CSC 591 Systems Attacks and Defenses

Alexandros Kapravelos
akaprav@ncsu.edu

Administration

- Class website
  - https://kapravelos.com/teaching/csc591-s20/schedule/
- Piazza
  - http://piazza.com/ncsu/spring2020/csc591
- Mail to instructor (for private matters)
  - akaprav@ncsu.edu
- Recorded classes
  - https://mediasite.wolfware.ncsu.edu/online/Channel/csc-591-002-sprg-2020

Material

- What material will we be using?
  - Unfortunately, there is no good book on systems security
  - Use the slides that I will post on the web site
  - Related papers/readings and online material (from the syllabus)

Grading

- What are the requirements to get a grade?
  - Two exams (midterm and final) - 30% of grade
  - Homework Assignments & live labs - 60% of grade
  - Participation - 10% of grade
    - Class Participation
    - Quizzes

Topics

Basics
Software Security
Web Security

You need to understand

- Networks and Operating Systems
- Basics of systems theory and implementation
  - E.g., file systems, distributed systems, networking, operating systems, …
- You will build stuff. I expect you to:
  - know how to code (in language of your choice*)
  - I will use mix of pseudocode, Python, Assembly, JavaScript, PHP and C
  - be(come) comfortable with Linux/UNIX

Goals

Learn how an attacker takes control of a system

Learn to defend and avoid common exploits

Learn how to architect secure systems

Assignments

- Individual homework assignments
- These are going to be hard!
- You are going to implement attacks and defenses
- Discovering a vulnerability is a frustrating, but very rewarding in the end!

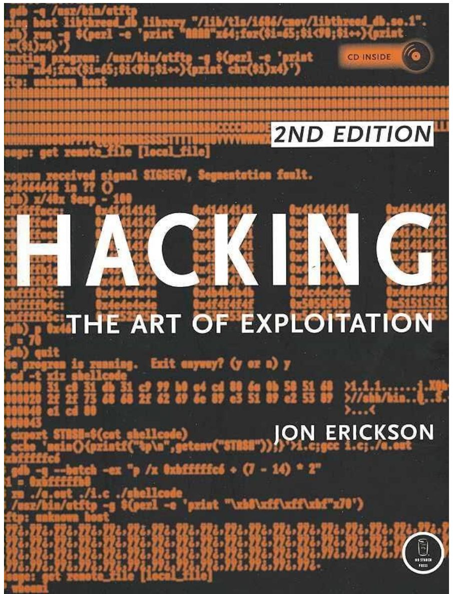
Labs - Flipped classroom

- Some of the lectures are going to be pre-recorded
- You will have to watch the lecture and study **before class**
- During the class we are going to do live exercises of what you’ve learned
- Security in practice

HackPack CTF

- Capture the Flag security competition
- 6 hours live hacking
- We’ll have pizzas & sodas
- **April 17th 1-7pm**
- It will count as one homework assignment
- There will be prizes for top places!

HackPack CTF prizes 2017

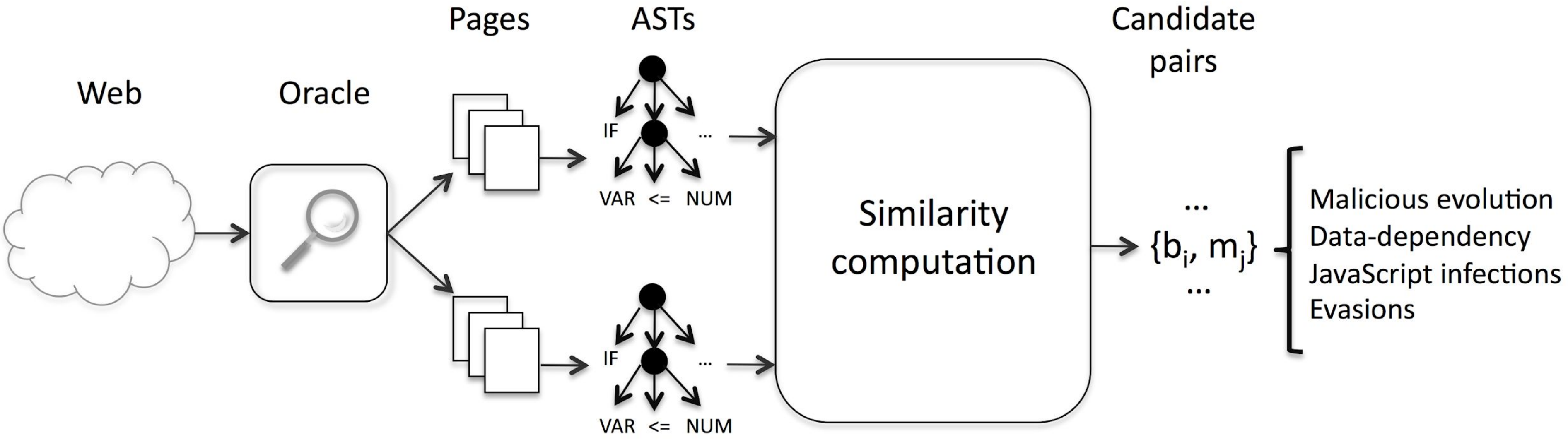


Readings

- There is a large amount of readings in this course covering various topics. These readings are intended to:
  - Support the lectures in the course (provide clarity)
  - Augment the lectures and provide a broader exposure to security topics
- **Students are required to do the reading!**
  - Some of the questions on the exams will be off the reading on topics that were not covered in class

Cheating policy

- Cheating is not allowed
- We run tools
- If you cheat you will probably get caught and get a failing grade in the course
- All academic dishonesty incidents will be reported without exception



Ethics

With great power comes great responsibility

- Topics will cover technologies whose abuse may infringe on the rights of others
- When in doubt, please contact the instructor for advice. Do not undertake any action which could be perceived as technology misuse anywhere and/or under any circumstances unless you have received explicit written permission from the instructor.

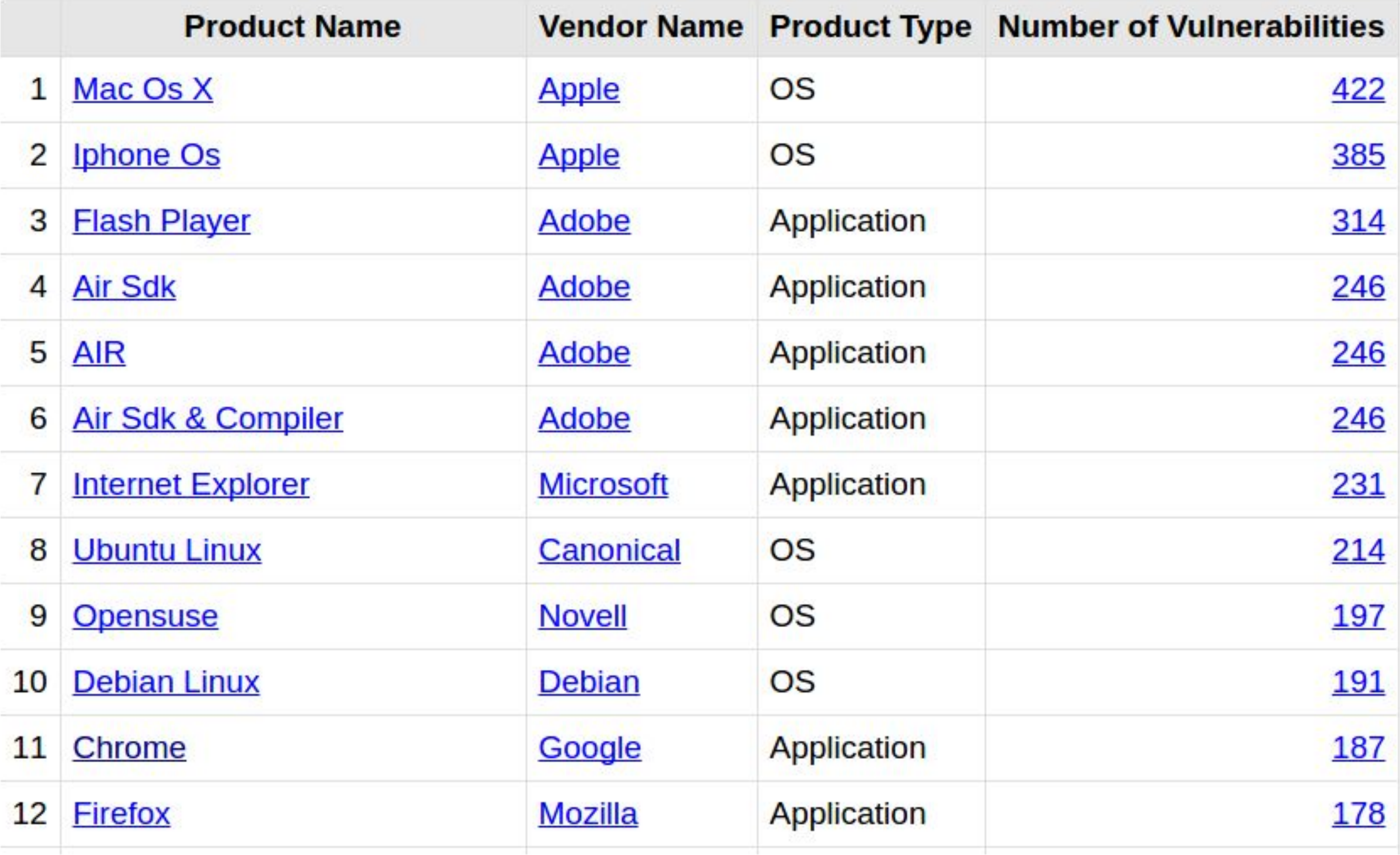
The computer security problem

- Security is everywhere (like the Matrix)
- Developers are not aware of security (we should fix this!)
  - Buggy software
  - Legacy software
  - Social engineering
- Vulnerabilities can be very damaging (and expensive)

Hacking used to be cool

But now everything is done for profit!

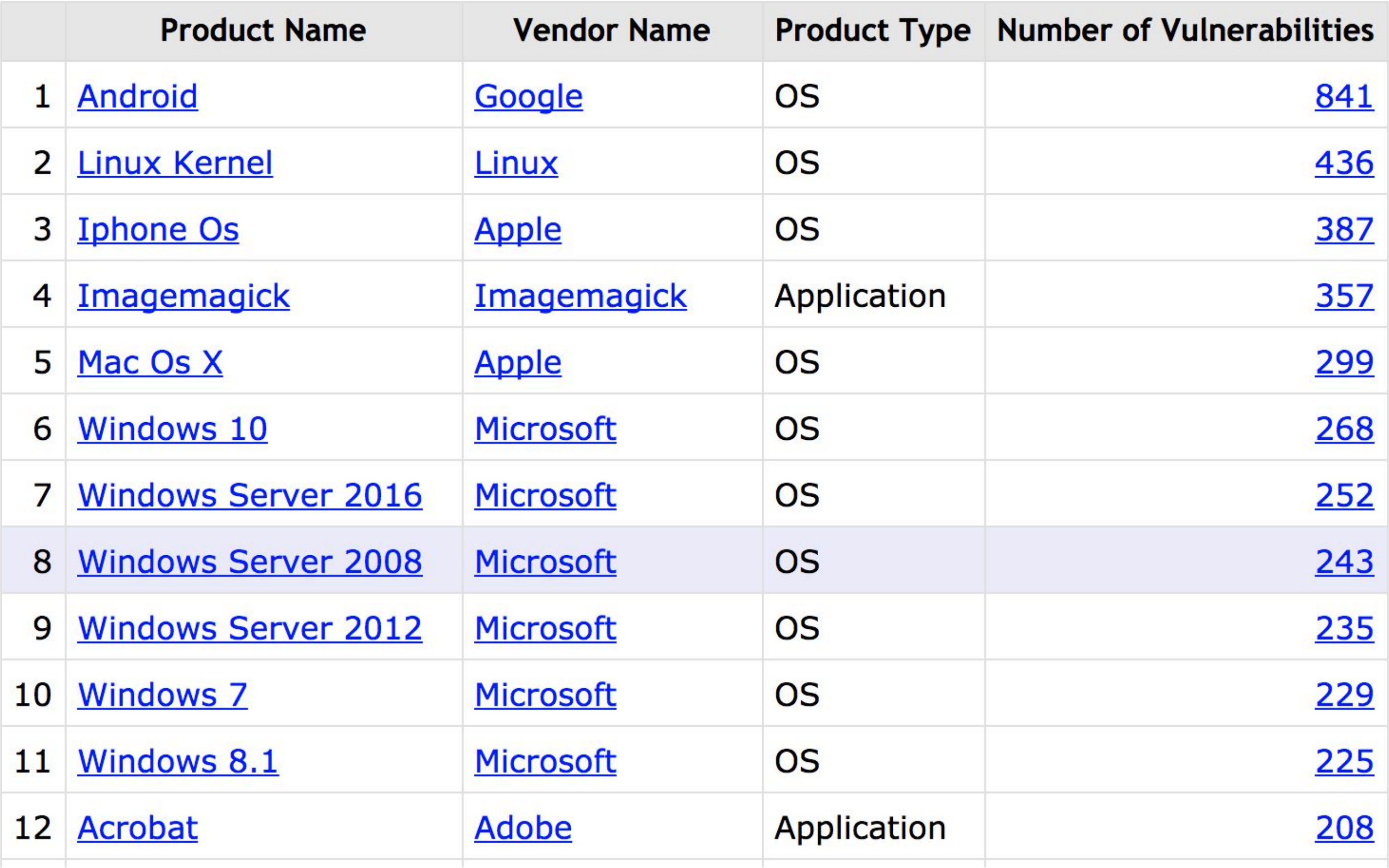
Vulnerabilities per product - 2015



	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Mac Os X	Apple	OS	422
2	Iphone Os	Apple	OS	385
3	Flash Player	Adobe	Application	314
4	Air Sdk	Adobe	Application	246
5	AIR	Adobe	Application	246
6	Air Sdk & Compiler	Adobe	Application	246
7	Internet Explorer	Microsoft	Application	231
8	Ubuntu Linux	Canonical	OS	214
9	Opensuse	Novell	OS	197
10	Debian Linux	Debian	OS	191
11	Chrome	Google	Application	187
12	Firefox	Mozilla	Application	178

Source: https://www.cvedetails.com/top-50-products.php?year=2015

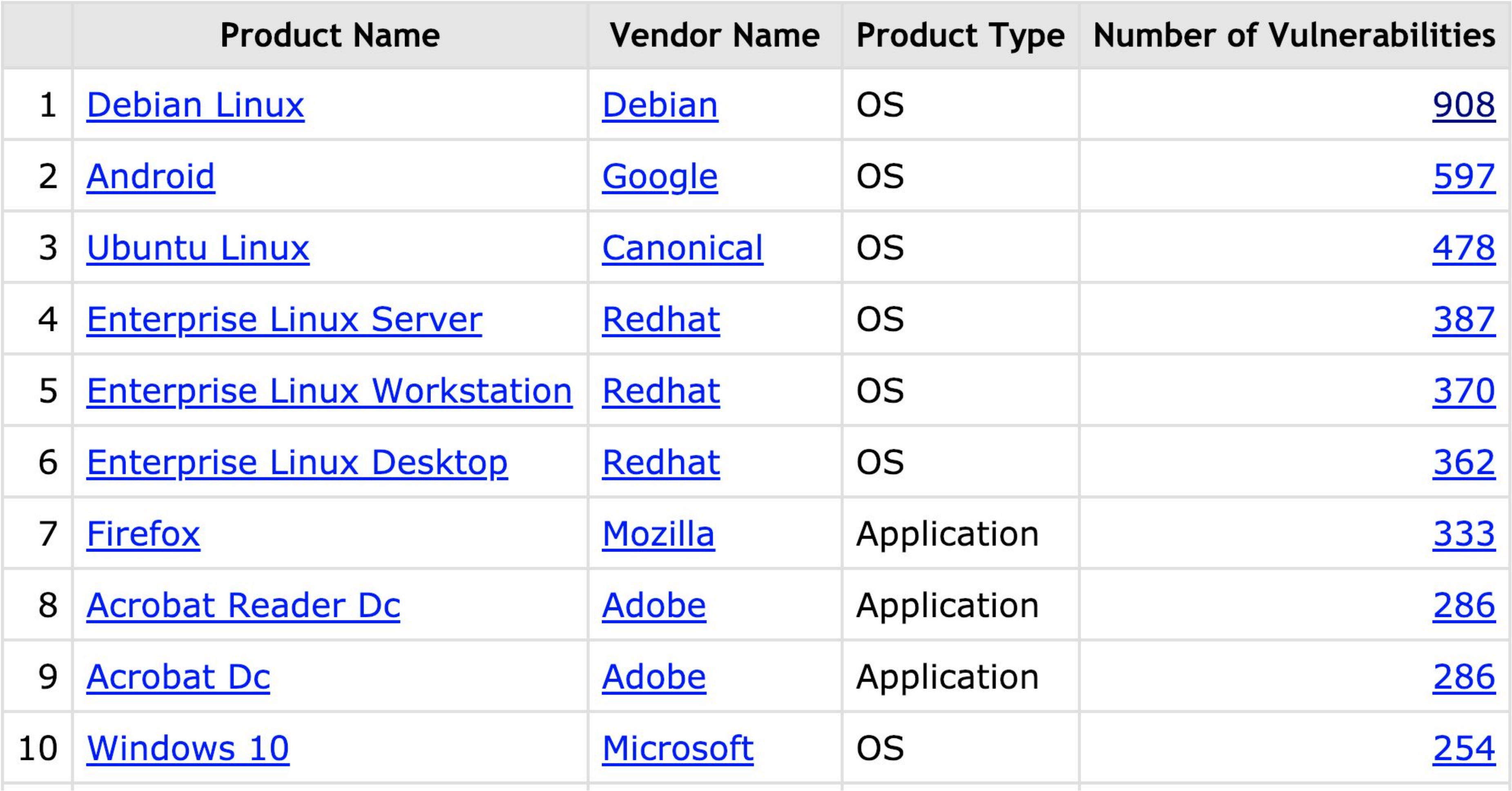
Vulnerabilities per product - 2017



	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Android	Google	OS	841
2	Linux Kernel	Linux	OS	436
3	Iphone Os	Apple	OS	387
4	Imagemagick	Imagemagick	Application	357
5	Mac Os X	Apple	OS	299
6	Windows 10	Microsoft	OS	268
7	Windows Server 2016	Microsoft	OS	252
8	Windows Server 2008	Microsoft	OS	243
9	Windows Server 2012	Microsoft	OS	235
10	Windows 7	Microsoft	OS	229
11	Windows 8.1	Microsoft	OS	225
12	Acrobat	Adobe	Application	208

Source: https://www.cvedetails.com/top-50-products.php?year=2017

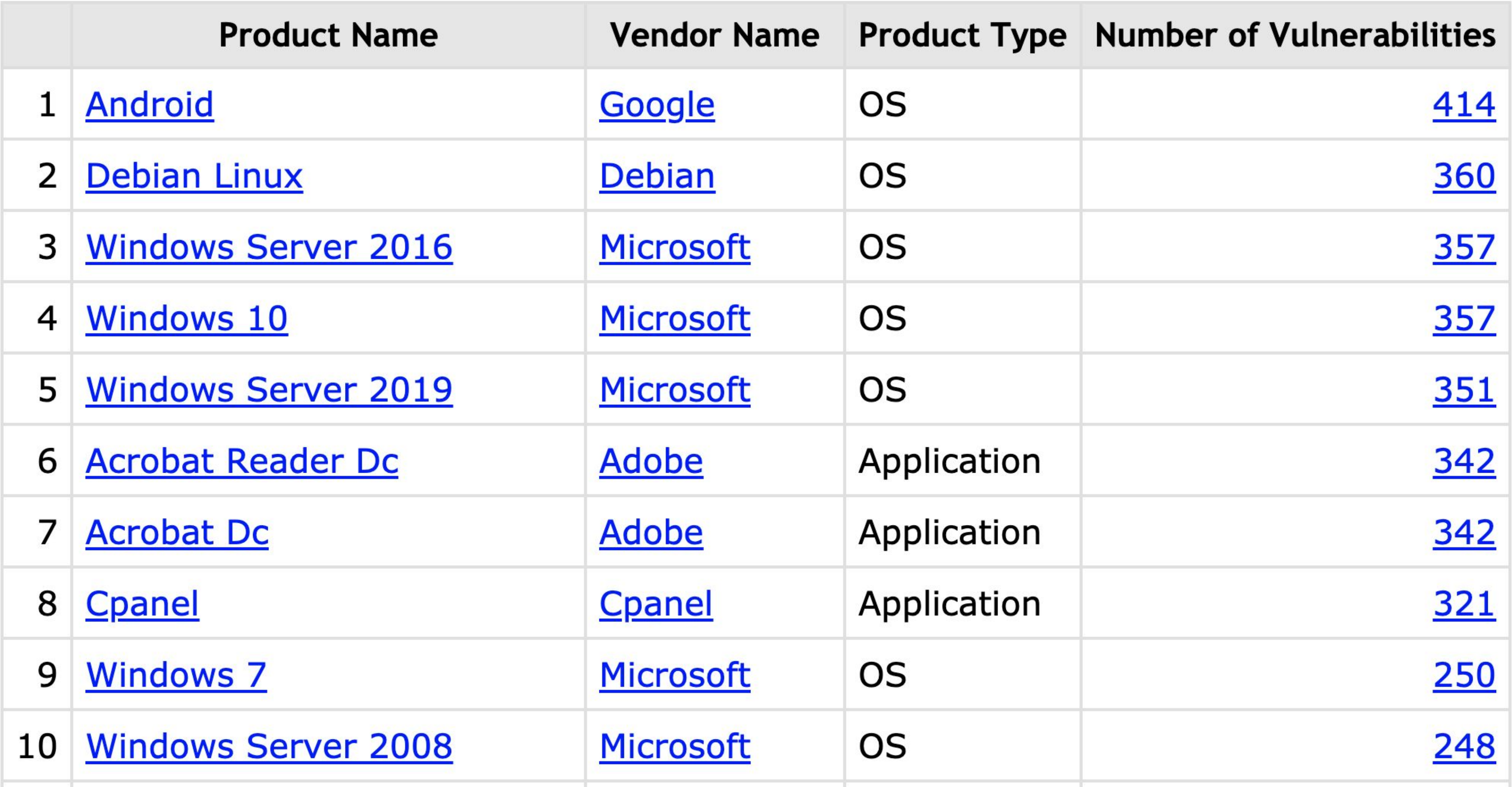
Vulnerabilities per product - 2018



	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Debian Linux	Debian	OS	908
2	Android	Google	OS	597
3	Ubuntu Linux	Canonical	OS	478
4	Enterprise Linux Server	Redhat	OS	387
5	Enterprise Linux Workstation	Redhat	OS	370
6	Enterprise Linux Desktop	Redhat	OS	362
7	Firefox	Mozilla	Application	333
8	Acrobat Reader Dc	Adobe	Application	286
9	Acrobat Dc	Adobe	Application	286
10	Windows 10	Microsoft	OS	254

Source: https://www.cvedetails.com/top-50-products.php?year=2018

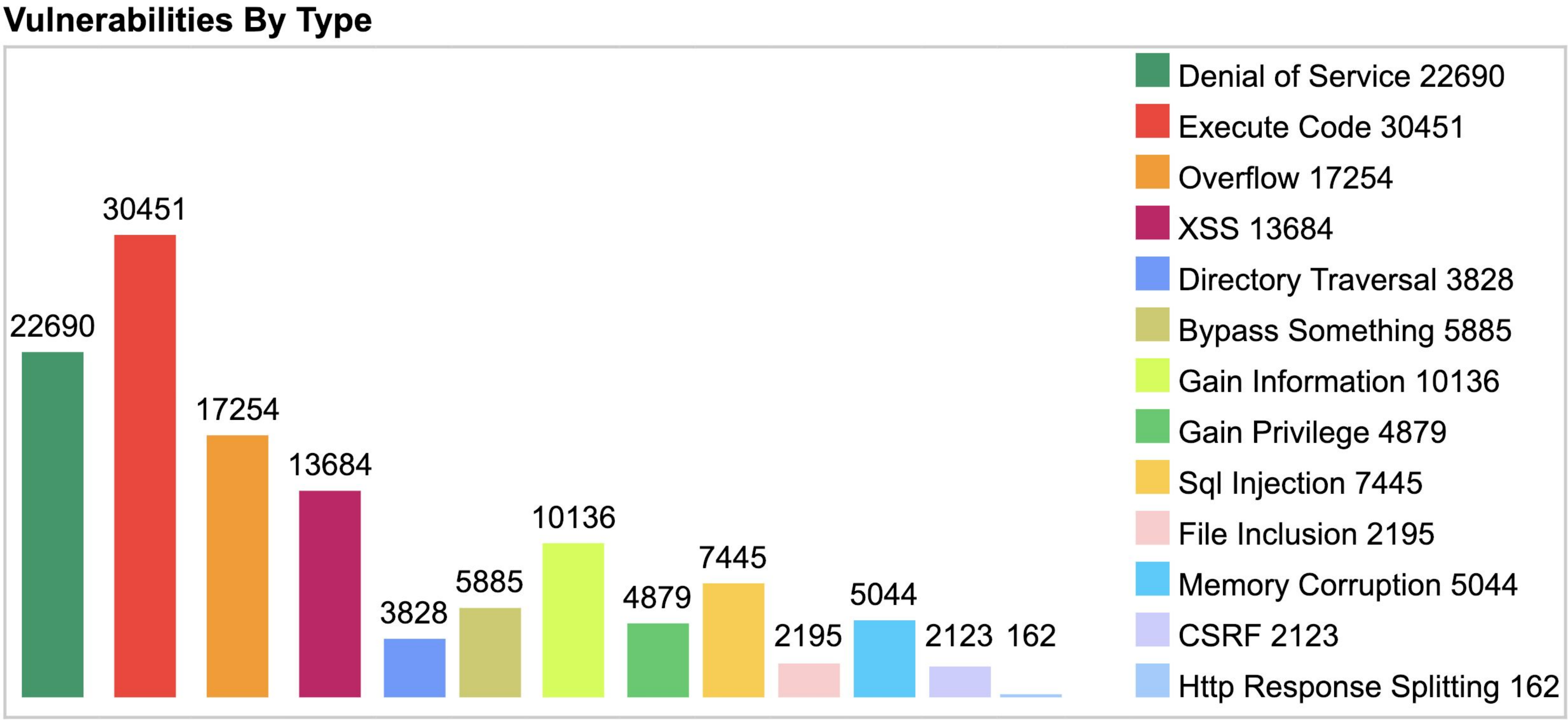
Vulnerabilities per product - 2019



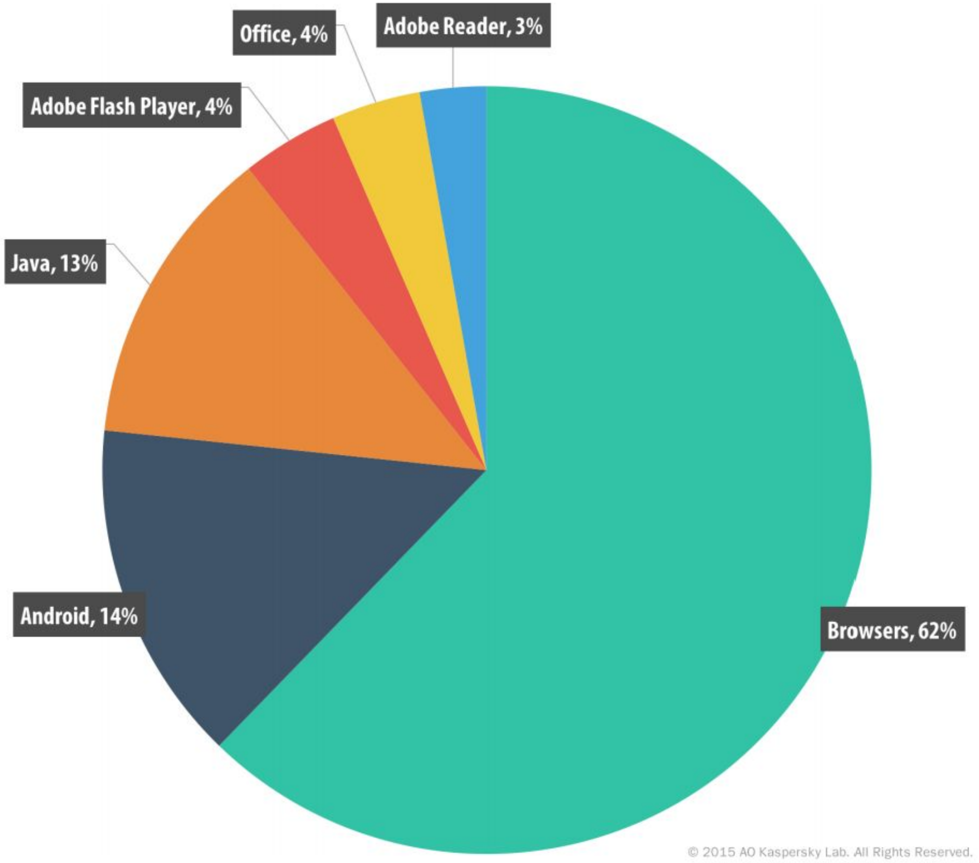
	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Android	Google	OS	414
2	Debian Linux	Debian	OS	360
3	Windows Server 2016	Microsoft	OS	357
4	Windows 10	Microsoft	OS	357
5	Windows Server 2019	Microsoft	OS	351
6	Acrobat Reader Dc	Adobe	Application	342
7	Acrobat Dc	Adobe	Application	342
8	Cpanel	Cpanel	Application	321
9	Windows 7	Microsoft	OS	250
10	Windows Server 2008	Microsoft	OS	248

Source: https://www.cvedetails.com/top-50-products.php?year=2019

Vulnerabilities per type - 1999-2018

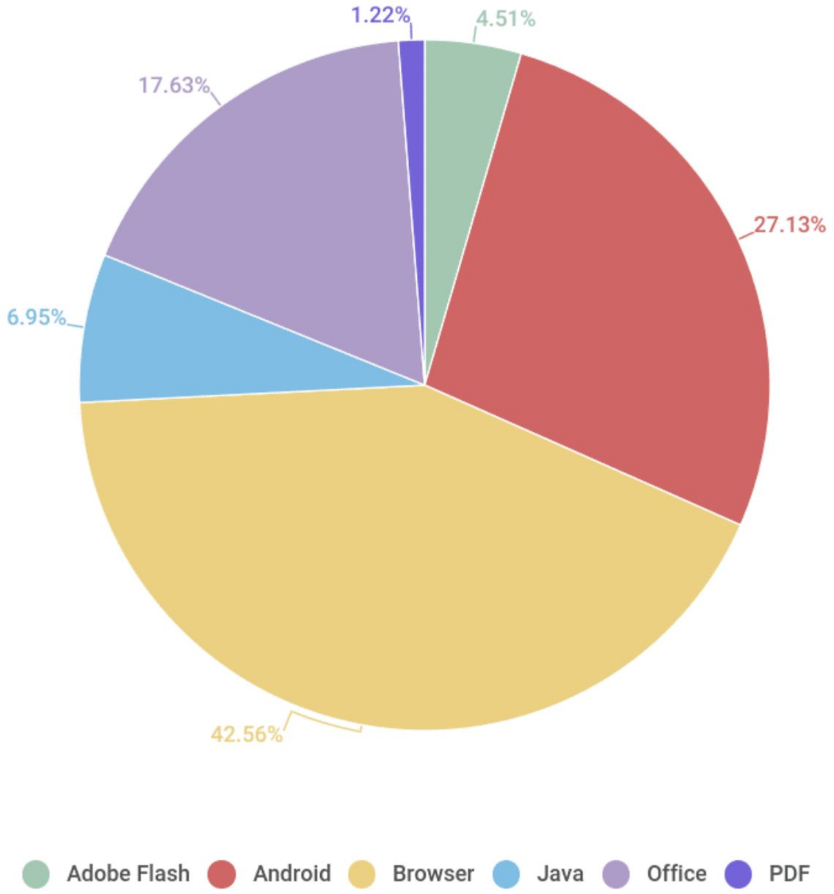


Distribution of exploits per application 2015



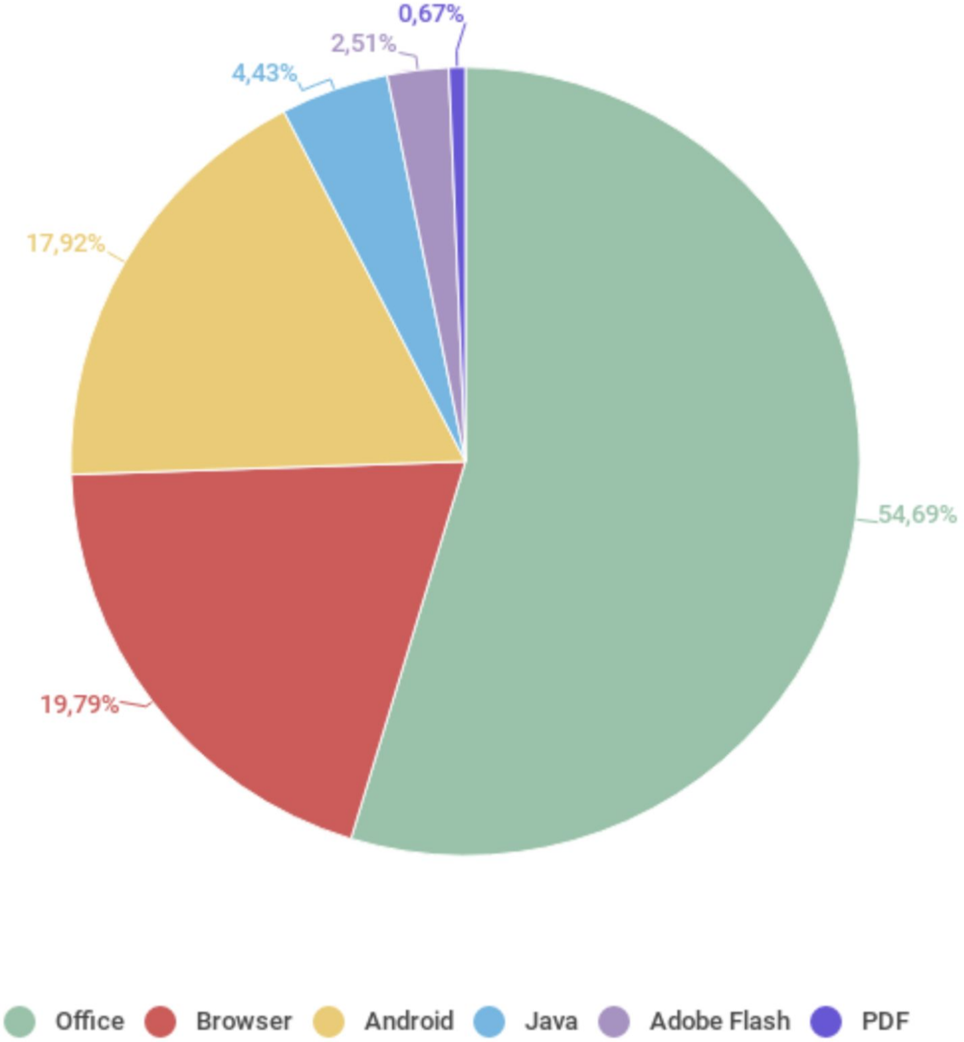
Source: Kaspersky Security Bulletin 2015

Distribution of exploits per application 2017



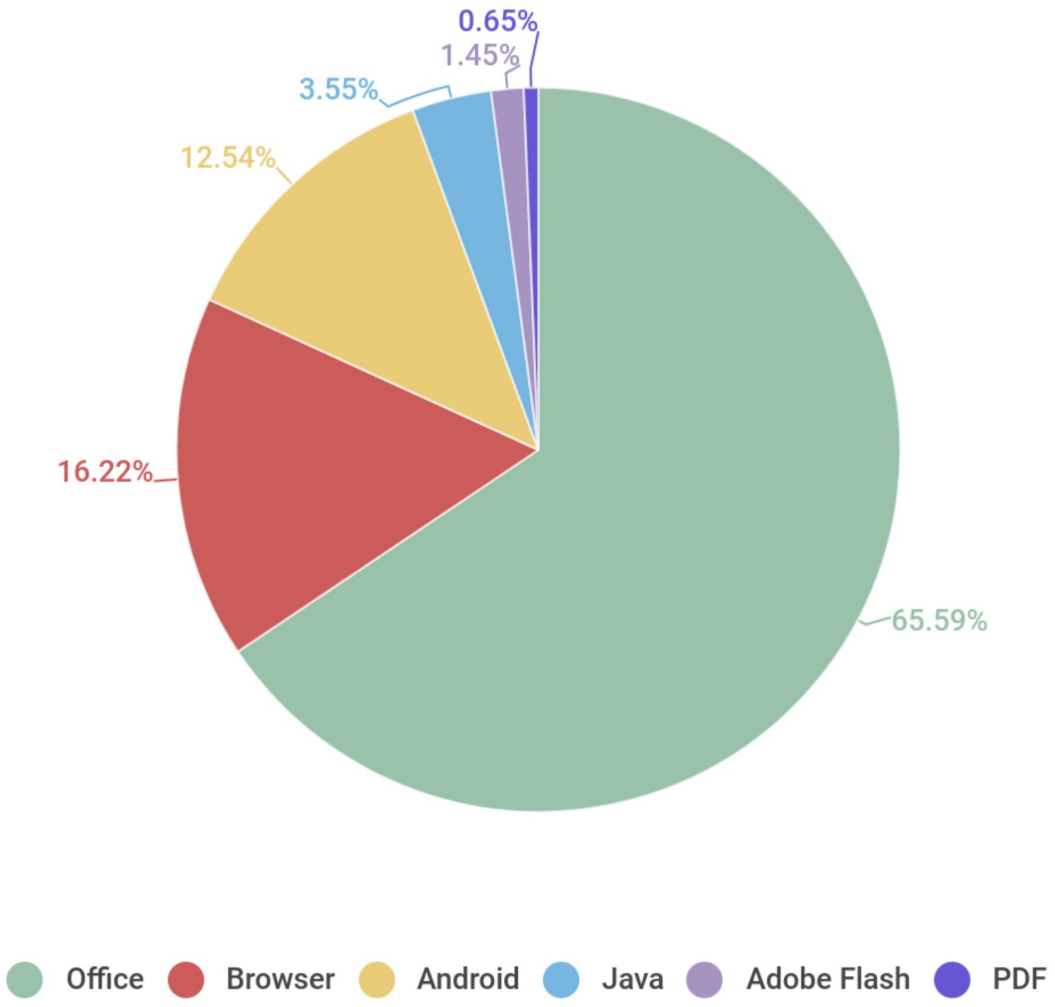
Source: Kaspersky Security Bulletin 2017

Distribution of exploits per application 2018



Source: Kaspersky Security Bulletin 2018

Distribution of exploits per application 2019



Source: Kaspersky Security Bulletin 2019

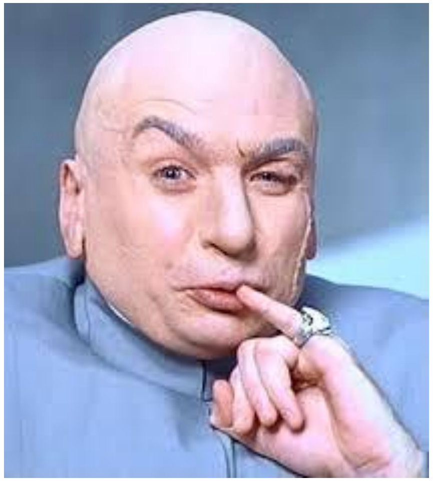
Bug bounty programs

- Companies will pay you money to report vulnerabilities
- Certain conditions and rules per program
  - No Denial-of-service attacks
  - Spam
  - … (depends on the program)

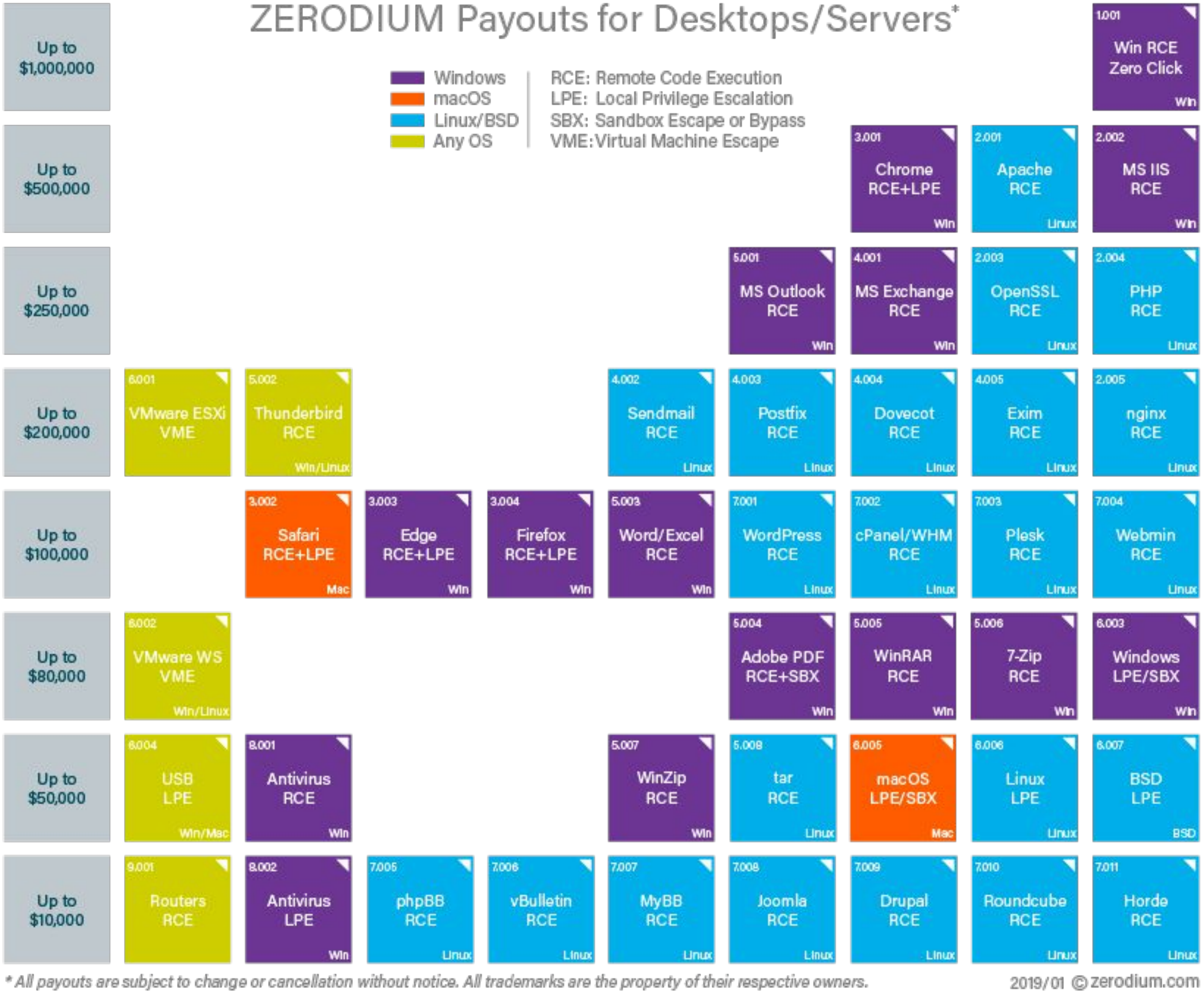
Black market for exploits

Last iOS exploit was sold for

1 million dollars



NC STATE UNIVERSITY



ZERODIUM Payouts for Desktops/Servers*

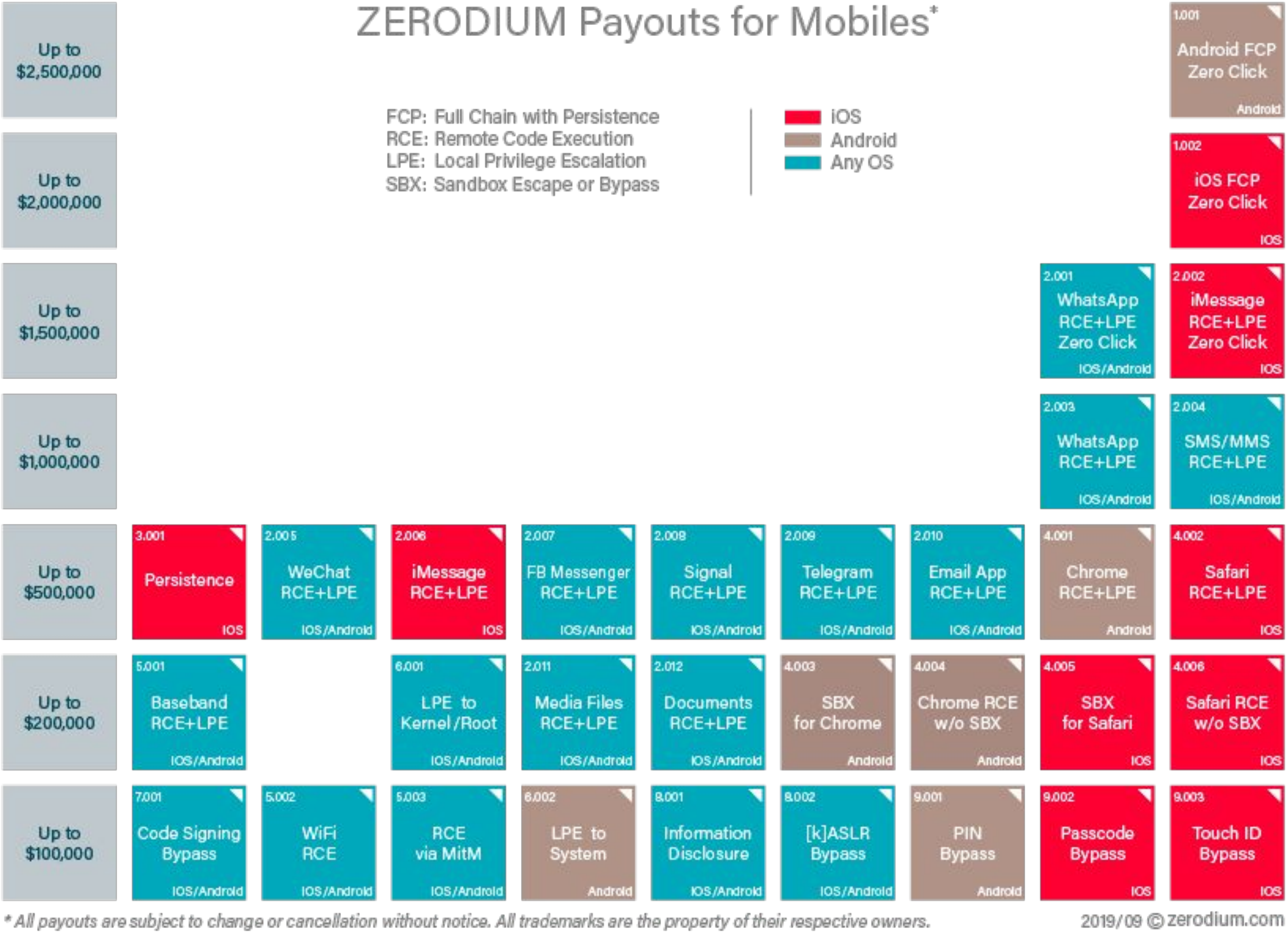
🟪 Windows
🟧 macOS
🟦 Linux/BSD
🟨 Any OS

RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass
VME: Virtual Machine Escape

Up to $1,000,000									1.001 Win RCE Zero Click Win
Up to $500,000							3.001 Chrome RCE+LPE Win	2.001 Apache RCE Linux	2.002 MS IIS RCE Win
Up to $250,000						5.001 MS Outlook RCE Win	4.001 MS Exchange RCE Win	2.003 OpenSSL RCE Linux	2.004 PHP RCE Linux
Up to $200,000	6.001 VMware ESXi VME	5.002 Thunderbird RCE Win/Linux			4.002 Sendmail RCE Linux	4.003 Postfix RCE Linux	4.004 Dovecot RCE Linux	4.005 Exim RCE Linux	2.005 nginx RCE Linux
Up to $100,000		3.002 Safari RCE+LPE Mac	3.003 Edge RCE+LPE Win	3.004 Firefox RCE+LPE Win	5.003 Word/Excel RCE Win	7.001 WordPress RCE Linux	7.002 cPanel/WHM RCE Linux	7.003 Plesk RCE Linux	7.004 Webmin RCE Linux
Up to $80,000	6.002 VMware WS VME Win/Linux					5.004 Adobe PDF RCE+SBX Win	5.005 WinRAR RCE Win	5.006 7-Zip RCE Win	6.003 Windows LPE/SBX Win
Up to $50,000	6.004 USB LPE Win/Mac	8.001 Antivirus RCE Win			5.007 WinZip RCE Win	5.008 tar RCE Linux	6.005 macOS LPE/SBX Mac	6.006 Linux LPE Linux	6.007 BSD LPE BSD
Up to $10,000	9.001 Routers RCE	8.002 Antivirus LPE Win	7.005 phpBB RCE Linux	7.006 vBulletin RCE Linux	7.007 MyBB RCE Linux	7.008 Joomla RCE Linux	7.009 Drupal RCE Linux	7.010 Roundcube RCE Linux	7.011 Horde RCE Linux

All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.

2019/01 © zerodium.com



ZERODIUM Payouts for Mobiles*

FCP: Full Chain with Persistence
RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass

🟥 iOS
▨ Android
🟦 Any OS

Up to $2,500,000									1.001 Android FCP Zero Click Android
Up to $2,000,000									1.002 iOS FCP Zero Click iOS
Up to $1,500,000								2.001 WhatsApp RCE+LPE Zero Click iOS/Android	2.002 iMessage RCE+LPE Zero Click iOS
Up to $1,000,000								2.003 WhatsApp RCE+LPE iOS/Android	2.004 SMS/MMS RCE+LPE iOS/Android
Up to $500,000	3.001 Persistence iOS	2.005 WeChat RCE+LPE iOS/Android	2.006 iMessage RCE+LPE iOS	2.007 FB Messenger RCE+LPE iOS/Android	2.008 Signal RCE+LPE iOS/Android	2.009 Telegram RCE+LPE iOS/Android	2.010 Email App RCE+LPE iOS/Android	4.001 Chrome RCE+LPE Android	4.002 Safari RCE+LPE iOS
Up to $200,000	5.001 Baseband RCE+LPE iOS/Android		6.001 LPE to Kernel/Root iOS/Android	2.011 Media Files RCE+LPE iOS/Android	2.012 Documents RCE+LPE iOS/Android	4.003 SBX for Chrome Android	4.004 Chrome RCE w/o SBX Android	4.005 SBX for Safari iOS	4.006 Safari RCE w/o SBX iOS
Up to $100,000	7.001 Code Signing Bypass iOS/Android	5.002 WiFi RCE iOS/Android	5.003 RCE via MitM iOS/Android	6.002 LPE to System Android	8.001 Information Disclosure iOS/Android	8.002 [k]ASLR Bypass iOS/Android	9.001 PIN Bypass Android	9.002 Passcode Bypass iOS	9.003 Touch ID Bypass iOS

*All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.

2019/09 © zerodium.com

Exploits for modern software are extremely difficult to write!

Chrome exploit

- Bug 1: run Native Client from any website
- Bug 2: integer underflow bug in the GPU command decoding -> ROP chain in GPU process
- Bug 3: impersonate the renderer from the GPU in the IPC channel
- Bug 4: allowed an unprivileged renderer to trigger a navigation to one of the privileged renderers -> launch the extension manager

Source: https://blog.chromium.org/2012/05/tale-of-two-pwnies-part-1.html

Chrome exploit

- Bug 5: specify a load path for an extension
- Bug 6: failure to prompt for confirmation prior to installing an unpacked NPAPI plug-in extension

Result: install and run a custom NPAPI plugin that executes outside the sandbox at full user privilege

Next class

Refresh your assembly skills!

Your Security Zen

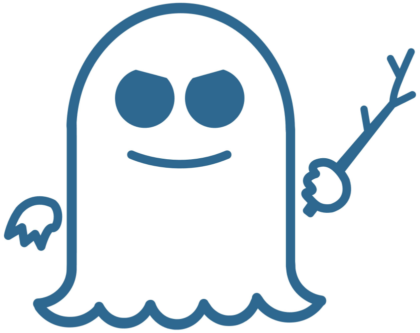
At the end of every lecture we will have a short discussion on a recent security topic

Use piazza or hackpack slack #random channel if you see in the news interesting security incidents!

Here’s one from a previous year

⬇

Your Security Zen



Meltdown and Spectre

two major security flaws in the microprocessors inside nearly all of the world’s computers (Intel, AMD, ARM)

Spectre: no easy fix, we have to redesign processors

Meltdown: 30% slow down

There are proof of concepts in the wild that can read host kernel memory from inside a KVM guest

Sources: https://googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html, https://meltdownattack.com/