Quantitative Security

Colorado State University

Yashwant K Malaiya

CS559

L10

CSU Cybersecurity Center
Computer Science Dept

L10 Outline

- Vulnerabilities
  - Finders
  - Classification
  - CVE numbering system
  - Rewards and sale
  - Data bases
  - Life Cycle

Quantitative Security

Colorado State University

Yashwant K Malaiya

CS559

Vulnerabilities

CSU Cybersecurity Center
Computer Science Dept

Security Holes: Types

- Software holes: Vulnerabilities
  - CVSS scores involving *exploitability* and *impact* is a type of risk measure.
- System/physical holes
- Personnel/Procedural holes:
  - e.g. Phishing
- Exploitation may involve multiple holes, perhaps of different types
- Classify them:
  - Target 2013 breach: credentials stolen from a HVAC contractor
  - Equifax 2017 breach: vulnerability patch not applied

Defects vs vulnerabilities

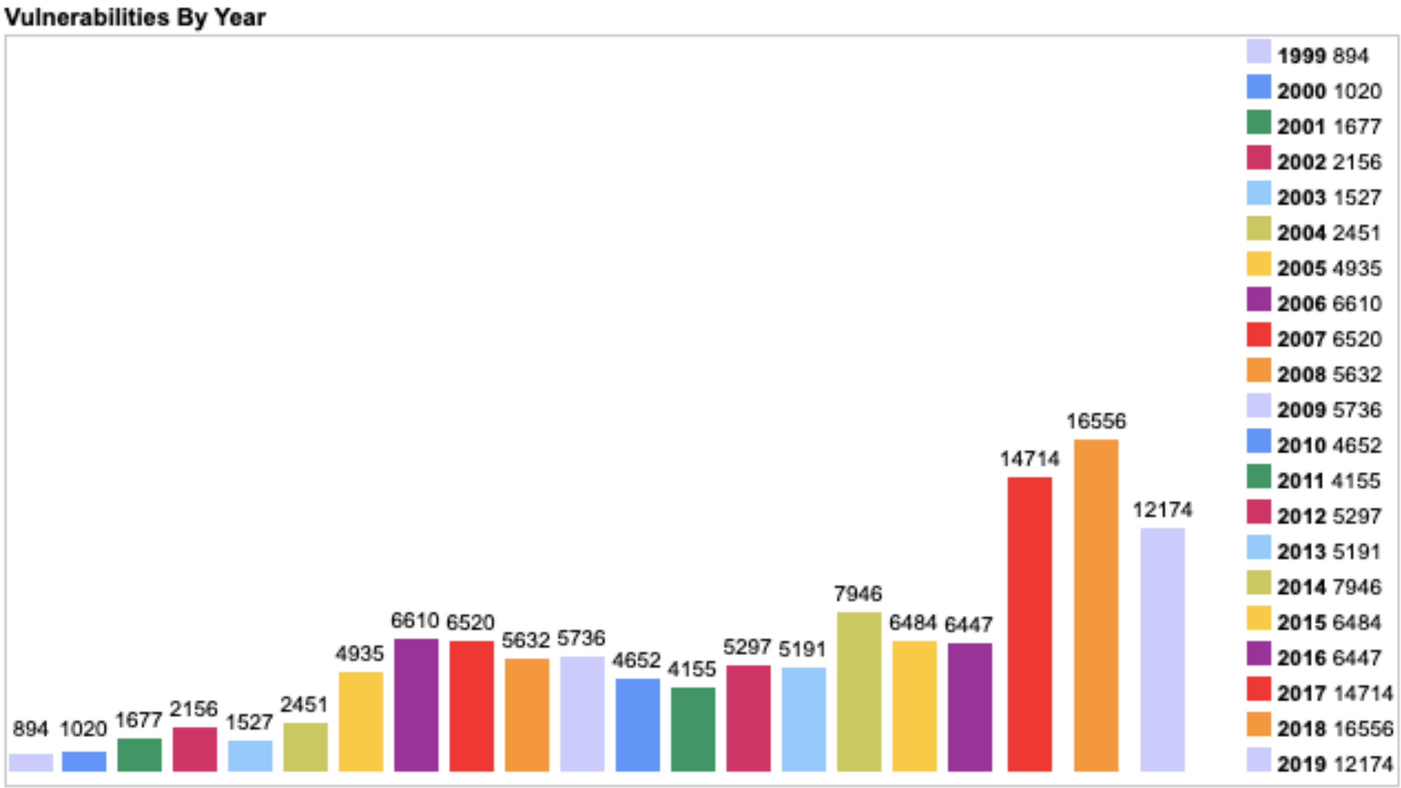
- Software defects
- Software Vulnerabilities
- Testing:
  - prior to release
  - Usage
  - Field testing

Components of Likelihood of Exploitation

- Internal
  - Presence of a vulnerability (Vulnerability Discovery*)
  - Vulnerability not patched
- External
  - Attacker’s motivation level
  - Technical capabilities, exploit availability*
  - Network access to vulnerable system
- Interface
  - Attack surface* of vulnerable system
  - Reachability* of vulnerability

We have published research related to these. See publications 2005-2016.

Vulnerabilities Trend

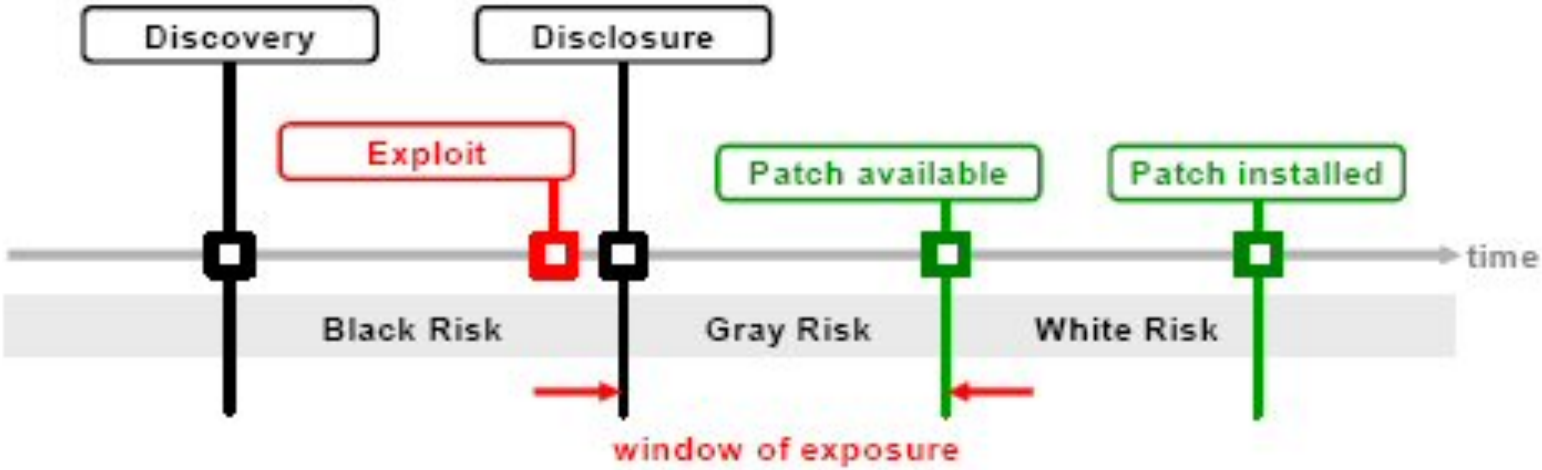


Vulnerability Data-bases

[NVD database]
[CVEDetails]
[VulnDB]
[ExploitDB]

https://www.cvedetails.com/browse-by-date.php

Vulnerability Lifecycle



Colorado State University
8

Vulnerability density and defect density

- **Vulnerability densities**: 95/98: 0.003-0.004 NT/2000/XP: 0.01-0.02
- **V_{KD}/D_{KD}**: 0.68-1.62% about 1%

System	MSLOC	Known Defects (1000s)	D_{KD} (/Kloc)	Known Vulner - abilies	V_{KD} (/Kloc)	Ratio V_{KD} /D_{KD}
Win 95	15	5	0.33	46	0.0031	0.92%
NT 4.0	16	10	0.625	162	0.0101	1.62%
Win 98	18	10	0.556	84	0.0047	0.84%
Win2000	35	63	1.8	508	0.0145	0.81%
Win XP	40	106.5*	2.66*	728	0.0182	0.68%*

Alhazmi, Malaiya, Ray, " Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems," Computers and Security, May 2007, P 219-228.

Who discovers vulnerabilities?

Discoverers	Safari's Vulnerabilities	Percentage	Chromium's Vulnerabilities	Percentage
Product's company discoverers	17	20%	0	0%
Product's company discoverers and others	0	0%	35	35%
Outside discoverers	66	80%	63	64%
Unknown discoverers	0	0%	1	1%

Vulnerability discoverers from July. 1, 2012 to December 31, 2012: insiders or outsiders

A. M. Algarni, and Y. K. Malaiya, "Most Successful Vulnerability Discoverers: Motivation and Methods", Int. Conference on Security and Management (SAM 2013), Las Vegas, July 2013, pp. 3-9.

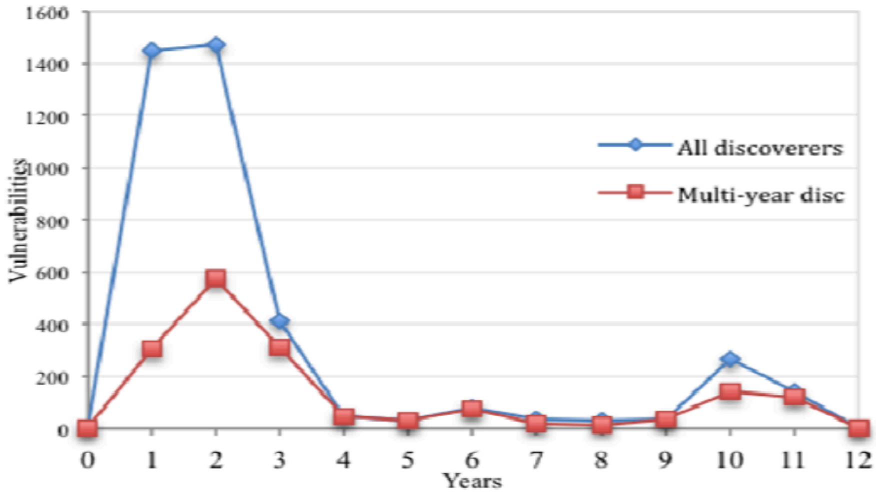
THE TOP VULNERABILITIES DISCOVERERS ON OSVDB

THE TOP VULNERABILITIES DISCOVERERS ON OSVDB

Discoverer	Country	Period	# Vuln	# Vuln types	Why they're interested	Stopped/Continued
r0t	Latvia	2005-08-09 to2010-09-16	810	10	N/A	N/A
Janek Vind "waraxe"	Estonia	2003-08-08 to2013-03-21	319	8	Vulnerability website	N/A
Lostmon Lords	Spain	2004-06-20 to2009-08-15	279	8	Security Researcher	Worked until July 2012
rgod	Italy	2005-06-06 to2012-08-29	277	12	Hacker	Worked until Aug. 2012
Luigi Auriemma	Italy	2000-07-08 to2013-03-16	267	9	Hobby	N/A
Russ McRee	USA	2008-01-14 to 2012-03-02	237	4	Specialist in security	N/A
Aliaksandr Hartsuyeu	Lithuania	2005-12-28 to2011-02-03	229	6	Security Company	Still working 2012
James Bercegay	USA	2003-06-03 to2008-09-04	200	12	Web developer	Worked until 2011
Kacper	Poland	2006-05-12 to2007-08-10	199	3	N/A	N/A
luny	N/A	2006-05-18 to 2006-07-13	142	6	N/A	N/A
Diabolic Crab	N/A	2004-09-25 to 2005-07-12	140	6	N/A	N/A
JeiAr	USA	2003-05-29to 2004-05-04	120	7	Web developer	Worked until 2011
Tan Chew Keong	Singapore	2004-07-29 to 2009-09-28	102	9	Information Security Specialist	N/A
Stefan Esser	Germany	2000-11-09 to 2012-06-03	86	10	Security Consultant	Still do jailbreak until 2012
M.Hasran Addahroni	Indonesia	2006-02-09 to 2009-02-07	80	2	Security Gossiper&Bugs Hunter	N/A

A. M. Algarni, and Y. K. Malaiya, "Most Successful Vulnerability Discoverers: Motivation and Methods", Int. Conference on Security and Management (SAM 2013), Las Vegas, July 2013, pp. 3-9.

What happend to discoverers?

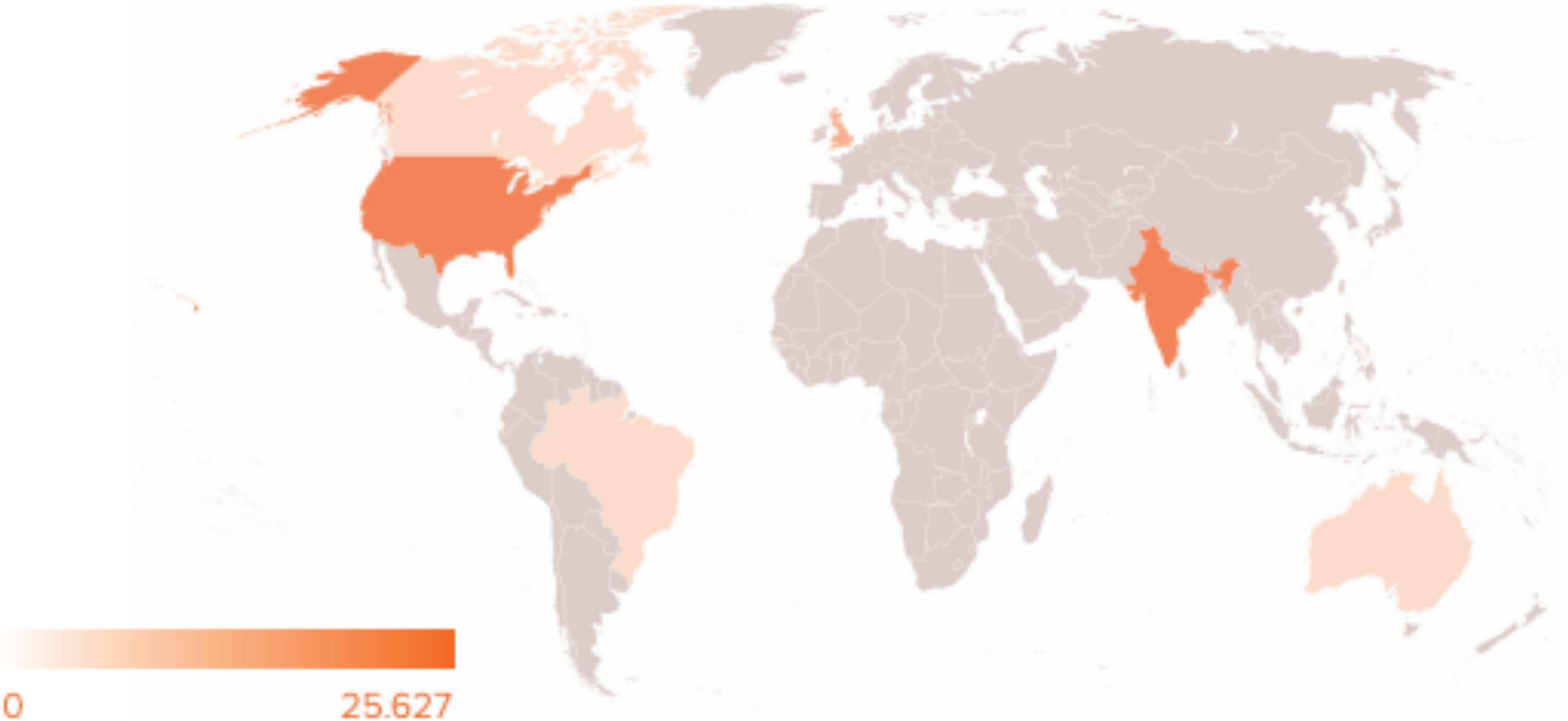


Most of the top discoverers here are credited with discovering the vulnerabilities during the first three years. However, a few discoverers have continued to discover vulnerabilities for several years.

Why do some very successful discoverers disappear from the scene after two to three years?

- A possible explanation is that, during those two to three years, they acquire the notoriety of being accomplished vulnerability discoverers.
- After that, they start offering their services to software developers or security service companies on a contract basis or as employees.
- Some of them may be able to start their own small organizations.

Vulnerability finders in 2019



Participants in bug bounty programs
https://static.carahsoft.com/concrete/files/2215/7296/5388/Bugcrowd_Priority_One_Report_2019.pdf

Colorado State University

Classification of Vulnerabilities

OWASP Top Ten
Open Web Application Security Project a nonprofit that works to improve the security of software

- A1 Injection: Sending hostile data to an interpreter (e.g. SQL, LDAP, command line)
- A2 Broken Authentication:
  - Weak session management
  - Credential stuffing
  - Brute force
  - Forgotten password
  - No multi-factor authentication
  - Sessions don't expire
- A3 Sensitive Data Exposure
  - Clear-text data transfer
  - Unencrypted storage
  - Weak crypto or keys
  - Certificates not validated
  - Exposing PII or Credit Cards
- A4 XML External Entities (XXE)
  - The application accepts XML, and assumes it is safe
- A5 Broken Access Control
  - Access hidden pages
  - Elevate to an administrative account
  - View other people's data
  - Modifying cookies or JWT tokens

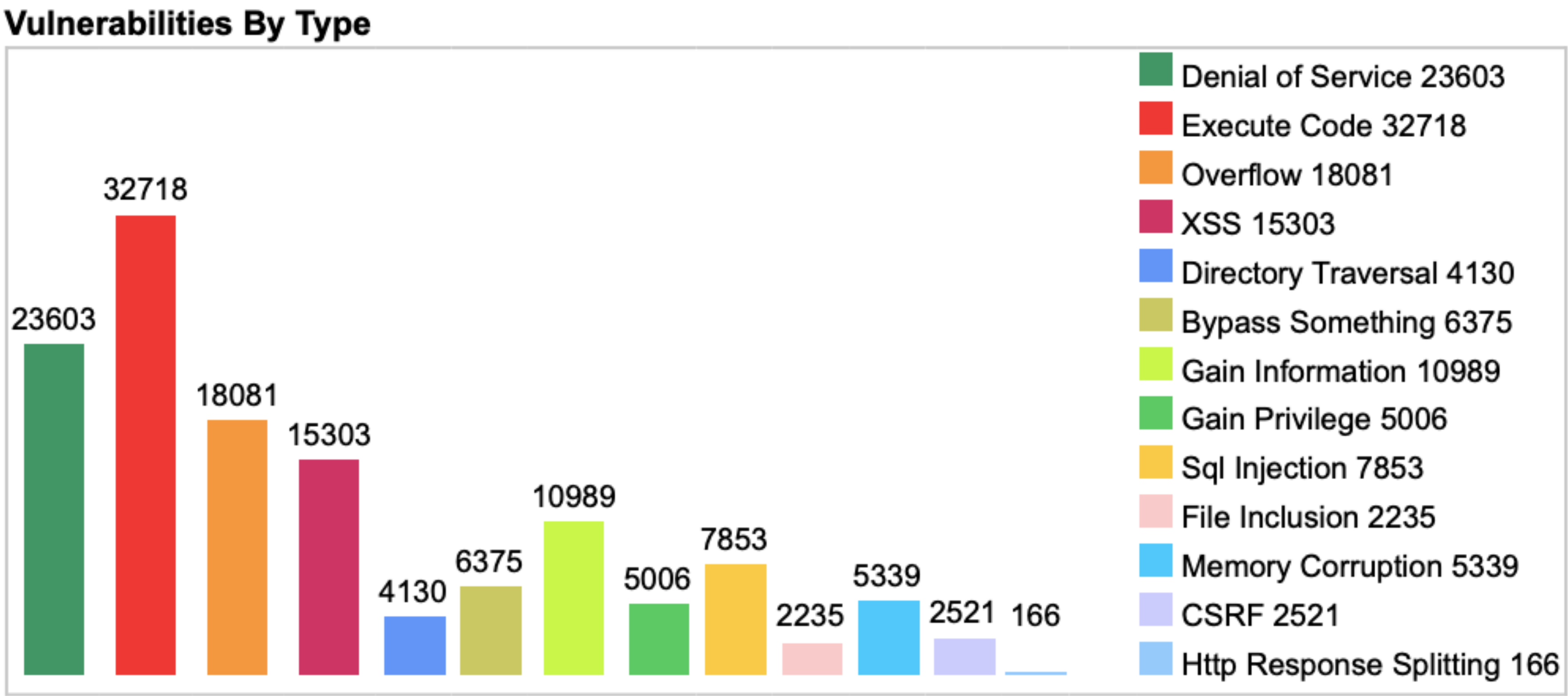
Introduction to the OWASP Top Ten Video

Classification of Vulnerabilities

OWASP Top Ten 2017

- A6 Security Misconfiguration
  - Security features not configured properly
  - Unnecessary features enabled
  - Default accounts not removed
  - Error messages expose sensitive information
- A7 Cross-Site Scripting (XSS)
  - HTML mixes content, presentation and code into one string (HTML+CSS+JS)
  - If an attacker can alter the DOM, they can do anything that the user can do.
  - XSS can be found using automated tools.
- A8 Insecure Deserialization
  - Programming languages allow you to turn a tree of objects into a string that can be sent to the browser.
  - If you deserialise untrusted data, you may allow objects to be created, or code to be executed.
- A9 Using Components with Known Vulnerabilities
  - Modern applications contain a lot of third-party code.
  - It's hard to keep it all up to date.
  - Attackers can enumerate the libraries you use and develop exploits.
- A10 Insufficient Logging & Monitoring
  - You can't react to attacks that you don't know about.
  - Logs are important for: Detecting incidents, Understanding what happened, Proving who did something

Vulnerability Classification CVE Details



https://www.cvedetails.com/vulnerabilities-by-types.php
(1999-2019)
Note: A vulnerability can have multiple types.

Classification CVE Details

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
1999	894	177	112	172			2	7		25	16	103			2
2000	1020	257	208	206		2	4	20		48	19	139			
2001	1677	403	403	297		7	34	123		83	36	220		2	2
2002	2156	498	553	435	2	41	200	103		127	74	199	2	14	1
2003	1527	381	477	371	2	49	129	60	1	62	69	144		16	5
2004	2451	580	614	410	3	148	291	111	12	145	96	134	5	38	5
2005	4935	838	1627	657	21	604	786	202	15	289	261	221	11	100	14
2006	6610	893	2719	663	91	967	1302	322	8	267	271	184	18	849	30
2007	6520	1101	2601	954	95	706	884	339	14	267	324	242	69	700	44
2008	5632	894	2310	699	128	1101	807	363	7	288	270	188	83	170	74
2009	5736	1035	2185	700	188	963	851	322	9	337	302	223	115	138	738
2010	4652	1102	1714	680	342	520	605	275	8	234	282	238	86	73	1493
2011	4155	1221	1334	770	351	294	467	108	7	197	409	206	58	17	557
2012	5297	1425	1459	843	423	243	758	122	13	344	389	250	166	14	624
2013	5191	1455	1186	859	366	156	650	110	7	352	511	274	123	1	205
2014	7946	1598	1574	848	420	305	1105	204	12	457	2106	239	264	2	401
2015	6484	1791	1826	1083	749	218	778	150	12	577	748	367	248	5	127
2016	6447	2028	1494	1324	717	94	497	99	15	444	843	600	87	7	1
2017	14714	3154	3004	2495	745	508	1518	279	11	629	1639	459	327	18	6
2018	16556	1853	3041	2368	400	517	2042	531	11	708	1424	247	461	31	4
2019	12174	919	2277	1247	296	410	1593	280	4	495	900	129	398	40	
Total	122774	23603	32718	18081	5339	7853	15303	4130	166	6375	10989	5006	2521	2235	4333
% Of All		19.2	26.6	14.7	4.3	6.4	12.5	3.4	0.1	5.2	9.0	4.1	2.1	1.8	

https://www.cvedetails.com/vulnerabilities-by-types.php (1999-2019)

Top ten software flaws used by crooks

According to the Recorded Future Annual Vulnerability report in 2019:

- CVE-2018-15982 – Adobe Flash Player
- CVE-2018-8174 – Microsoft Internet Explorer
- CVE-2017-11882 – Microsoft Office
- CVE-2018-4878 – Adobe Flash Player
- CVE-2019-0752 – Microsoft Internet Explorer
- CVE-2017-0199 – Microsoft Office
- CVE-2015-2419 – Microsoft Internet Explorer
- CVE-2018-20250 – Microsoft WinRAR
- CVE-2017-8750 – Microsoft Internet Explorer
- CVE-2012-0158 – Microsoft Office

The year is the year of disclosure. Most vulnerabilities are disclosed and patched at the same time.

https://www.zdnet.com/article/these-are-the-top-ten-software-flaws-used-by-crooks-make-sure-youve-applied-the-patches

CVE numbering system

CVE, (Common Vulnerabilities and Exposures), is a list of publicly disclosed computer security flaws.

- A CVE, usually means the CVE ID number assigned to a security flaw ("vulnerability").
- Security advisories issued almost always mention at least one CVE ID.
- CVE is overseen by the MITRE corporation
  - funded from the Cybersecurity and Infrastructure Security Agency, part of the U.S. Department of Homeland Security
- CVE identifiers are assigned by a CVE Numbering Authority (CNA).
  - There are about 100 CNAs, including major IT vendors as well as security companies and research organizations. MITRE can also issue CVEs directly.
  - CVE reports can come from a vendor, a researcher, or anyone who has discovered a flaw
  - There are mechanisms for responsible disclosure.
  - a CVE ID is assigned before a security advisory is made public. Vendors generally keep security flaws secret until a fix has been developed and tested.
  - Once made public, a CVE entry includes
    - the CVE ID (in the format "CVE-2019-1234567"),
    - a brief description of the security vulnerability or exposure, and
    - references, which can include links to vulnerability reports and advisories.

Is it a vulnerability?

CNAs have the discretion to determine whether something is a vulnerability. https://cve.mitre.org/cve/cna/rules.html#section_7-1_what_is_a_vulnerability

- Root CNAs may provide additional guidance to their child CNAs. This allows the program to adapt to definitions used in different industries, legal regimes, and cultures.
  - If a product owner considers an issue to be a vulnerability in its product, then the issue MUST be considered a vulnerability, regardless of whether other parties (e.g., other vendors whose products share the affected code) agree.
  - If the CNA determines that an issue violates the security policy of a product, then the issue SHOULD be considered a vulnerability.
  - If a CNA receives a report about a new vulnerability that has a negative impact, then the reported vulnerability MAY be considered a vulnerability.
- If a *weakness* cannot be exploited by an attacker, it is a weakness, not a *vulnerability*. CWE stands for *Common Weakness Enumeration*.

Responsible Disclosure

A vulnerability discoverer may

- Submit a report to the company with a Vulnerability Disclosure Program
  - If they have a disclosure program, they will acknowledge the report.
  - Evaluate the report to see if the report is valid.
  - Provide a reward if they have a Reward Program
  - If they do no respond within a fixed time (90-120 days), the discoverer may disclose the vulnerability.
- Sell it to a vulnerability broker (TrendMicro ZDI, iDefence VCP etc.)
- Sell it to a government organization
- Sell it in the black market

Vulnerability Reward programs (2014)

TABLE I
SOME CURRENT VULNERABILITY REWARDS PROGRAMS

Program	# Vulns. type	Max reward	Min reward	# of beneficiaries	Trend
Vulnerability Reward Program for Google web properties	5	$20,000	$100	2010: 51 2011: 122 2012: 189 2013: 226	Increase
Chrome Vulnerability Reward Program	Any security bug	>= 10,000	$500	543	N/A
The Mozilla Security Bug Bounty Program	Certain bugs depending on some criteria	$3000 (US) cash reward and a Mozilla T-shirt	$500	N/A	N/A
Facebook	Certain qualifying security bugs	No maximum	$500	Prior to 2011: 43 2011: 46 2012: 111 2013: 235	Increase
WordPress Security Bug Bounty Program	11	$1000	$25	N/A	N/A
CCBill Vulnerability Reward Program	7	$ 500	$300	42	Hold
Secunia Vulnerability Coordination Reward Program (SVCRP)	Most bugs depending on some criteria	Most Valued Contributor& Most Interesting Coordination Report	N/A	N/A	N/A
ZDI Rewards Program (TippingPoint)	Particular bugs depending on some criteria	$25,000	$1000	N/A	N/A
iDefense (Verisign)	N/A	N/A	N/A	Significant number	N/A

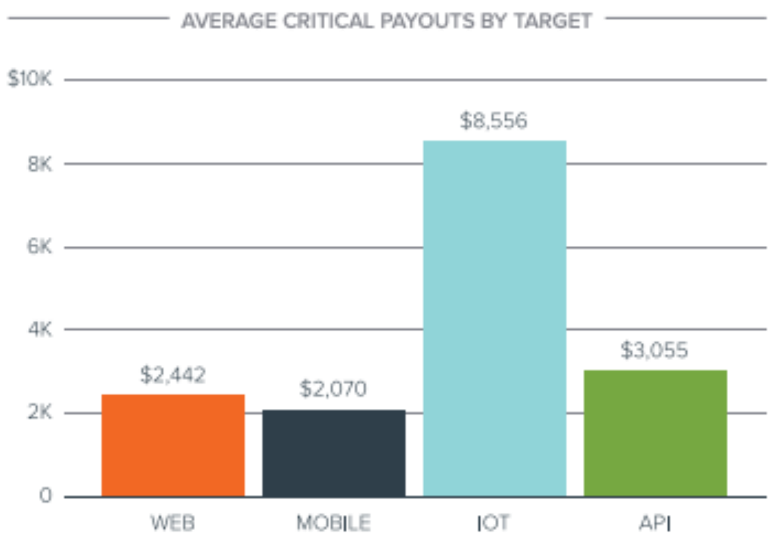
Colorado State University

Average Reward level

Bugcrowd Priority One Report 2019

Total payouts increased 83% year over year.

- The average payout for a critical vulnerability in 2019 is $2,669.92, a 27% increase year over year.
- In the first half of 2019, we saw a 29% increase in the number of programs launched versus the same time the year before and a 50% increase in public programs launched.
- Submissions have increased 92% overall, with submissions on IoT targets increasing more than any other at 384%.
- BugCrowd (?), HackerOne ($23 million in 2018)



Colorado State University

HackerOne Millionaire

..As if Pereira's story isn't enough, we have to mention another 19-year-old South American who is killing the bug bounty game: Argentina's Santiago Lopez, the first person to top $1 million in earnings on HackerOne's platform.

The self-taught hacker says he got his start by watching YouTube videos and reading blogs on his own, but the thing that jumpstarted his interest in hacking? What else? The 1995 movie Hackers. (Photo by United Artists/Getty Images)

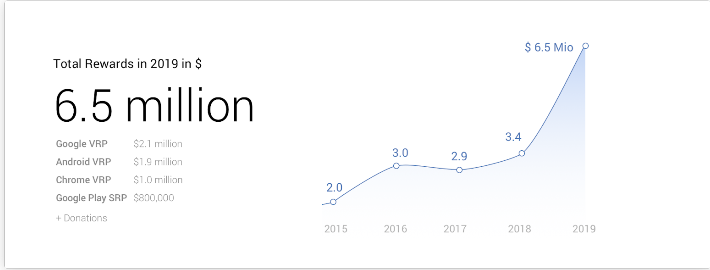
https://www.pcmag.com/news/7-huge-bug-bounty-payouts

Google Reward Programs

	High-quality report with functional exploit	High-quality report	Baseline
Sandbox escape / Memory corruption in a non-sandboxed process	$30,000	$20,000	Up to $15,000
Universal Cross Site Scripting (includes Site Isolation bypass)	$20,000	$15,000	Up to $10,000
Renderer RCE / memory corruption in a sandboxed process	$10,000	$7,500	Up to $5,000
Security UI Spoofing	$7,500	N/A [1]	Up to $3,000
User information disclosure	$5,000 - $20,000	N/A [1]	Up to $2,000
Web Platform Privilege Escalation	$5,000	$3,000	Up to $1,000
Exploitation Mitigation Bypass	$5,000	$3,000	Up to $1,000
Chrome OS	See below		
Chrome Fuzzer Bonus	$1,000		
Chrome Patch Bonus	$500 - $2,000		

[1] For these classes of bugs, high quality reports are expected to demonstrate the UI spoof or show how user information could be disclosed, which we treat as a functional exploit.

Non-qualifying flaws: things that are nearly impossible to exploit, legitimate features, which are not Google's fault, etc.



https://security.googleblog.com/2020/01/vulnerability-reward-program-2019-year.html

Colorado State University

Vulnerabilities for sale

- Stefan Frei, NSS Labs looked at reports about some of those private vendors
  - Endgame Systems, Exodus Intelligence, Netragard, ReVuln and VUPEN
  - concluded that jointly these firms alone have the capacity to sell more than 100 zero-day exploits per year.

Provider	Offering	Remark / Source
Endgame Systems	25 exploits/year USD $2.5 million	*Business Week* http://www.businessweek.com/magazine/cyber-weapons-the-new-arms-race-07212011.html
Exodus Intelligence	60 exploits/year	Service Offering https://www.exodusintel.com/rsrc/ExodusIntelligence_EXP.pdf
ReVuln	> 9 exploits/year	Minimum estimate by counting exploits demonstrated here: http://vimeo.com/53806381 (2013-09-27)
VUPEN	> 7 exploits/year > 15 to 20 binary analysis and private 1-day exploits/month	Minimum estimate by counting list of published exploits here: http://www.vupen.com/blog/ (2013-09-27) Service Offering: http://www.vupen.com/english/services/ba-gov.php

Frei's minimum estimate of exploits offered by boutique exploit providers each year.

https://krebsonsecurity.com/2013/12/how-many-zero-days-hit-you-today/

Vulnerability Data Bases

- Vulnerabilities are security related defects.
- For ordinary defects found, the developers do not have an obligation to report them.
  - They just have to address them in the next patch/version.
- All recognized vulnerabilities are analyzed and reported in the data-bases.
- The new entries in the data-bases are used by antivirus developers/maintainers to add to their signatures.
- Used by researchers like us.

Major vulnerability databases

- NVD National Vulnerability Database https://nvd.nist.gov/
  - US 2005, includes vulnerability operation, its CVSS rating, and links to any available patches/fixes
- CVE Mitre DB https://cve.mitre.org/data/downloads/index.html
- CWE Common weakness enumeration https://cwe.mitre.org/
- VulnDB https://vulndb.cyberriskanalytics.com
  - Open source vulnerability database 2004. Commercia.
- WhiteSource Vuln DB https://www.whitesourcesoftware.com/vulnerability-database/
  - Includes not yet recognized vulnerabilities
- CVEDetails https://www.cvedetails.com better interface?
- Exploit DB https://www.exploit-db.com available exploits

Colorado State University

Example: CVE-2018-8174

A remote code execution vulnerability exists in the way that the VBScript engine handles objects in memory

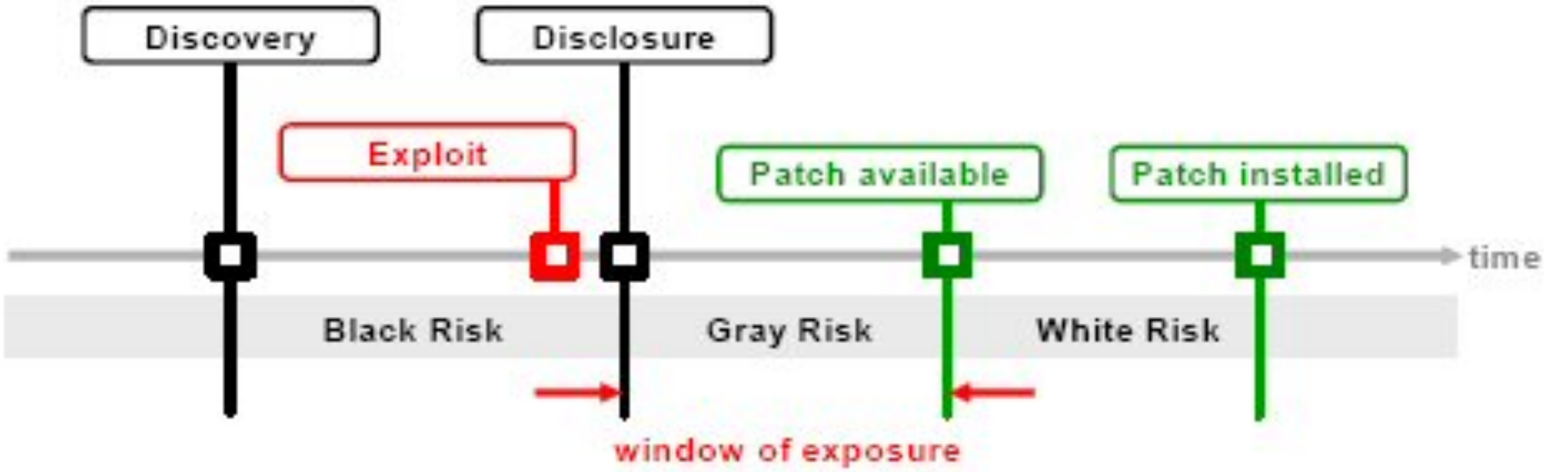
- Microsoft acknowledgements
- CVE Details cvss2.0, products/versions affected, links
- Exploit-db has an exploit

Quantitative Security

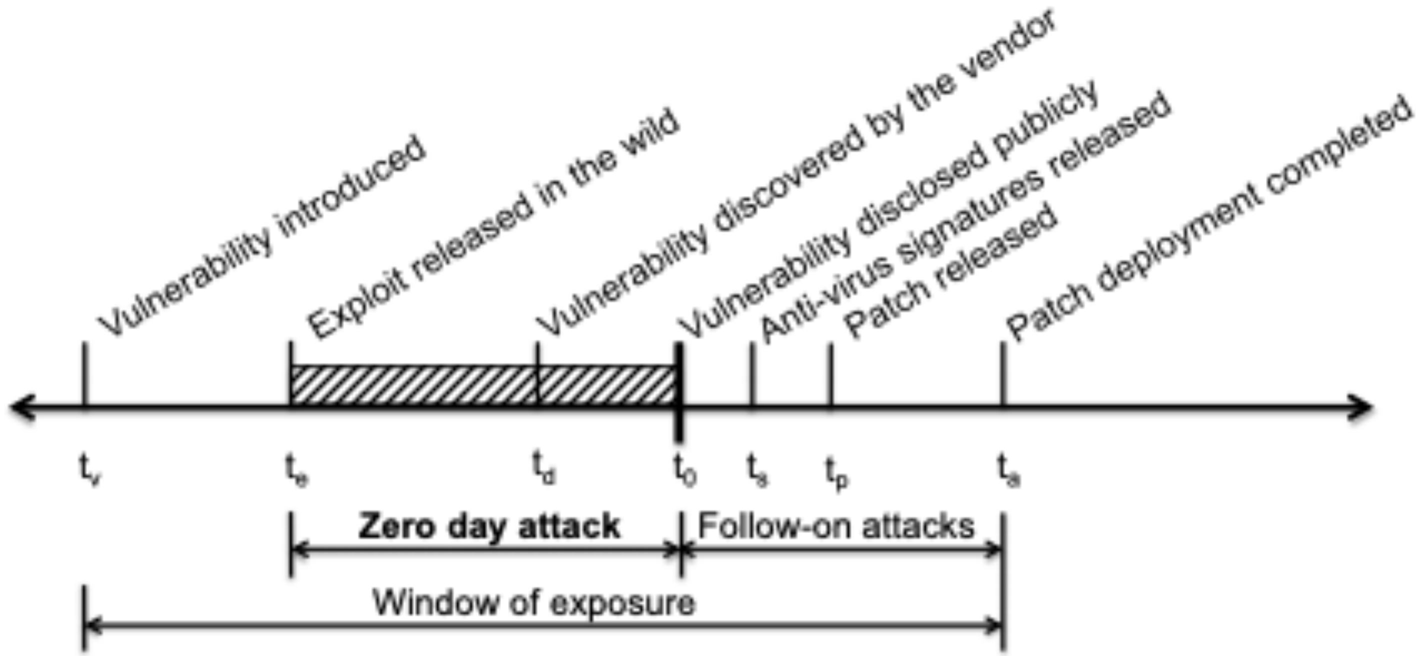
Colorado State University
Yashwant K Malaiya
CS 559
Vulnerability Life Cycle

CSU Cybersecurity Center
Computer Science Dept

Vulnerability Lifecycle



Timeline



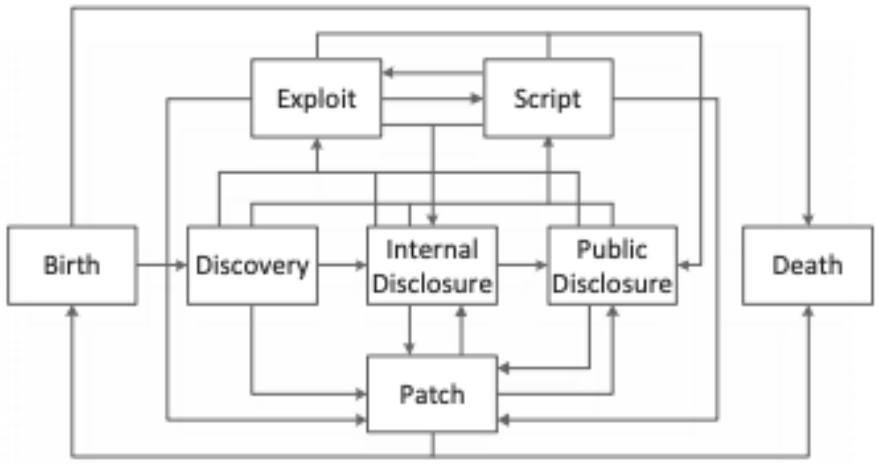
Attack timeline. These events do not always occur in this order, but ta > $t_p \geq t_d > t_v$ and $t_0 \geq t_d$.
The relation between t_d and t_e cannot be determined in most cases. For a zero-day attack t0 > t_e.

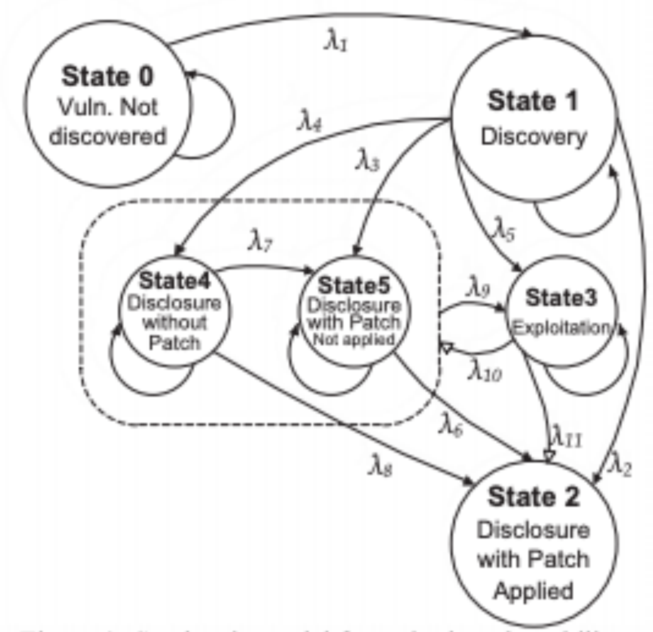
[Before We Knew It An Empirical Study of Zero-Day Attacks In The Real World]

Vulnerability Lifecycle

- **Vulnerability introduced**. A bug is introduced in software (time = t_v).
- **Exploit released in the wild.** Actors in the underground economy discover the vulnerability, create a working exploit and use it to conduct stealth attacks against selected targets (time = t_e)
- **Vulnerability discovered by the vendor.** The vendor learns about the vulnerability, assesses its severity, assigns a priority for fixing it and starts working **on a** patch (time = t_d).
- **Vulnerability disclosed publicly. The** vulnerability is disclosed, either by the vendor or on public forums and mailing lists. A CVE identifier (e.g., CVE-2010-2568) is assigned to the vulnerability (time = t_0).
- **Anti-virus signatures released.** Once the vulnerability is disclosed, anti-virus vendors release new signatures (time = t_s),
- **Patch released.** On the disclosure date, or shortly afterward the software vendor releases a patch for the vulnerability. After this point, the hosts that have applied the patch are no longer susceptible to the exploit (time = t_p)
- **Patch deployment completed.** All vulnerable hosts worldwide are patched and the vulnerability ceases to have an impact (time = t_a).

Stochastic Modeling





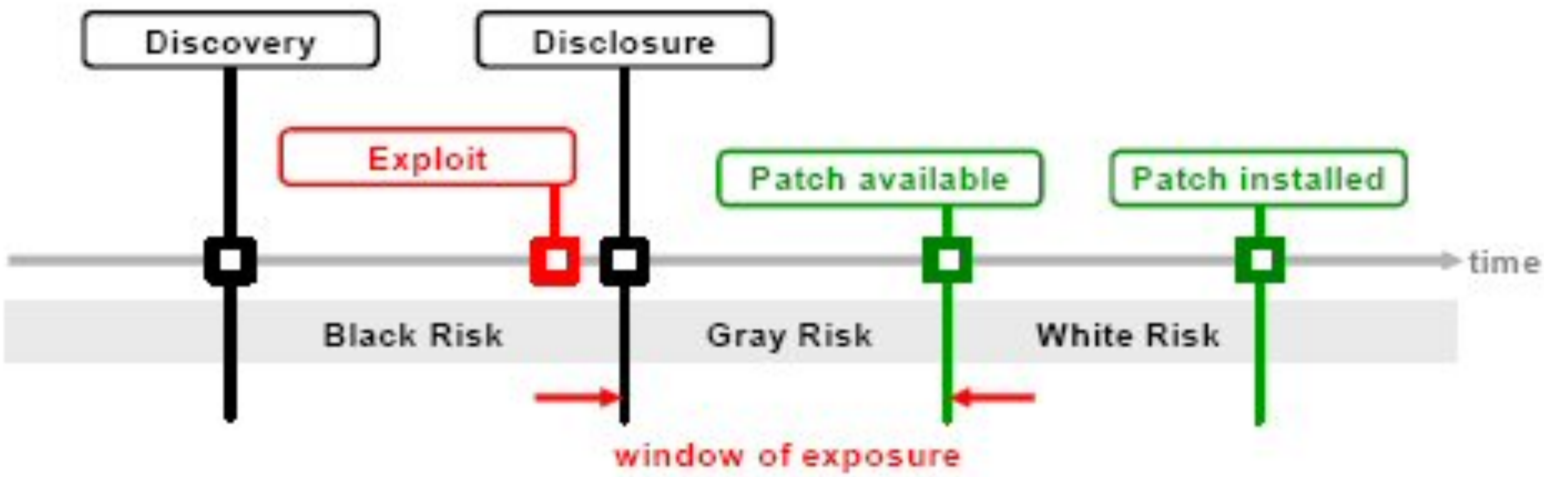
For a single vulnerability, the cumulative risk in a specific system at time t can be expressed as

- probability of the vulnerability being in State 3 at time t
- multiplied by
- the consequence of the vulnerability exploitation.

Joh and Malaiya, "A Framework for Software Security Risk Evaluation using the Vulnerability Lifecycle and CVSS Metrics" 2010

Colorado State University

Zero-day attacks



- A **zero-day attack** is a cyber attack exploiting a vulnerability that has not been disclosed publicly.
- There is almost no defense against a zero-day attack:
  - while the vulnerability remains unknown, the software affected cannot be patched and
  - anti-virus products cannot detect the attack through signature-based scanning
- Notable zero-day attacks include (Bilge, Dumitras)
  - the 2010 Hydraq trojan, also known as the “Aurora” attack
  - the 2010 Stuxnet worm, which combined four zero-day vulnerabilities to target
  - industrial control systems and
  - the 2011 attack against RSA.

Colorado State University
35

Source

- Leyla Bilge and Tudor Dumitraş. Before we knew it: an empirical study of zero-day attacks in the real world. In Proceedings of the 2012 ACM conference on Computer and communications security (CCS '12). ACM, New York, NY, USA, 833-844.

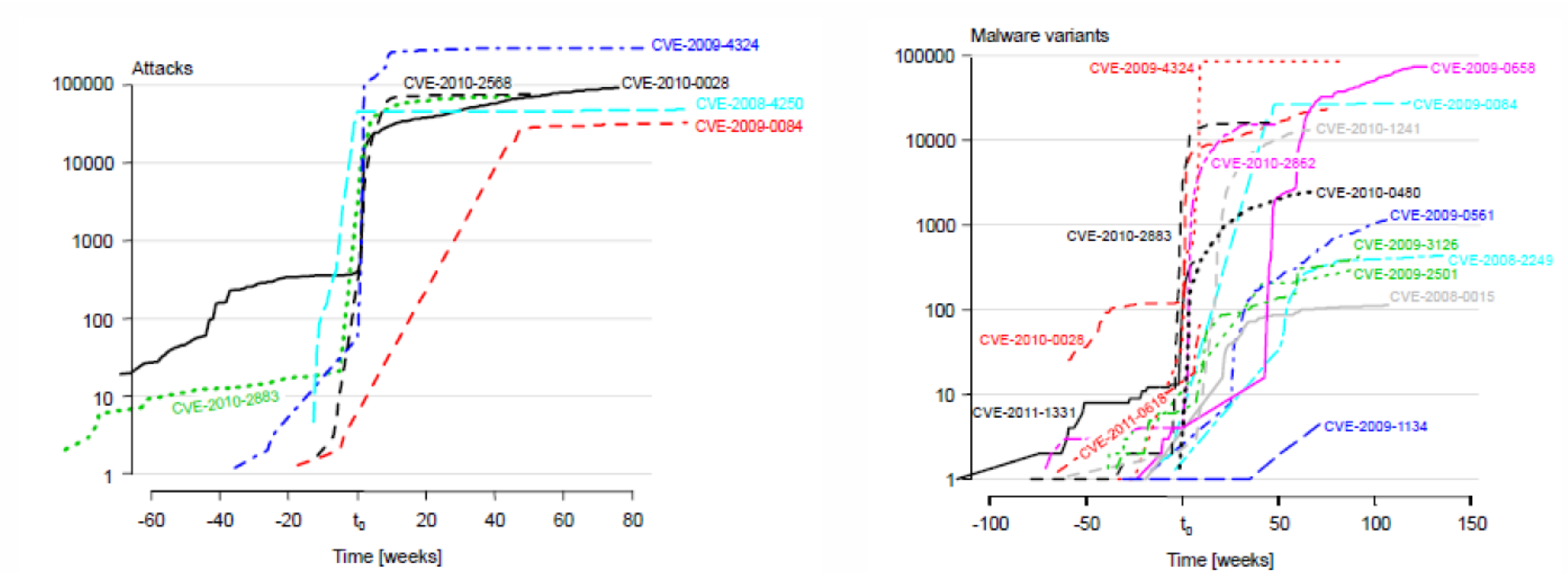
An Empirical Study of Zero-Day Attacks In The Real World

- Field-gathered data for 11 million real hosts around the world.
- Searching this data set for malicious files that exploit known vulnerabilities indicates which files appeared on the Internet before the vulnerabilities were disclosed.
- They identify 18 vulnerabilities exploited before disclosure, of which 11 were not previously known to have been employed in zero-day attacks.
- They also find that a typical zero-day attack lasts 312 days on average
- After vulnerabilities are disclosed publicly, the volume of attacks exploiting them increases by up to 5 orders of magnitude.

Summary of findings

Findings	Implications
Zero-day attacks are more frequent than previously thought: 11 out of 18 vulnerabilities identified were not known zero-day vulnerabilities.	Zero-day attacks are serious threats that may have a significant impact on the organizations affected.
Zero-day attacks last between 19 days and 30 months, with a median of 8 months and an average of approximately 10 months.	Zero-day attacks are not detected in a timely manner using current policies and technologies.
Most zero-day attacks affect few hosts, with the exception of a few high-profile attacks (e.g., Stuxnet).	Most zero-day vulnerabilities are employed in targeted attacks.
58% of the anti-virus signatures are still active at the time of writing.	Data covering 4 years is not sufficient for observing all the phases in the vulnerability lifecycle.
After zero-day vulnerabilities are disclosed, the number of malware variants exploiting them increases 183–85,000 times and the number of attacks increases 2–100,000 times.	The public disclosure of vulnerabilities is followed by an increase of up to five orders of magnitude in the volume of attacks.
Exploits for 42% of all vulnerabilities employed in host-based threats are detected in field data within 30 days after the disclosure date.	Cyber criminals watch closely the disclosure of new vulnerabilities, in order to start exploiting them.

Impact of discosure



(a) Attacks exploiting zero-day vulnerabilities before and after the disclosure (time = t_0).

(b) Malware variants exploiting zero-day vulnerabilities before and after disclosure (time = t_0).

Figure 6: Impact of vulnerability disclosures on the volume of attacks. We utilize logarithmic scales to illustrate an increase of several orders of magnitude after disclosure.

Colorado State University

Time to exploit

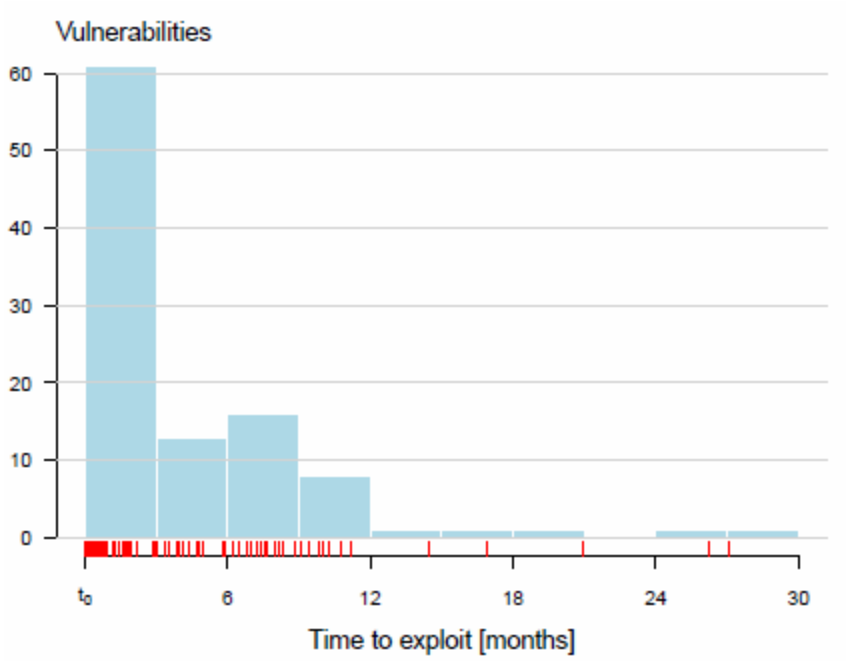
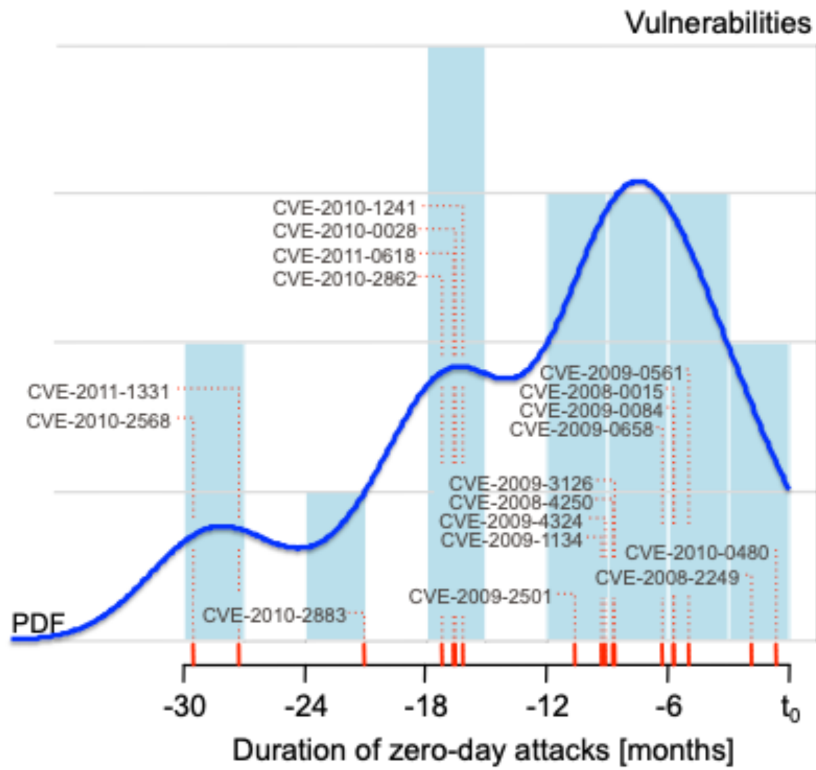


Figure 7: Time before vulnerabilities disclosed between 2008–2011 started being exploited in the field. The histograms group the exploitation lag in 3-month increments, after disclosure, and the red rug indicates the lag for each exploited vulnerability. The zero-day attacks are excluded from this figure.

Duration of zero-day attacks

- The zero-day attacks they identify lasted between
- 19 days (CVE-2010-0480) and
- 30 months (CVE-2010-2568), and
- the average duration of a zero-day attack is 312 days.

Before We Knew It An Empirical Study of Zero-Day Attacks In The Real World



Colorado State University