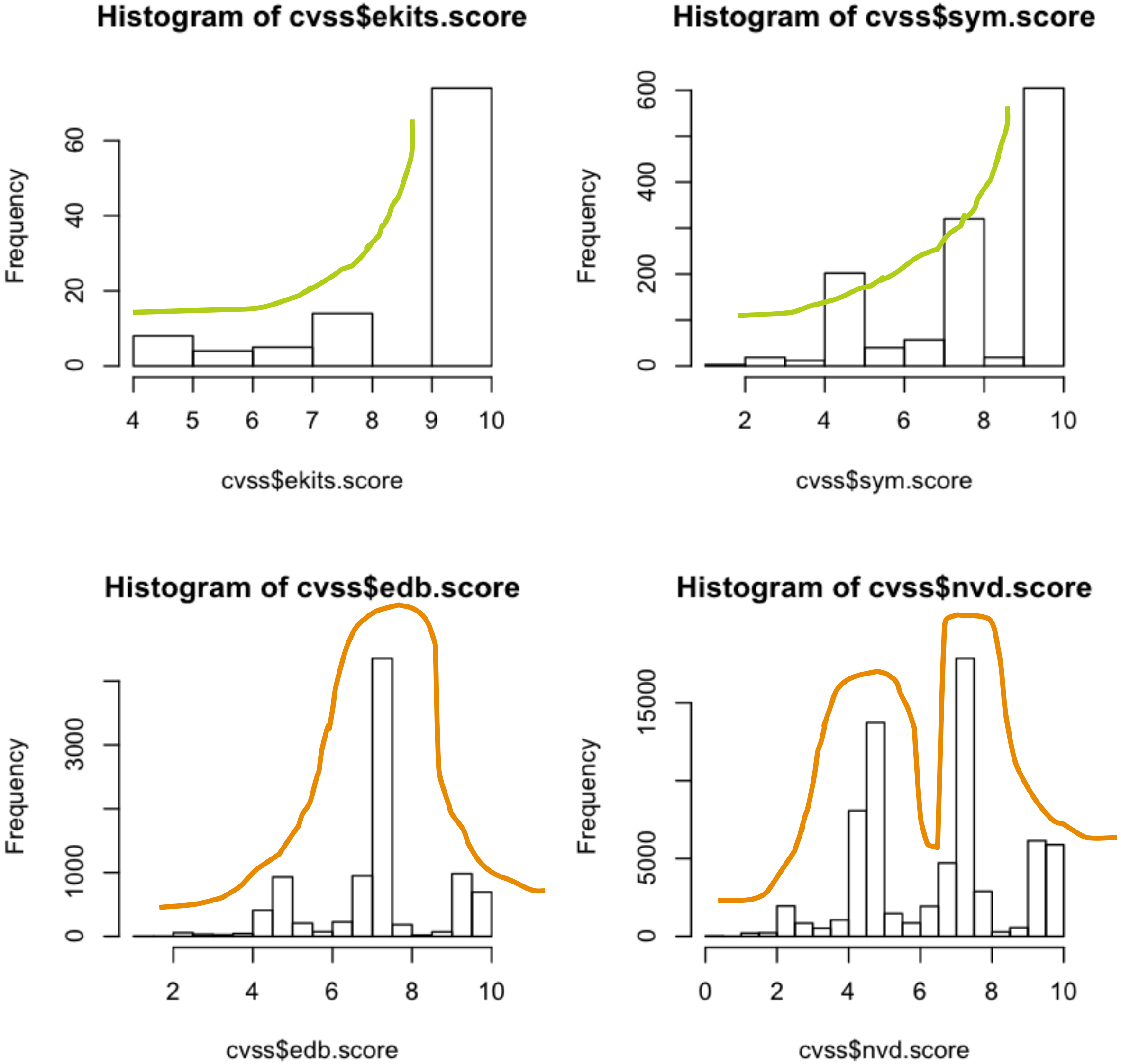
CVSS Distribution: HIST



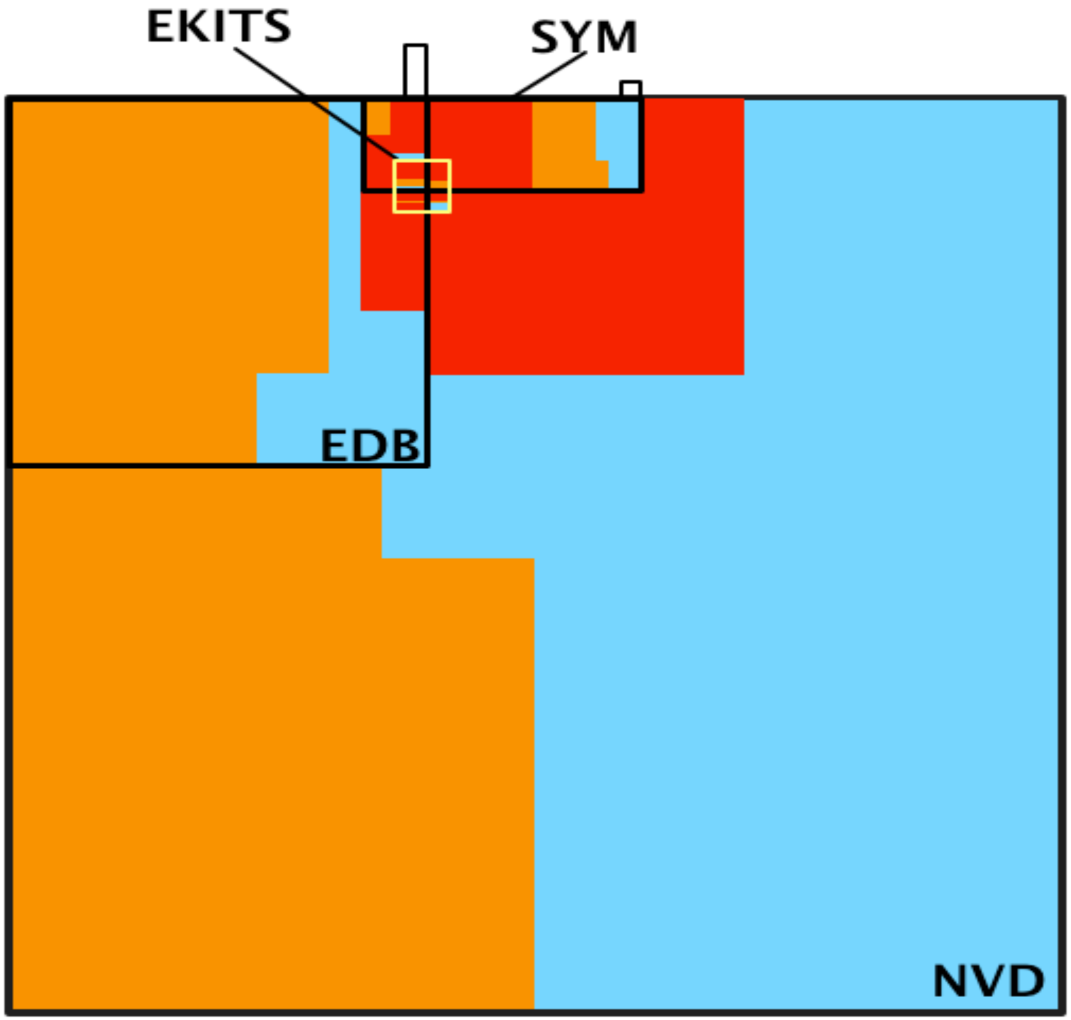
CVSS Distribution: HIST



CVSS Distribution: HIST



CVSS Distribution: VENN



🟦	LOW CVSS
🟧	MEDIUM CVSS
🟥	HIGH CVSS

Observational conclusions

- Attackers choose vulnerabilities autonomously:
  - They do not care about every vulnerability (NVD)
  - They do not care about every exploit (EDB)

- HIGH, MED+LOW score vulnerabilities are uniformly distributed in SYM dataset
- If you take NVD and fix all HIGH score vulnerabilities first [SCAP] you will:
  - Waste a lot of money patching all HIGH score vulnerabilities
  - Have addressed only 50% of final possible threats

What makes the CVSS so inaccurate?

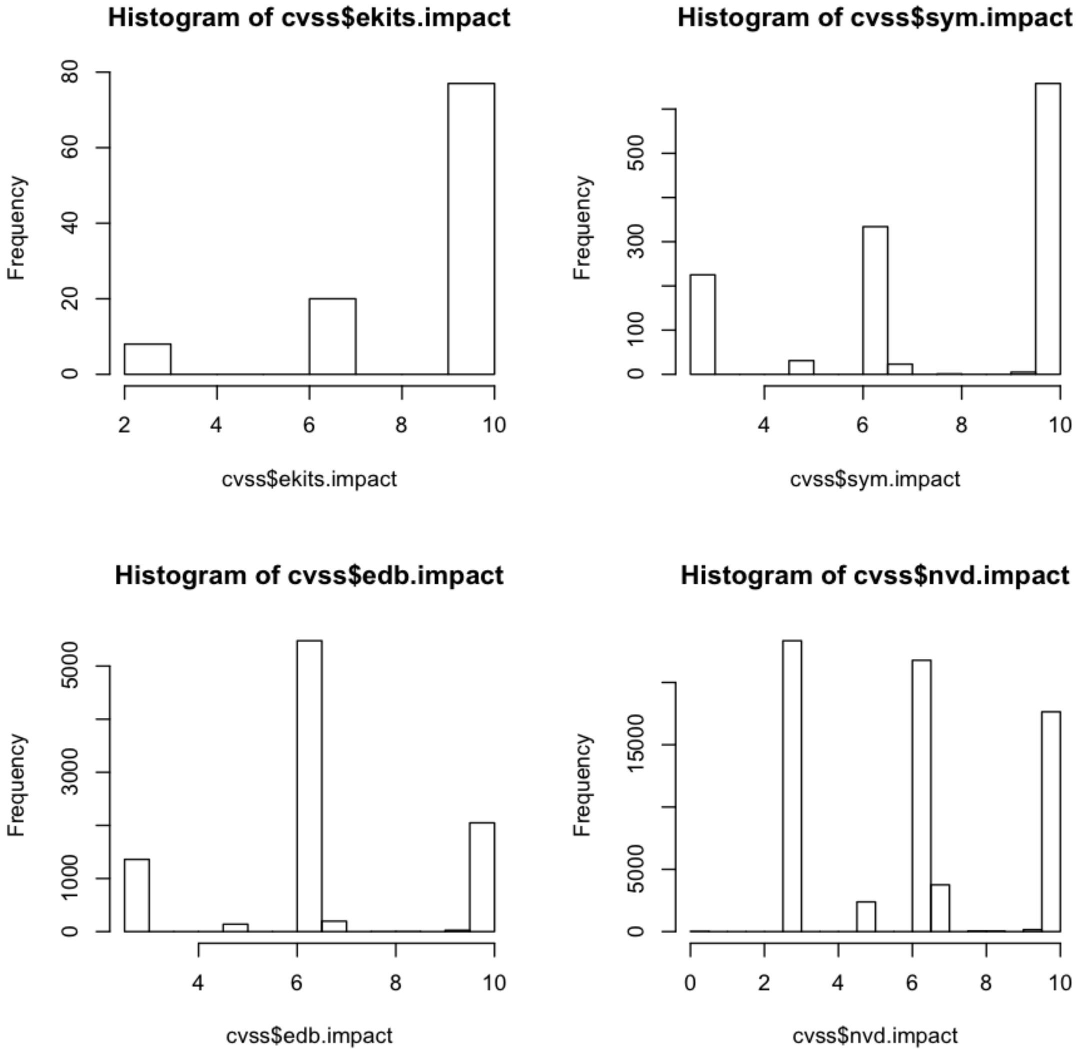
CVSS Metrics

- CVSS measures risk in the form

Risk = Impact x Likelihood

CVSS score = Impact x Exploitability

CVSS Metrics: Impact



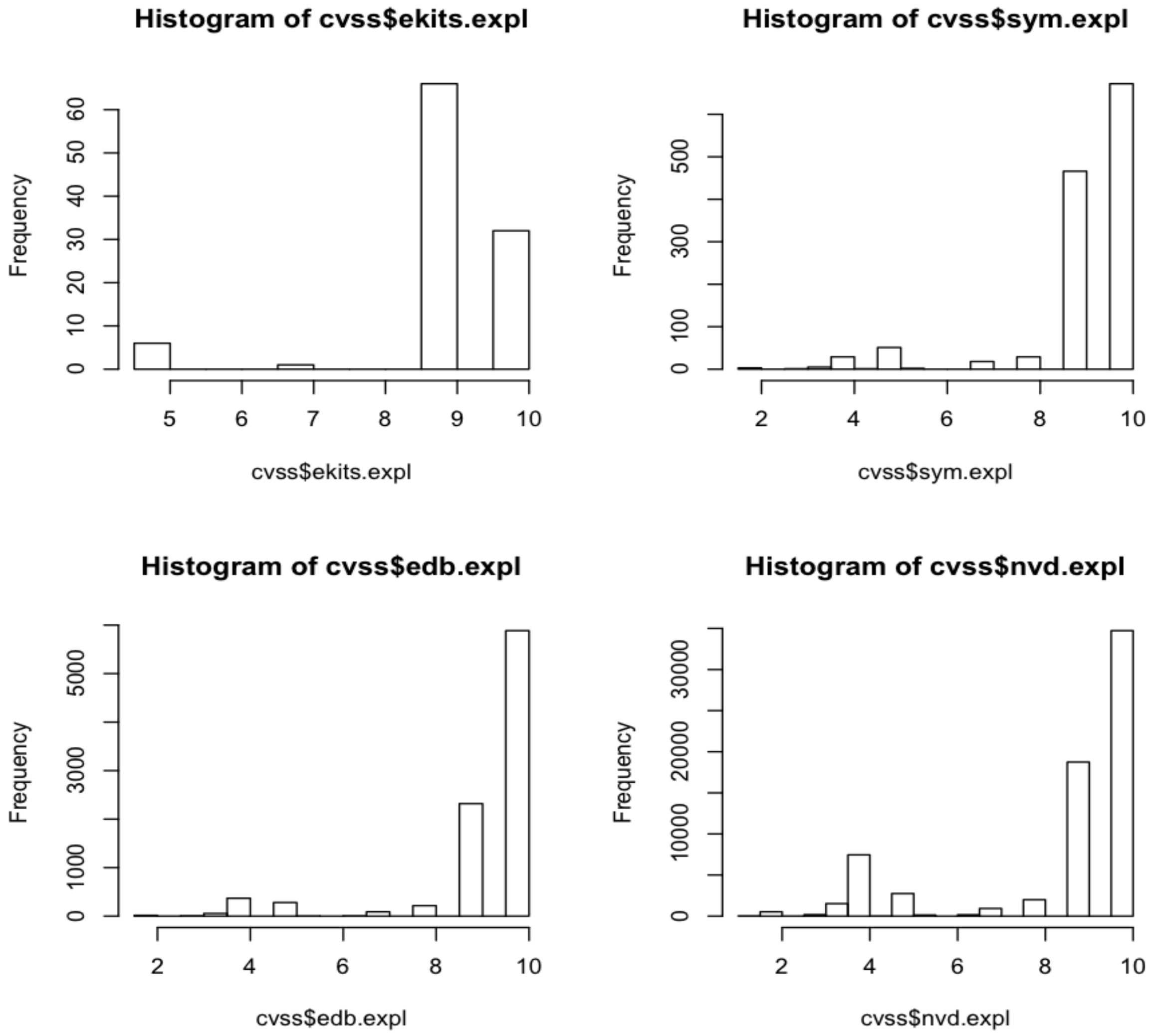
Histogram of cvss$ekits.impact

Histogram of cvss$sym.impact

Histogram of cvss$edb.impact

Histogram of cvss$nvd.impact

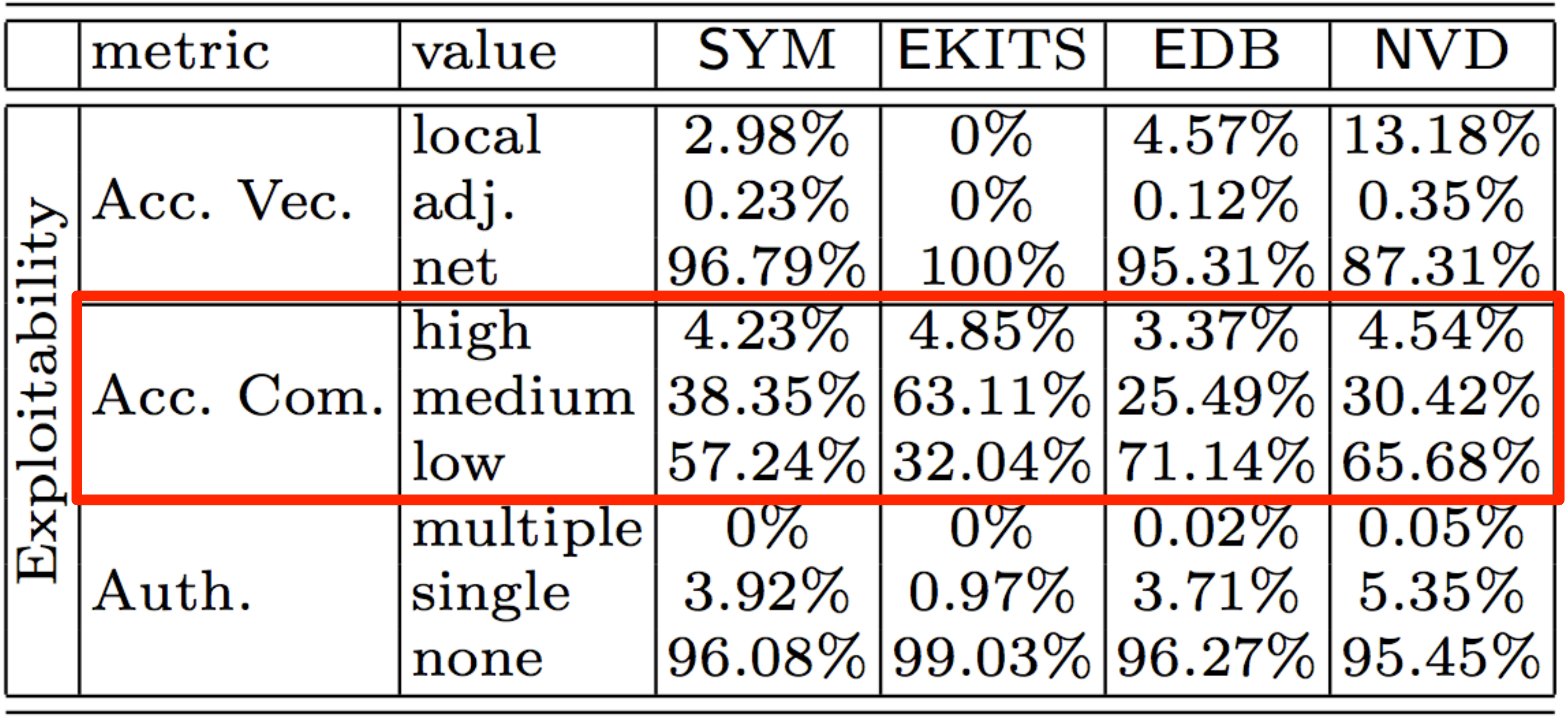
CVSS Metrics: Exploitability



CVSS Metrics: Exploitability explained

- Everything is exploitable → Exploitability is not an interesting variable at all!
  - Is actually a constant
- CVSS lacks of any real measure of likelihood
  - Based on "easiness to exploit"
    - Access Vector = All from Network VAR ≅ 0
    - Authentication = All None VAR ≅ 0
    - Access Complexity = Only interesting variable. VAR != 0
- Let's see what effects does this have to the final CVSS assessment

CVSS Metrics: Exploitability



	metric	value	SYM	EKITS	EDB	NVD
Exploitability	Acc. Vec.	local	2.98%	0%	4.57%	13.18%
		adj.	0.23%	0%	0.12%	0.35%
		net	96.79%	100%	95.31%	87.31%
	Acc. Com.	high	4.23%	4.85%	3.37%	4.54%
		medium	38.35%	63.11%	25.49%	30.42%
		low	57.24%	32.04%	71.14%	65.68%
	Auth.	multiple	0%	0%	0.02%	0.05%
		single	3.92%	0.97%	3.71%	5.35%
		none	96.08%	99.03%	96.27%	95.45%

CVSS case controlled study

- We test the CVSS score against exploitation
  - First step: build the population of vulns
    - Cannot compare apples with oranges
  - Second step: test the CVSS score
    - Does High CVSS predict exploitation?

CVSS case controlled study

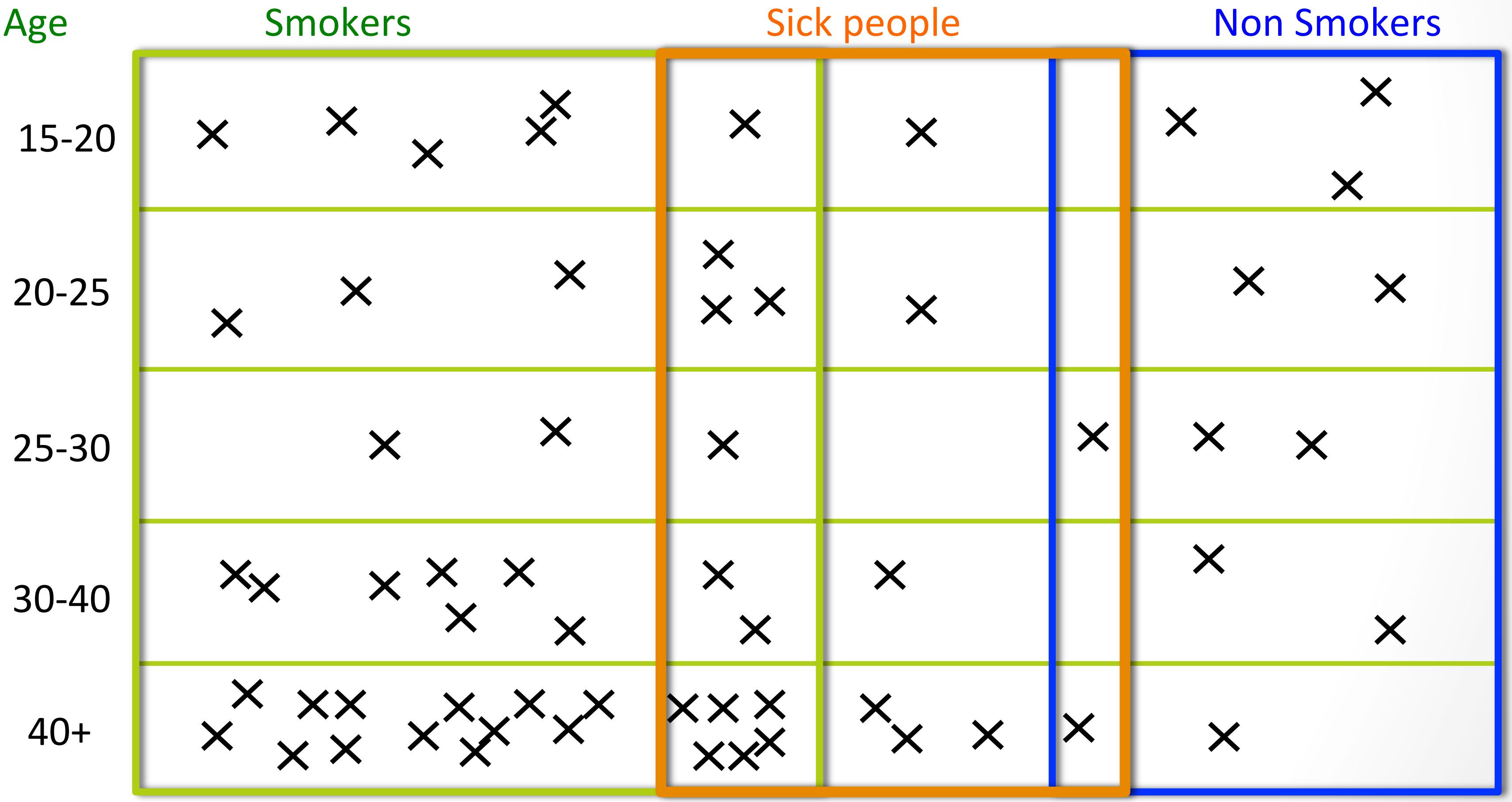
- 1st step
- Do smoking habits predict cancer? [Doll & Bradfor Hill, BMJ]
  - You can’t ask people to start smoking so you can’t run a controlled experiment

- Do high CVSS scores predict exploitation?
  - You can’t attack users so you can’t run a controlled experiment

How to perform a case-controlled observational study

- Instead of performing an experiment, one can still make a observational study
  - Experiment:
    - You control and experimental environment and get the results
  - Observation:
    - You get the results and control the population that generated it
- Let’s use the smokers example
- You can’t pick up people at random
- You need of course smokers, non smorkers and sick people

How to perform a case-controlled observational study



CVSS case controlled experiment

Study	Cases	Controls (possible confounding variables)	Explanatory variable
Carcinoma of the lung	People with cancer	• Age • Sex • Location	• Smoke much • smoke some • Doesn't smoke
CVSS	Exploited vulnerabilities	• Access complexity • Access vector • Authentication • Impact type	• CVSS is HIGH • CVSS is LOW • Vuln is in {NVD,EDB,EKITS}

CVSS case controlled experiment

- 2nd step
- CVSS Score+DB as a “medical test”

- Sensitivity → Pr(true positives)
  - You want to capture as many sick people as possible

    Pr(test said: you’re sick | you are sick)
- Specificity → Pr(true negatives)
  - You REALLY don’t want to cure people who don’t need it

    Pr(test said: you’re **not** sick | you are **not** sick)

CVSS Case Controlled Experiment

- Triple Blood Test Down Syndrome - Women aged 40+ [Kennard 1997]
  - Sensitivity: 69%
    - 31% of women carrying a fetus with Down syndrome will not be caught by the test
  - Specificity: 95%
    - only 5% of healthy pregnant women would be mislead by the test to undergo additional expensive or dangerous tests
  - Remember: most (but really a lot of) women have healthy pregnancies
- Prostate Serum Antigen - Men aged 50+ [Labrie 1992]
  - Sensitivity: 81%
  - Specificity: 90%

Security Rating as "Generate Panic" test

- Sensitivity: is High/Med CVSS good marker for v∈SYM?

Sensitivity = Pr(HIGH+MED | v in SYM)

- Specificity: is Low CVSS good marker for v∉SYM?

Specificity = Pr(LOW | v not in SYM)

Security Rating as “Generate Panic” test

DB	Sensitivity	Specificity
EKITS	89.17%	49.73%
EDB	98.14%	24.39%
NVD	89.70%	22.22%
3BT: Down Syndrome	69%	95%
PSA: Prostate Cancer	81%	90%

Security Rating as "Generate Panic" test - Explained

- **Sensitivity (+)**
  - CVSS is good in marking exploitation
- **Specificity (-)**
  - Peaks in NVD and EDB at less than 25%
  - 1 out of 4 non-exploited vulnerabilities are marked LOW
  - 3 out of 4 non-exploited vulnerabilities are marked HIGH
- Remember this is a controlled study:
  - We are looking only at vulnerabilities representative of SYM CVSS
- Let's assume linearity of cost for number of fixed vulnerabilities
- You are following US Governement SCAP Guidelines? -> You are spending up to 300% more money than you should

Plug this in into the general risk

- Baye’s theorem of conditional probability
- Assume that I have fixed a HIGH score vulnerability
  - What is the probability that this will prevent the attacker from infecting me?

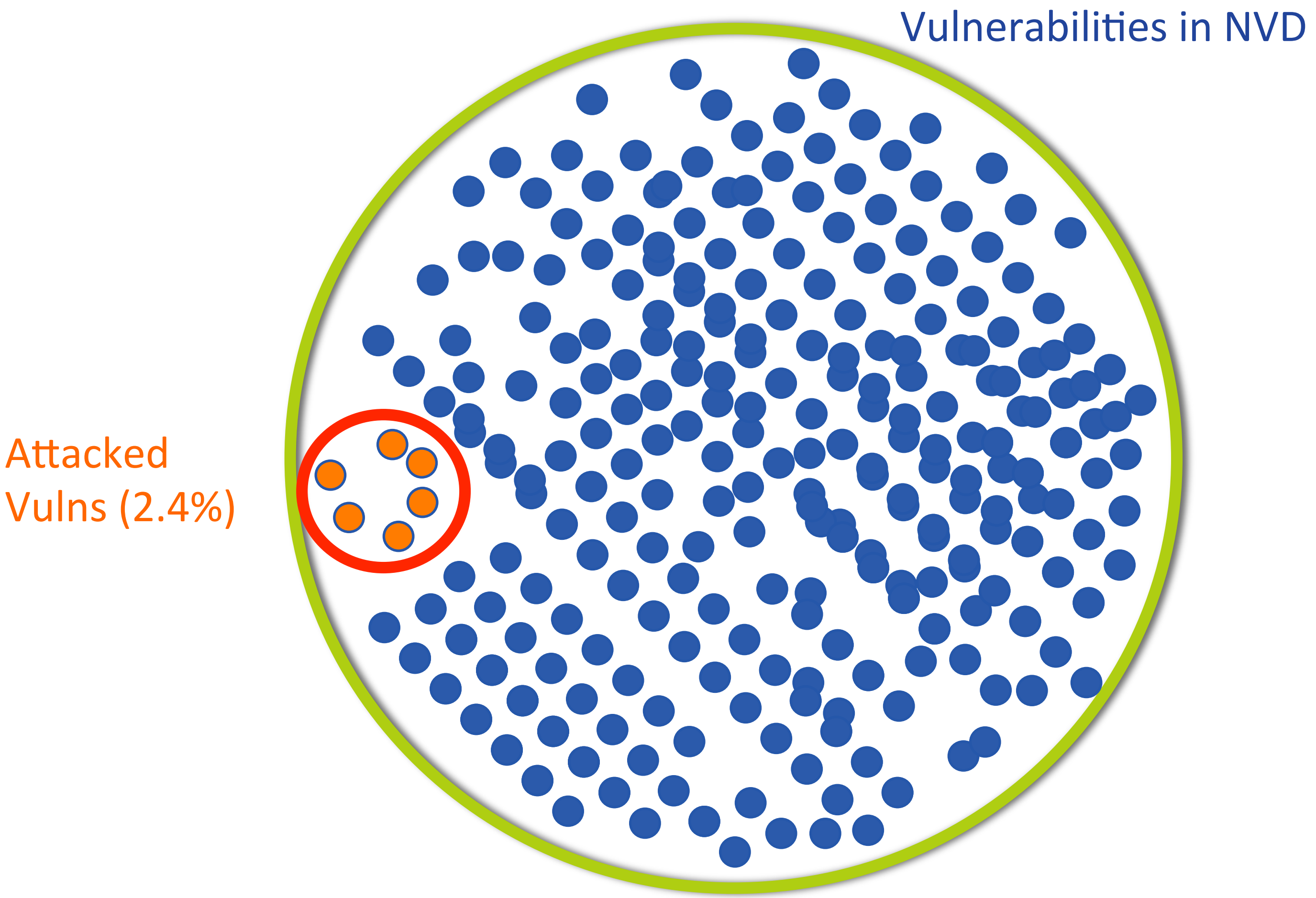
Pr(v in SYM | v patched)

- So, we have:
  - 1200 attacked vulns / 50000 vulns = 2.4%
  - Sensitivity = Probability that an attacked vuln gets HIGH risk score = 89.7%
  - 1- Specificity = Probability that a non-attacked vuln gets HIGH risk score = 87.8%

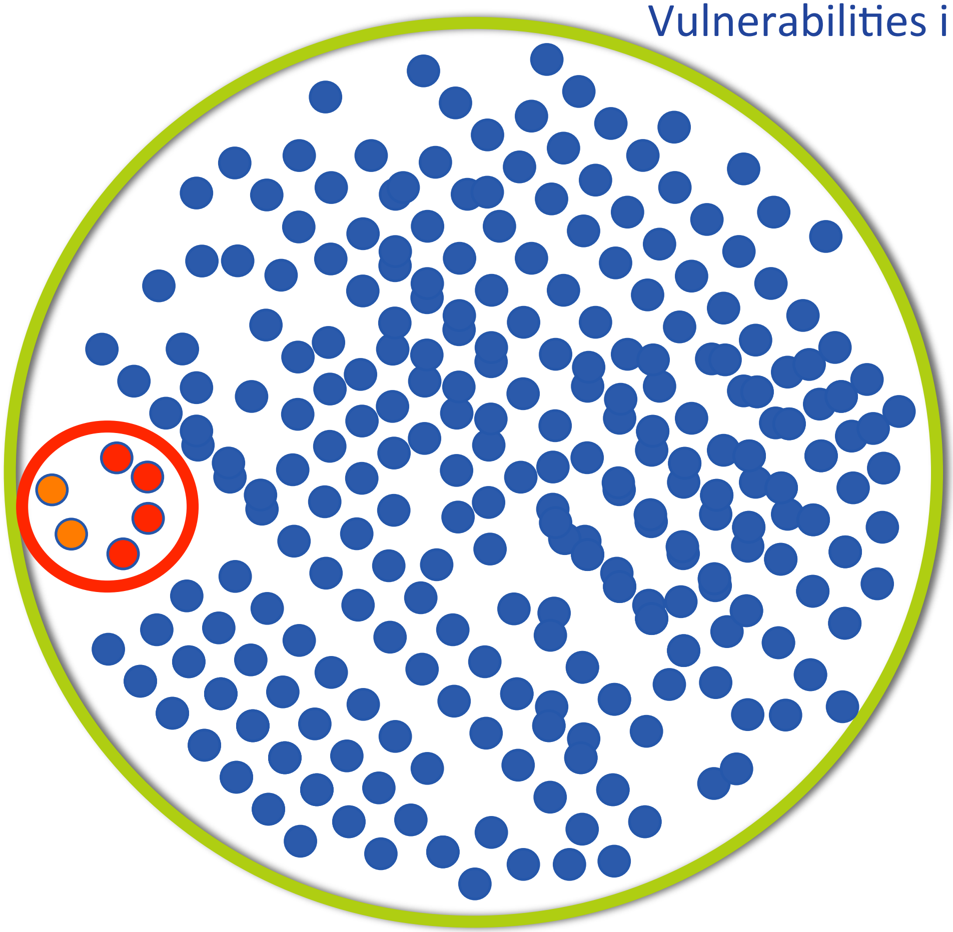
Visualizing it

Vulnerabilities in NVD

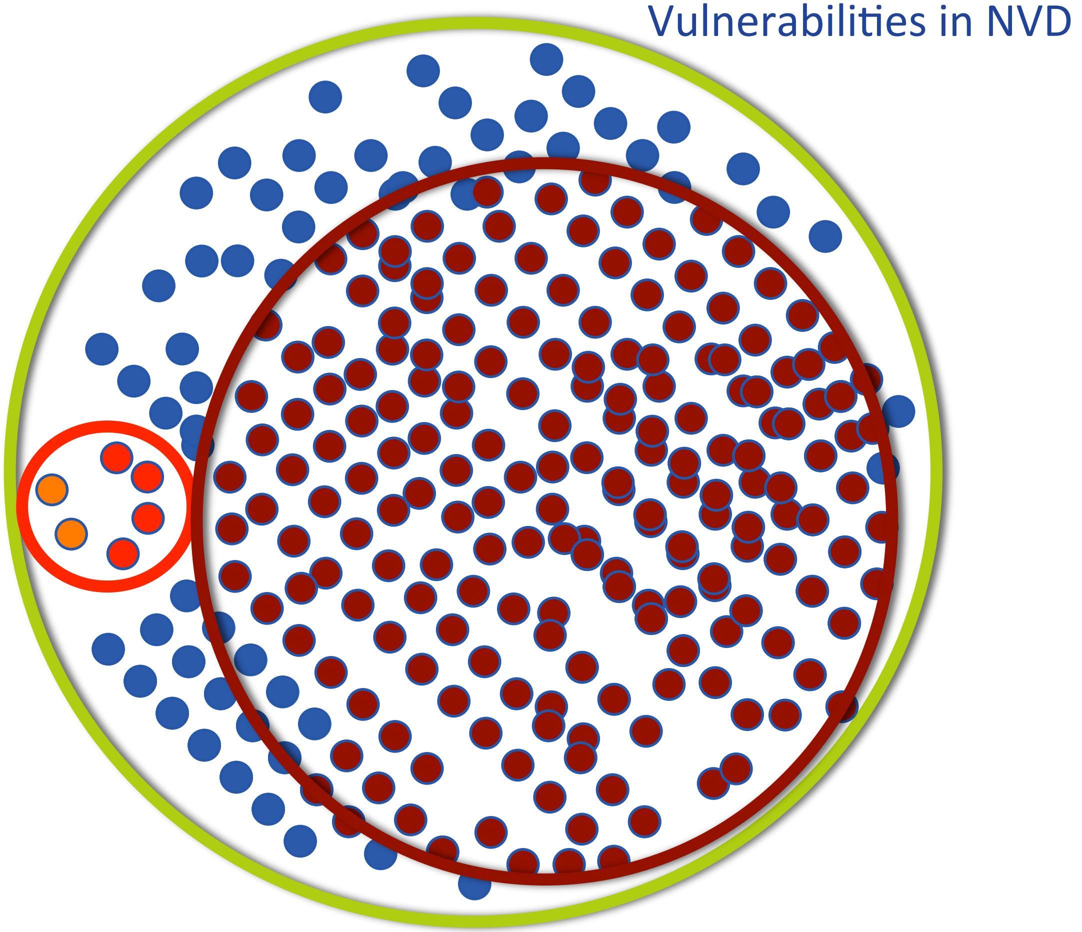
Visualizing it



Visualizing it



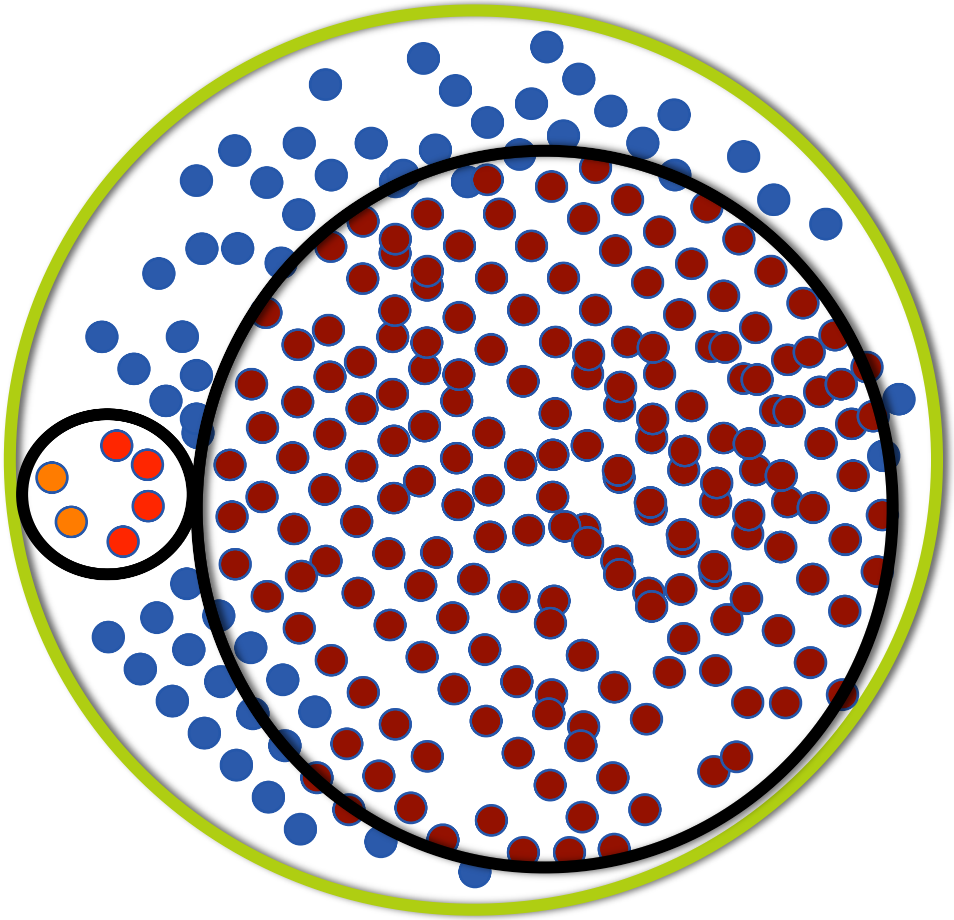
Visualizing it



87.8% of NON attacked Vulns are scored HIGH

Visualizing it

Pr(v in SYM | v patched)*=2.38%



*For the sake of simplicity we do not control the population here, but numbers don't change much

Ok, but is this at least the best decision I can make?

- What really matters is change in relative probabilities

- Example = Usage of Safety Belts
  - Few people actually die in car crashes vs #crashes [Evans 1986]
  - Pr(Death x Safety Belt on) – Pr(Death x Safety Belt off)
  - 43% improvement of chances of survival
- Our Study = Patching High score vulnerabilities
  - Few vulnerabilities are actually exploited vs #vulns
  - Pr(Attack x CVSS High Patched) – Pr(Attack x CVSS Low Patched)
  - X% improvement of chances of NOT being attacked

Not really, no.

	Pr(H+M)-Pr(L)
EKIT	
vuln in SYM	+46.3%
vuln !in SYM	-47.28%
EDB	
vuln in SYM	+14.5%
vuln !in SYM	-14.49%
NVD	
vuln in SYM	+3.5%
vuln !in SYM	-3.46%

What does this mean?

- What the CIO really wants to know:
  - I read on the news that a "security researcher" exploited a vulnerability on X to do some bad stuff. Should I worry?
- You monitor the black markets and fix all HIGH CVSS vulnerabilities you find there?
  - Your risk of suffering from an attack from the black markets decreases by 46%
- You use EDB or NVD to know what exploits are out there, and fix all HIGH CVSS vulnerabilities?
  - Diminished risk: EDB = 14%; NVD = 3%.
  - Arguably a bad investment

Preliminary conclusions

- Where should we look for "real" exploits?
  - EDB, NVD are the wrong datasets
- Should the CIO do what SCAP protocol says?
  - No datasets shows high Specificity:
    - CVSS doesn't rule out "un-interesting" vulns
    - Huge over-investment
- It may be possible to narrow down vulnerabilities the CIO should actually fix
  - Rule out 80% of risk = worth the update pain, measurable gain
  - We need better attacker model -> Research challange ahead

Questions

- You can also mail me for anything
- If you are interested in a PhD@UniTn feel free to exploit me for info
  - luca.allodi@unitn.it

- http://disi.unitn.it/~allodi/

- Papers, current research, challenges:
- https://securitylab.disi.unitn.it/