

<https://crypto.stanford.edu/cs155>



CS155

Computer Security

Course overview

Acknowledgments: Lecture slides are from the Computer Security course taught by Dan Boneh at Stanford University. When slides are obtained from other sources, a reference will be noted on the bottom of that slide. A full list of references is provided on the last slide.

The computer security problem

- Lots of buggy software
- Social engineering is very effective
- Money can be made from finding and exploiting vulns.

1. Marketplace for vulnerabilities
2. Marketplace for owned machines (PPI)
3. Many methods to profit from owned machines

current state of computer security

Lots of vulnerability disclosures (2015)

	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Mac Os X	Apple	OS	385
2	Iphone Os	Apple	OS	376
3	Flash Player	Adobe	Application	313
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	Chrome	Google	Application	187
9	Firefox	Mozilla	Application	178
10	Windows Server 2012	Microsoft	OS	155
11	Ubuntu Linux	Canonical	OS	152
12	Windows 8.1	Microsoft	OS	151

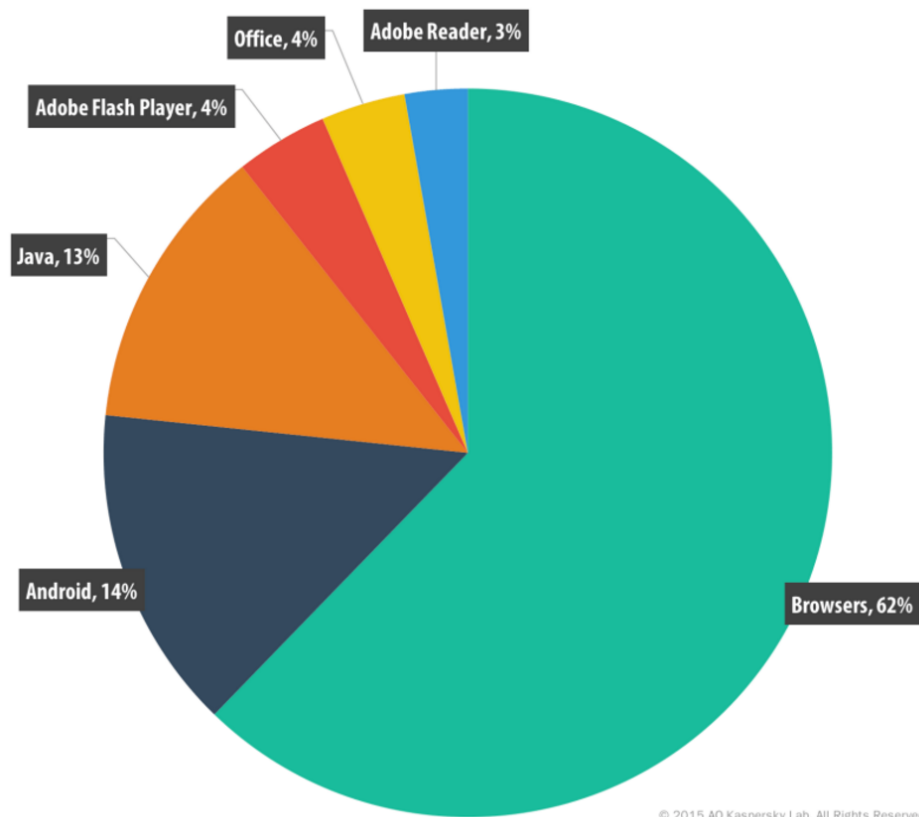
source: www.cvedetails.com/top-50-products.php?year=2015

Lots of vulnerability disclosures (2018)

	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Debian Linux	Debian	OS	1130
2	Ubuntu Linux	Canonical	OS	652
3	Android	Google	OS	611
4	Enterprise Linux Server	Redhat	OS	479
5	Enterprise Linux Workstation	Redhat	OS	462
6	Enterprise Linux Desktop	Redhat	OS	453
7	Firefox	Mozilla	Application	333
8	Acrobat Reader Dc	Adobe	Application	299
9	Acrobat Dc	Adobe	Application	299
10	Windows 10	Microsoft	OS	256
11	Sd 212 Firmware	Qualcomm	OS	253
12	Sd 210 Firmware	Qualcomm	OS	253

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

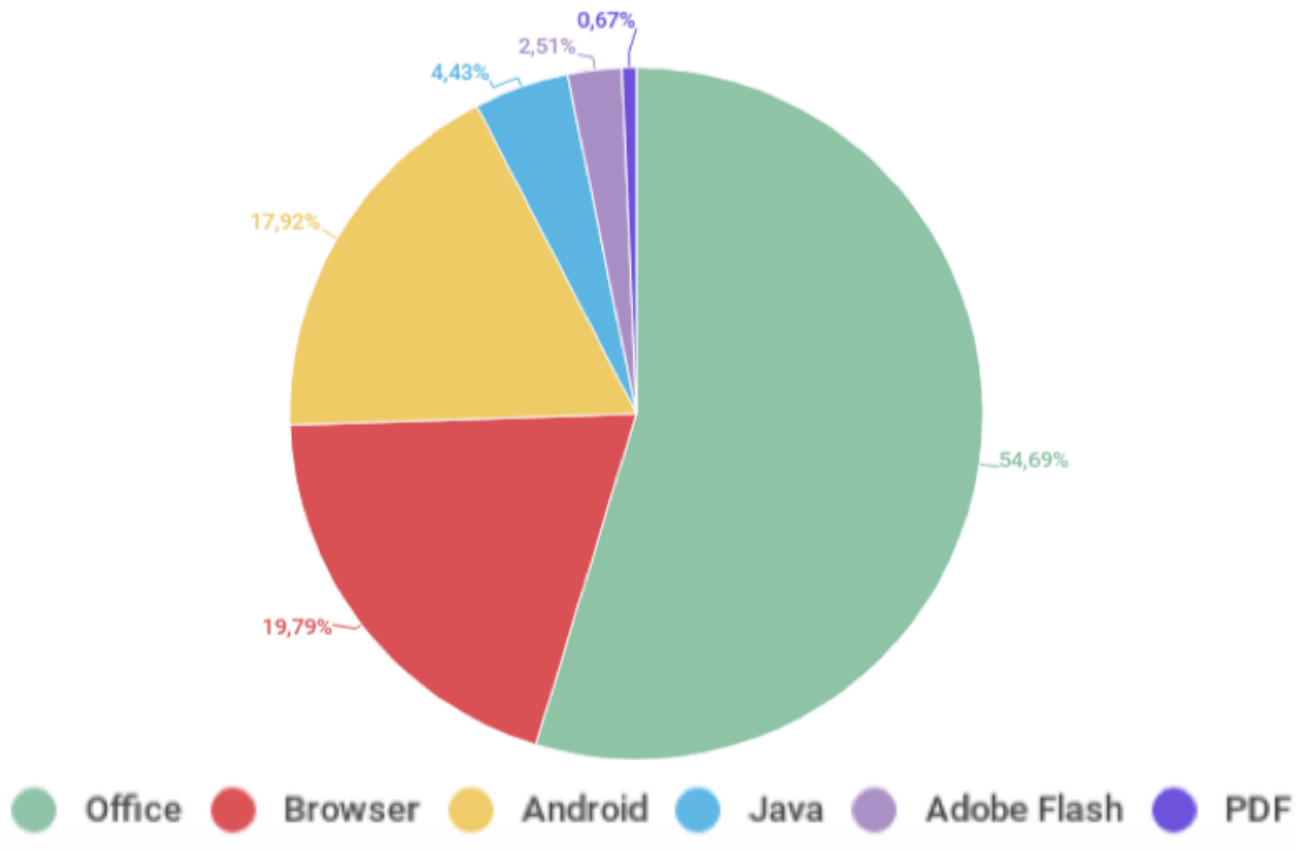
Vulnerable applications being exploited



© 2015 AO Kaspersky Lab. All Rights Reserved.

Source: Kaspersky Security Bulletin 2015

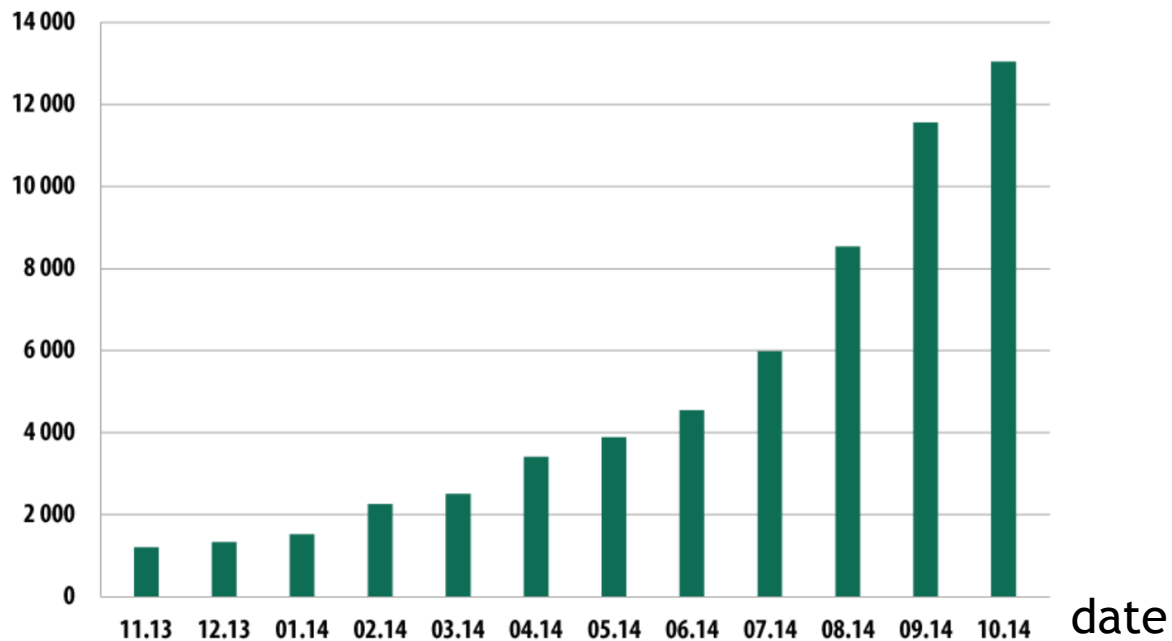
Vulnerable applications being exploited



Source: Kaspersky Security Bulletin 2018

Mobile malware

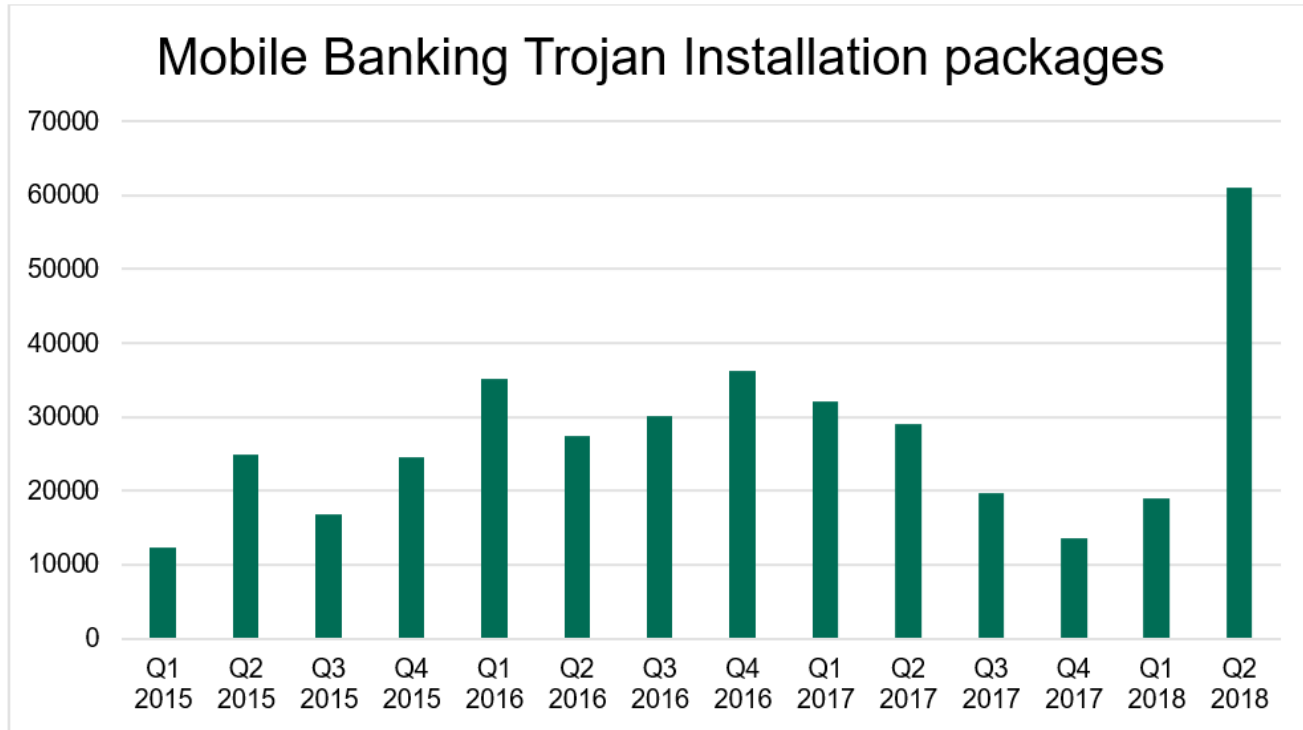
(Nov. 2013 - Oct. 2014)



The rise of mobile banking Trojans

(Kaspersky Security Bulletin 2014)

Mobile malware



The rise of mobile banking Trojans

(Kaspersky Security Bulletin 2018)



Introduction

Sample attacks

Why own machines:

1. IP address and bandwidth stealing

Attacker's goal: look like a random Internet user

Use the IP address of infected machine or phone for:

- **Spam** (e.g. the storm botnet)

Spamalytics: 1:12M pharma spams leads to purchase

1:260K greeting card spams leads to infection

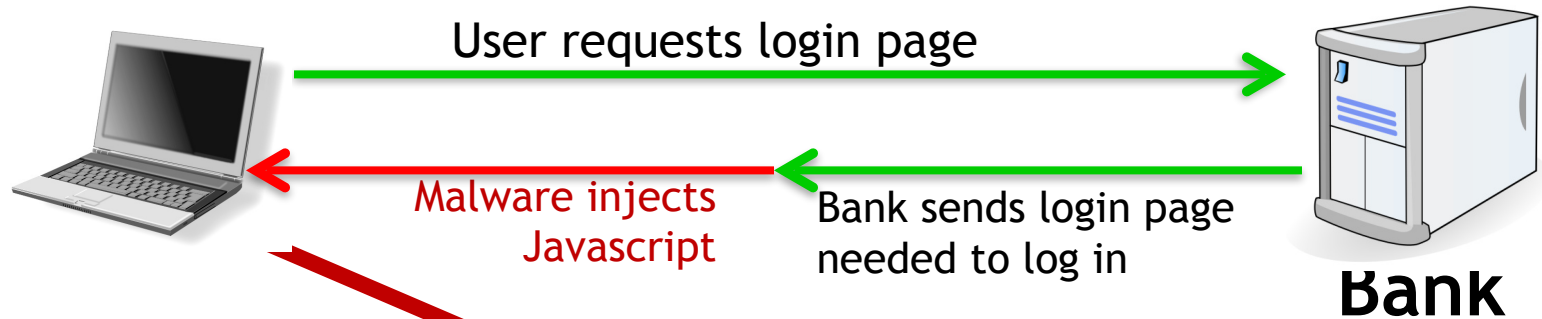
- **Denial of Service: Services:** 1 hour (20\$), 24 hours (100\$)
- **Click fraud** (e.g. Clickbot.a)

Why own machines:

2. Steal user credentials and inject ads

keylog for banking passwords, web passwords, gaming pwds.

Example: SilentBanker (and many like it)



When user submits information, also sent to attacker

Man-in-the-Browser (MITB)



Similar mechanism used by Zeus botnet

Lots of financial malware

1 Trojan-Downloader.Win32.Upatre

2 Trojan-Spy.Win32.Zbot

3 Trojan-Banker.Win32.ChePro

4 Trojan-Banker.Win32.Shiotob

5 Trojan-Banker.Win32.Banbra

6 Trojan-Banker.Win32.Caphaw

7 Trojan-Banker.AndroidOS.Faketoken

8 Trojan-Banker.AndroidOS.Marcher

9 Trojan-Banker.Win32.Tinba

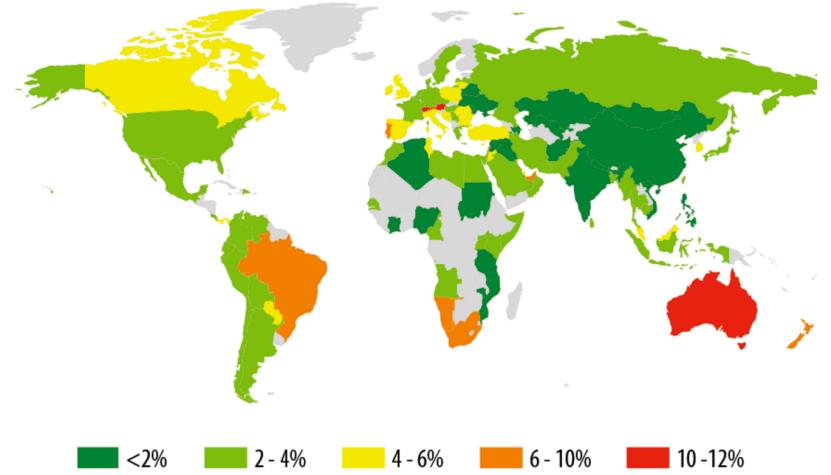
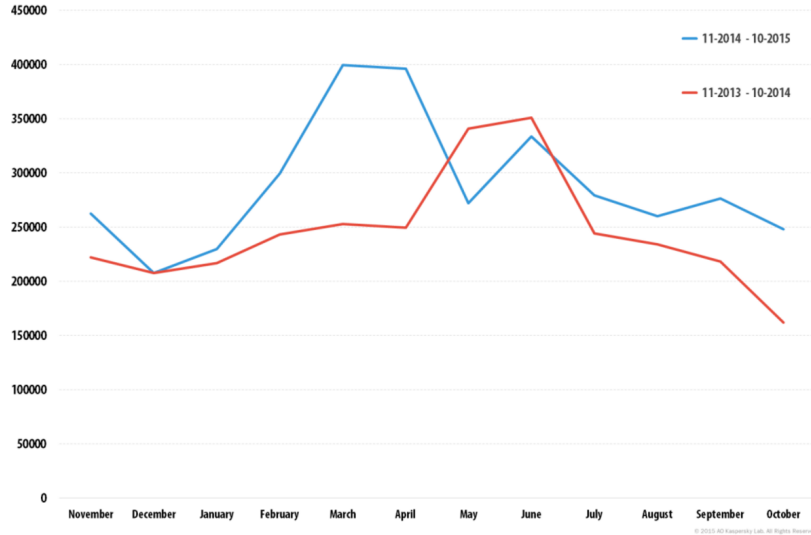
10 Trojan-Banker.JS.Agent

- size: 3.5 KB
- spread via email attachments
- also found on home routers

Lots of financial malware

	Name	%*
1	Trojan.Win32. Zbot	26.3
2	Trojan.Win32. Nymaim	19.8
3	Backdoor.Win32. SpyEye	14.7
4	Backdoor.Win32. Caphaw	5.2
5	Trojan-Banker.Win32.RTM	5.2
6	Backdoor.Win32. Emotet	4.9
7	Trojan.Win32. Neurevt	3.9
8	Trojan-Banker.Win32.Tinba	1.9
9	Trojan.Win32. Gozi	1.8
10	Trojan-Banker.Win32.Trickster	1.5

Users attacked: stats

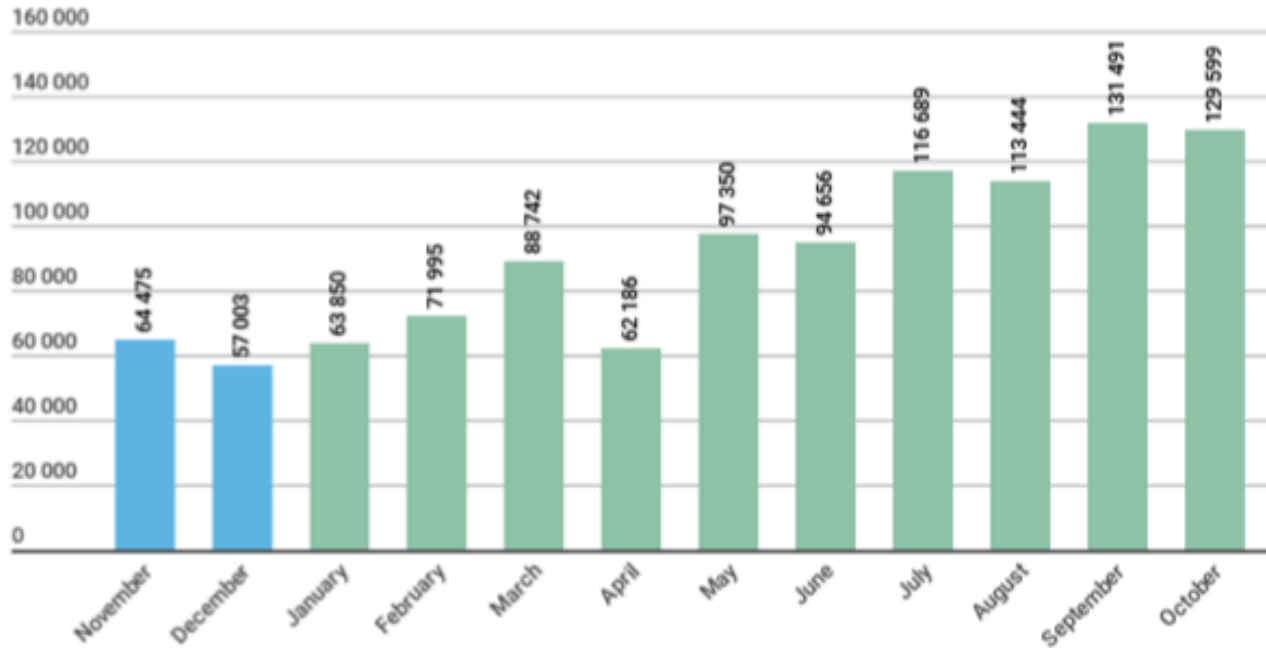


© 2015 AO Kaspersky Lab. All Rights Reserved.

≈ 300,000 users worldwide

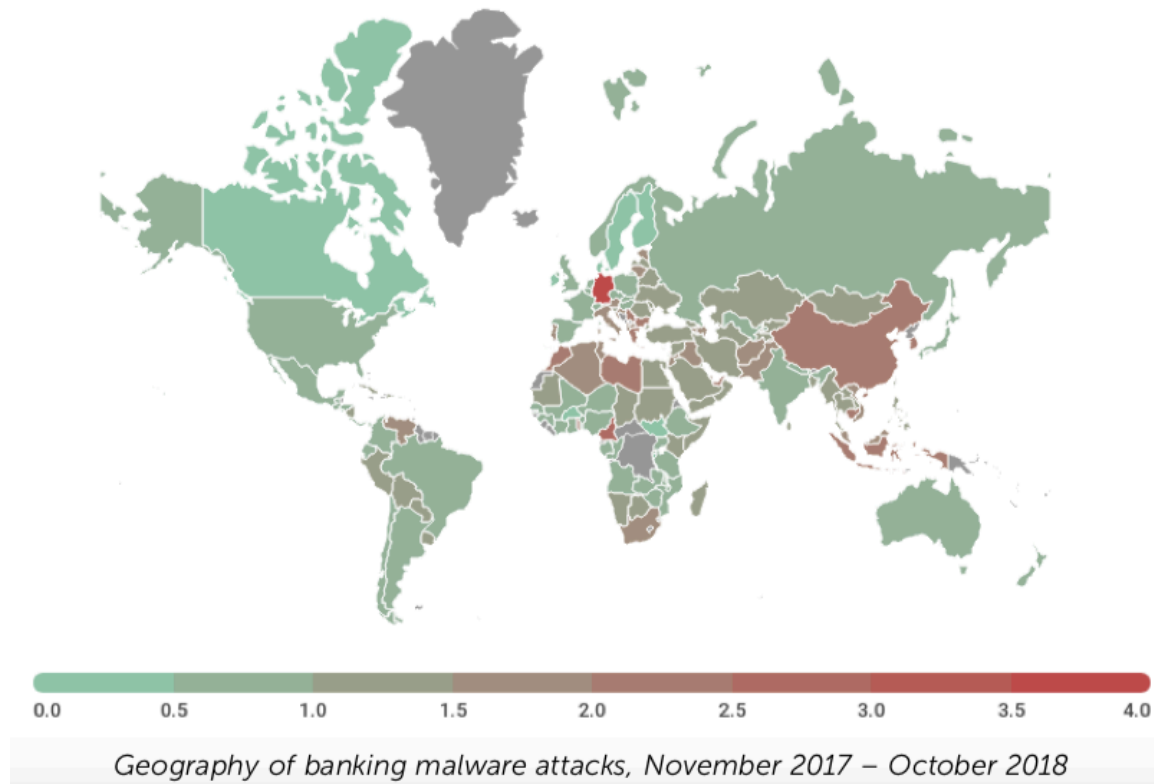
A worldwide problem

Users attacked: stats



*Number of unique users attacked by banking malware,
November 2017 – October 2018*

Users attacked: stats



A worldwide problem

Why own machines:

3. Ransomware

1	Trojan-Ransom.HTML.Agent
2	Trojan-Ransom.JS.Blocker
3	Trojan-Ransom.JS.InstallExtension
4	Trojan-Ransom.NSIS.Onion
5	Trojan-Ransom.Win32.Cryakl
6	Trojan-Ransom.Win32.Cryptodef
7	Trojan-Ransom.Win32.Snocry
8	Trojan-Ransom.BAT.Scatter
9	Trojan-Ransom.Win32.Crypmo
10	Trojan-Ransom.Win32.Shade

CryptoWall (2014-)

- targets Windows
- spread by spam emails

≈ 200,000 machines in 2015

A worldwide problem.

Why own machines: 3. Ransomware

	Name	Verdict	%*
1	WannaCry	Trojan-Ransom.Win32.Wanna	29.32
2	(generic verdict)	Trojan-Ransom.Win32.Phny	11.43
3	GandCrab	Trojan-Ransom.Win32.GandCrypt	6.67
4	Cryakl	Trojan-Ransom.Win32.Cryakl	4.59
5	PolyRansom/VirLock	Virus.Win32.PolyRansom	2.86
6	(generic verdict)	Trojan-Ransom.Win32.Gen	2.40
7	Shade	Trojan-Ransom.Win32.Shade	2.29
8	Cerber	Trojan-Ransom.Win32.Zerber	2.20
9	Purgen/GlobelImposter	Trojan-Ransom.Win32.Purgen	1.82
10	Crysis/Dharma	Trojan-Ransom.Win32.Crusis	1.72

≈ 765,538
machines in
2018

A worldwide
problem.

Why own machines:

4. Spread to isolated systems

Example: **Stuxnet**

Windows infection ⇒

Siemens PCS 7 SCADA control software on Windows ⇒

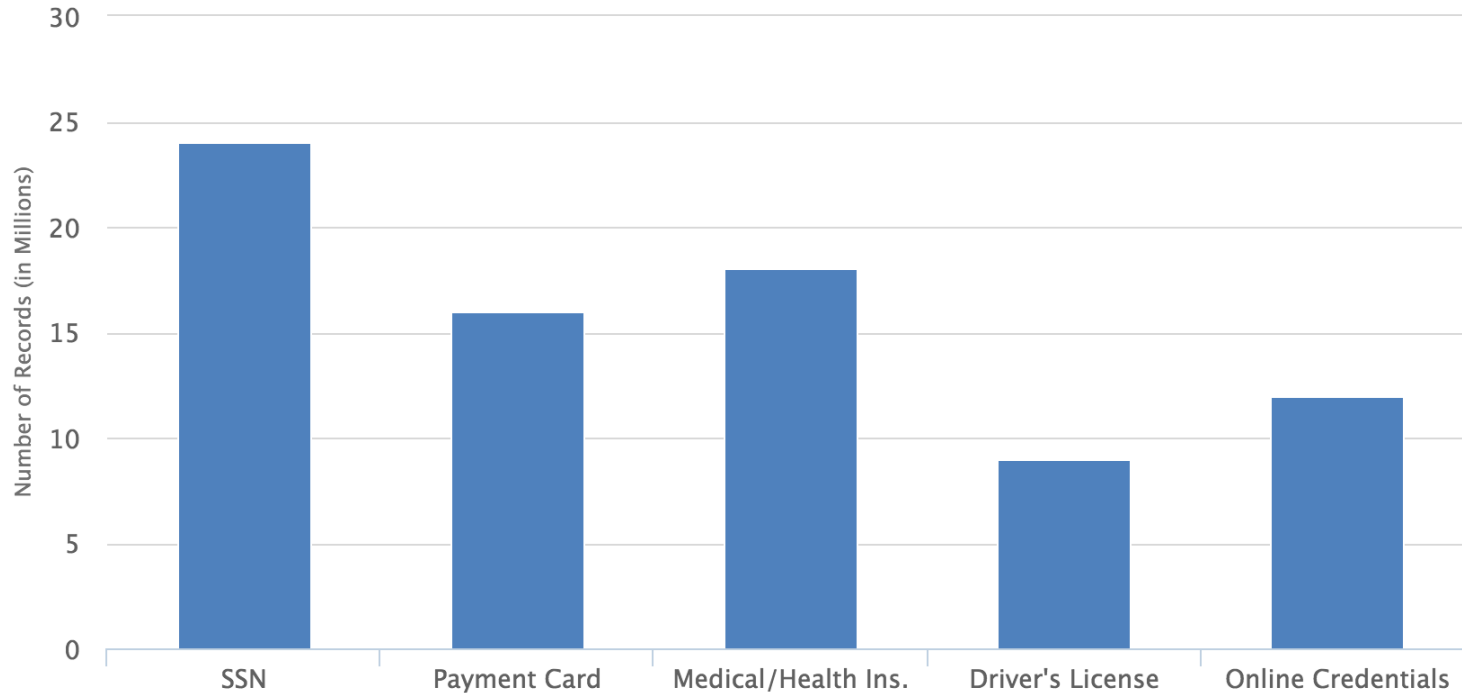
Siemens device controller on isolated network

More on this later in course

Server-side attacks

- Financial data theft: often credit card numbers
 - Example: Target attack (2013), \approx 140M CC numbers stolen
 - Many similar (smaller) attacks since 2000
- Political motivation:
 - Aurora, Tunisia Facebook (Feb. 2011), GitHub (Mar. 2015)
- Infect visiting users

Types of data stolen (2012-2015)



Source: California breach notification report, 2015

Example: Mpack

- PHP-based tools installed on compromised web sites
 - Embedded as an iframe on infected page
 - Infects browsers that visit site
- Features
 - management console provides stats on infection rates
 - Sold for several 100\$
 - Customer care can be purchased, one-year support contract
- Impact: 500,000 infected sites (compromised via SQL injection)
 - Several defenses: e.g. Google safe browsing

Insider attacks: example

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))  
    retval = -EINVAL;
```

Insider attacks: example

Hidden trap door in Linux (nov 2003)

- Allows attacker to take over a computer
- Practically undetectable change (uncovered via CVS logs)

Inserted line in wait4()

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))  
    retval = -EINVAL;
```

Looks like a standard error check, but ...

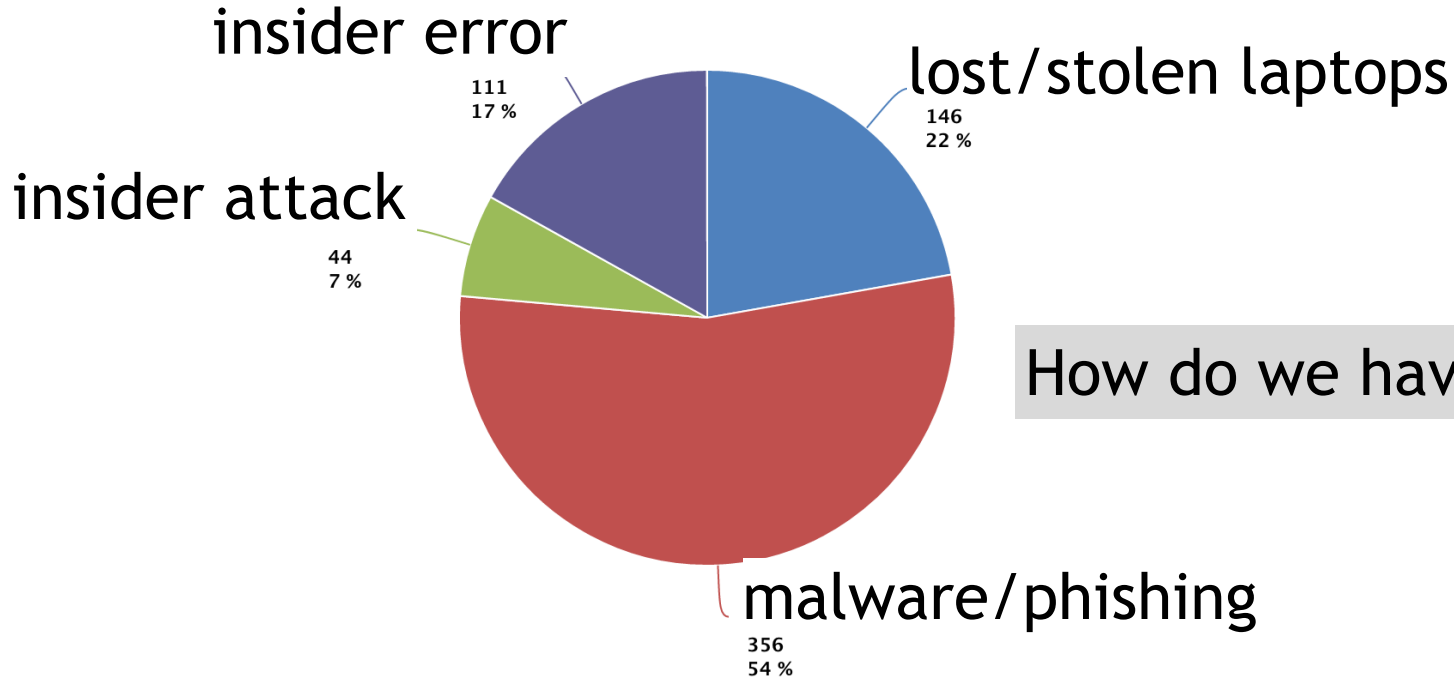
Many more examples

- Access to SIPRnet and a CD-RW: 260,000 cables \Rightarrow Wikileaks
- SysAdmin for city of SF government.
 Changed passwords, locking out city from router access
- Inside logic bomb took down 2000 UBS servers

⋮

Can security technology help?

How companies lose data



How do we have this data?



Introduction

The Marketplace for Vulnerabilities

Marketplace for Vulnerabilities

Option 1: bug bounty programs (many)

- Google Vulnerability Reward Program: up to \$20K
- Microsoft Bounty Program: up to \$100K
- Mozilla Bug Bounty program: \$7500
- Pwn2Own competition: \$15K

Option 2:

- Zero day initiative (ZDI), iDefense: \$2K - \$25K

Example: Mozilla

Novel vulnerability and exploit, new form of exploitation or an exceptional vulnerability	High quality bug report with clearly exploitable critical vulnerability ₁	High quality bug report of a critical or high vulnerability ₂	Minimum for a high or critical vulnerability ₃	Medium vulnerability
\$10,000+	\$7,500	\$5,000	\$3,000	\$500 - \$2500

Marketplace for Vulnerabilities

Option 3: black market

ADOBE READER	\$5,000-\$30,000
MAC OSX	\$20,000-\$50,000
ANDROID	\$30,000-\$60,000
FLASH OR JAVA BROWSER PLUG-INS	\$40,000-\$100,000
MICROSOFT WORD	\$50,000-\$100,000
WINDOWS	\$60,000-\$120,000
FIREFOX OR SAFARI	\$60,000-\$150,000
CHROME OR INTERNET EXPLORER	\$80,000-\$200,000
IOS	\$100,000-\$250,000

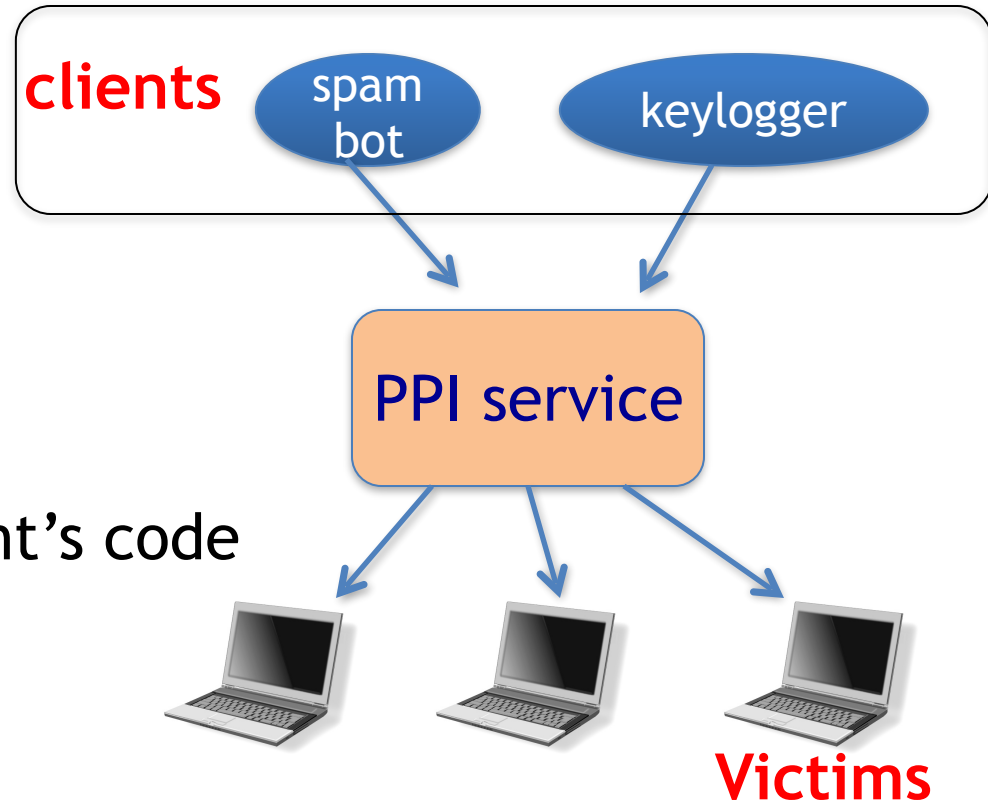
Source: Andy Greenberg (Forbes, 3/23/2012)

Marketplace for owned machines

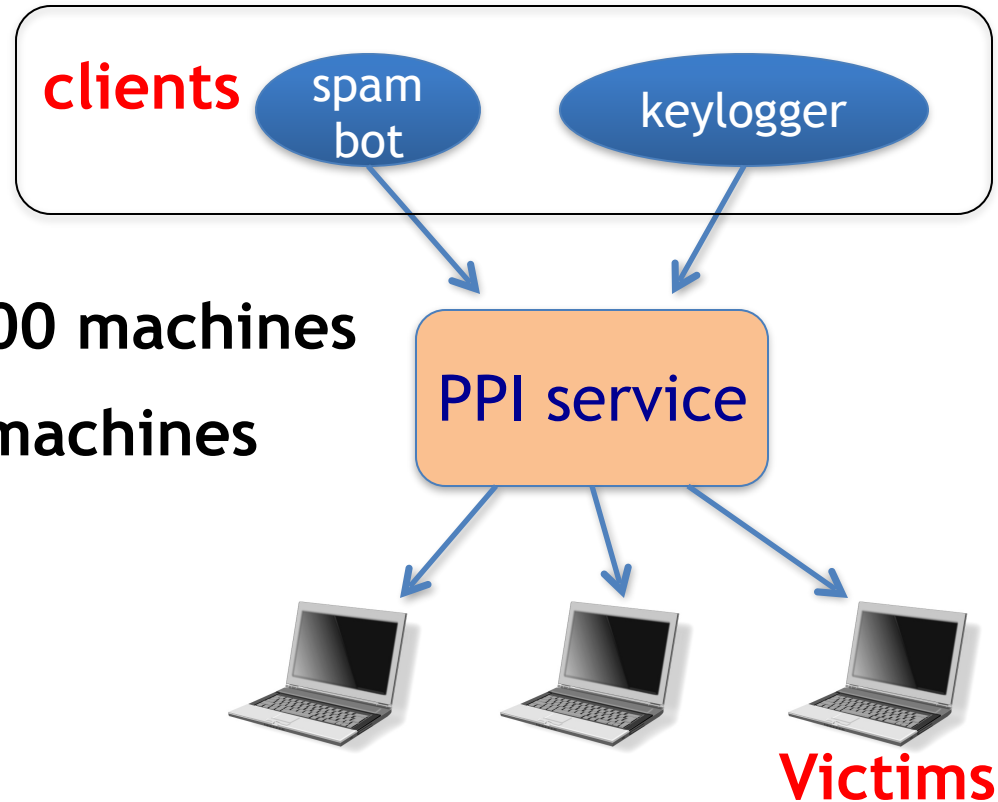
Pay-per-install (PPI) services

PPI operation:

1. Own victim's machine
2. Download and install client's code
3. Charge client



Marketplace for owned machines



Cost: US - 100-180\$ / 1000 machines

Asia - 7-8\$ / 1000 machines

This course

Goals:

- Be aware of exploit techniques
- Learn to defend and avoid common exploits
- Learn to architect secure systems

This course

Part 1: **basics** (architecting for security)

- Securing apps, OS, and legacy code
Isolation, authentication, and access control

Part 2: **Web security** (defending against a web attacker)

- Building robust web sites, understand the browser security model

Part 3: **network security** (defending against a network attacker)

- Monitoring and architecting secure networks.

Part 4: **securing mobile applications**

Don't try this at home !

Ken Thompson's clever Trojan