Systems and Network Security
CSE 628/628A

Sandeep K. Shukla
Indian Institute of Technology
Kanpur

Module 0.1: Introduction

Introduction to the context and landscape

Acknowledgements

- Dan Boneh (Stanford University)
- John C. Mitchell (Stanford University)
- Nicolai Zeldovich (MIT)
- Jungmin Park (Virginia Tech)
- Patrick Schaumont (Virginia Tech)
- Web Resources

The computer security problem

Two factors:

- **Lots of buggy software** (and gullible users)
- **Money can be made from finding and exploiting vulnerabilities**.

The computer security problem

Two factors:

- **Lots of buggy software** (and gullible users)
- **Money can be made from finding and exploiting vulnerabilities**.

1. Marketplace for vulnerabilities

The computer security problem

Two factors:

- **Lots of buggy software** (and gullible users)
- **Money can be made from finding and exploiting vulnerabilities**.

1. Marketplace for vulnerabilities
2. Marketplace for owned machines (PPI)

The computer security problem

Two factors:

- **Lots of buggy software** (and gullible users)
- **Money can be made from finding and exploiting vulnerabilities**.

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

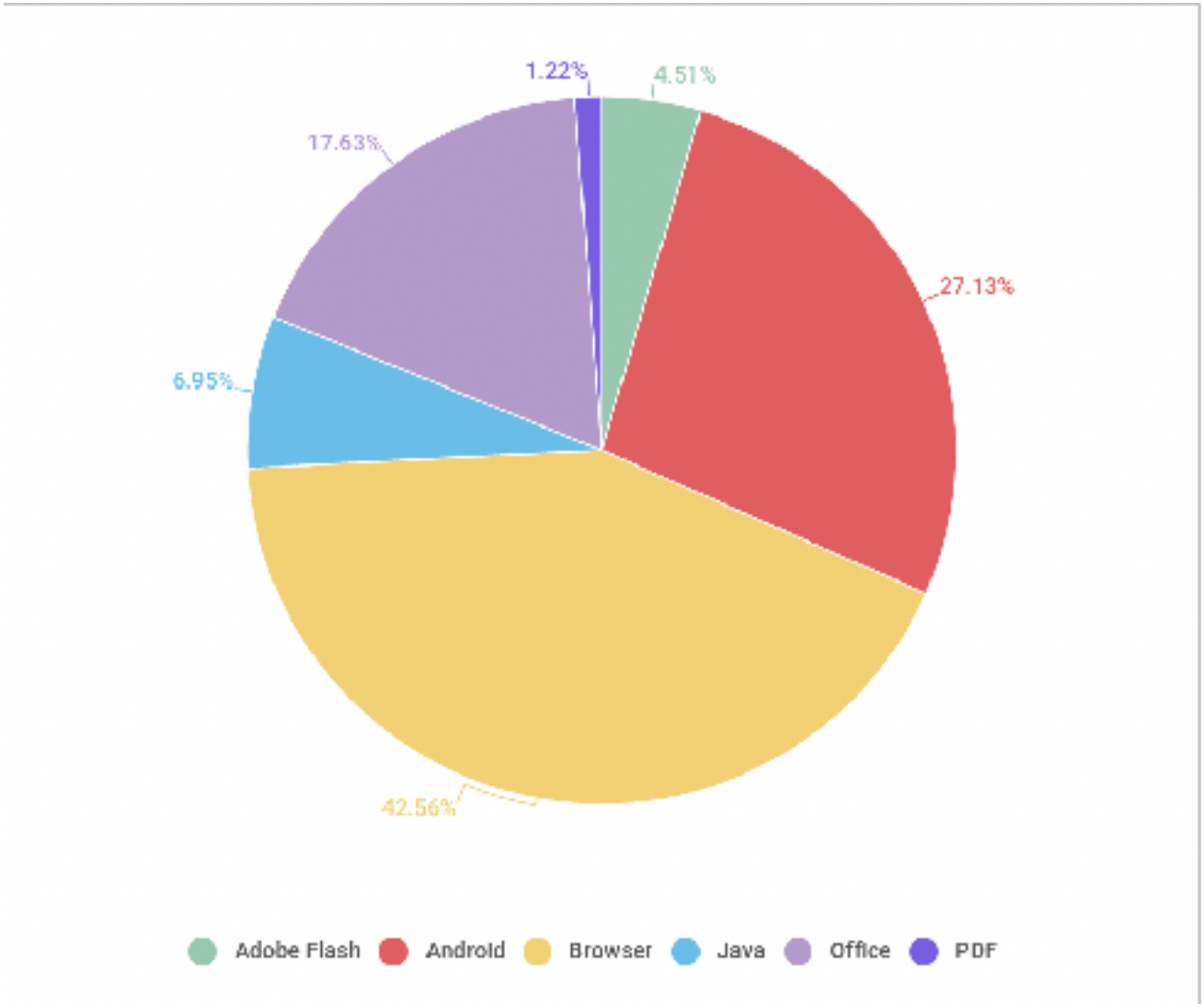
The computer security problem

Two factors:

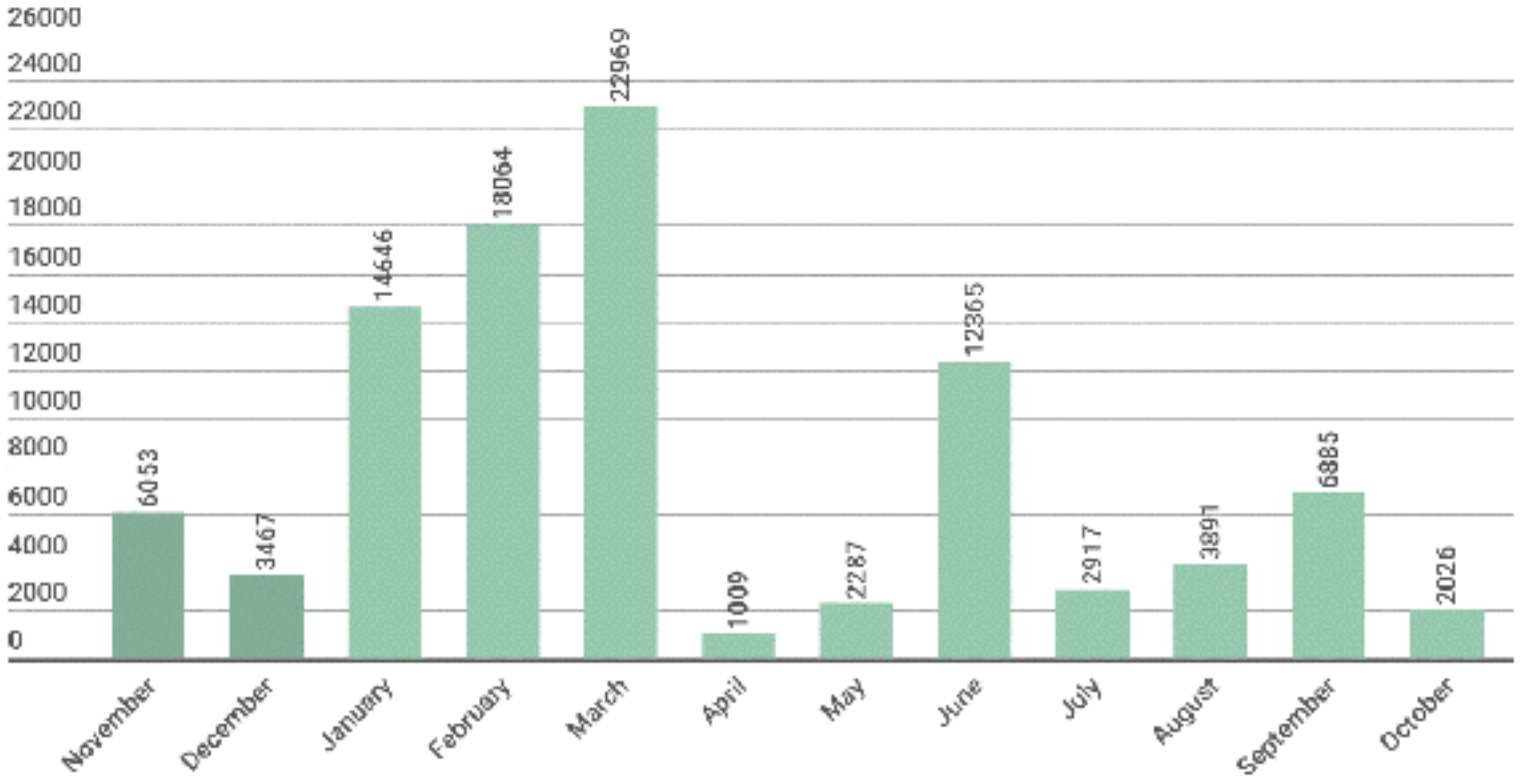
- **Lots of buggy software** (and gullible users)
- **Money can be made from finding and exploiting vulnerabilities**.

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

Type of Applications Attacked in 2017

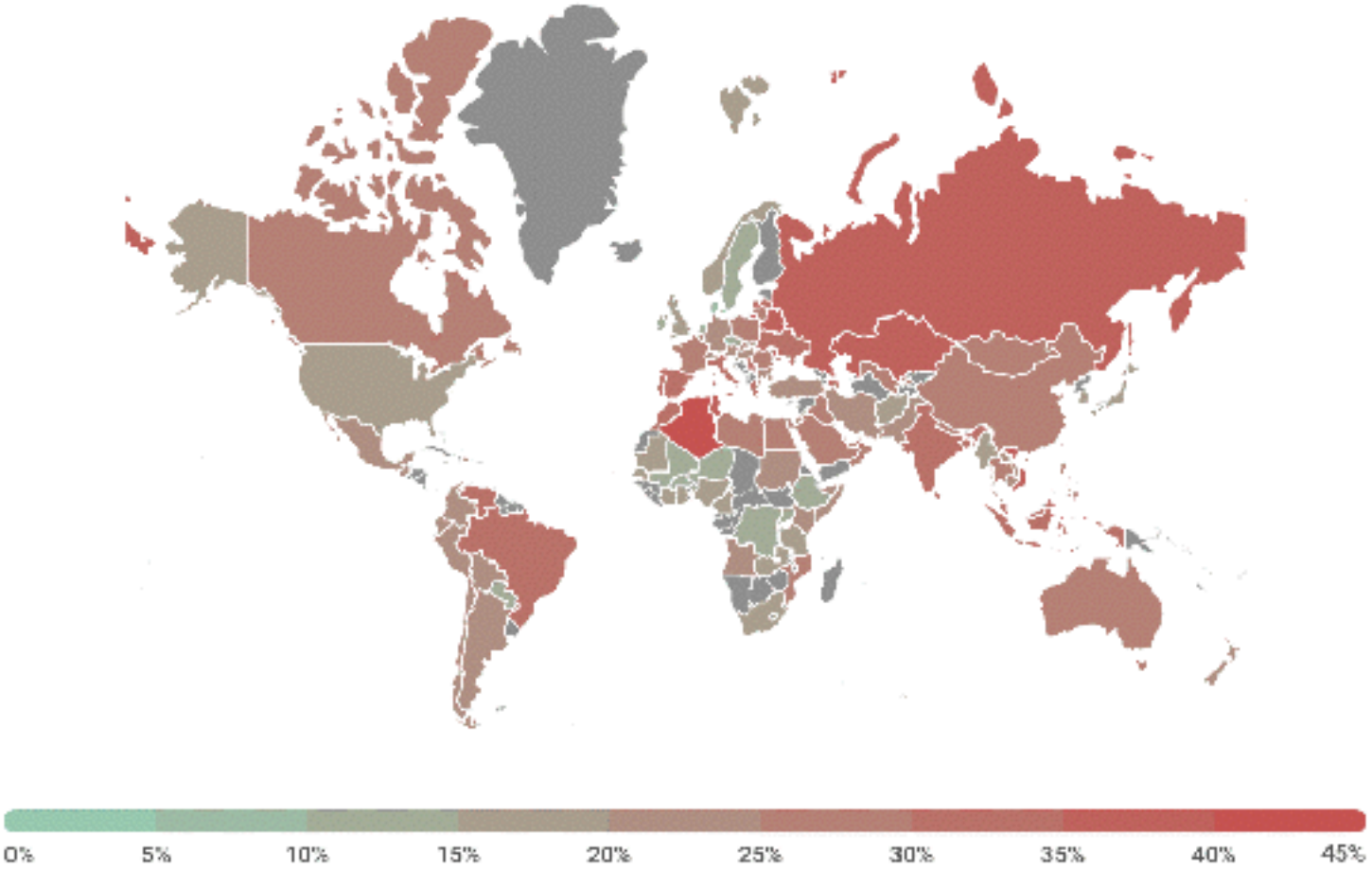


96K modifications and 38 new families of Ransomware in 2017



KASPERSKY lab

Geography of malicious web attacks in 2017 (ranked by percentage of users attacked)



KASPERSKY lab

Malware Galore

- **29.4%** of user computers were subjected to at least one Malware-class web attack over the year.
- Kaspersky Lab solutions repelled **1 188 728 338** attacks launched from online resources located all over the world.
- **199 455 606** unique URLs were recognized as malicious by web antivirus components.
- Kaspersky Lab's web antivirus detected **15 714 700** unique malicious objects.
- **939 722** computers of unique users were targeted by encryptors.
- Kaspersky Lab solutions blocked attempts to launch malware capable of stealing money via online banking on **1 126 701** devices.

Vulnerabilities Galore

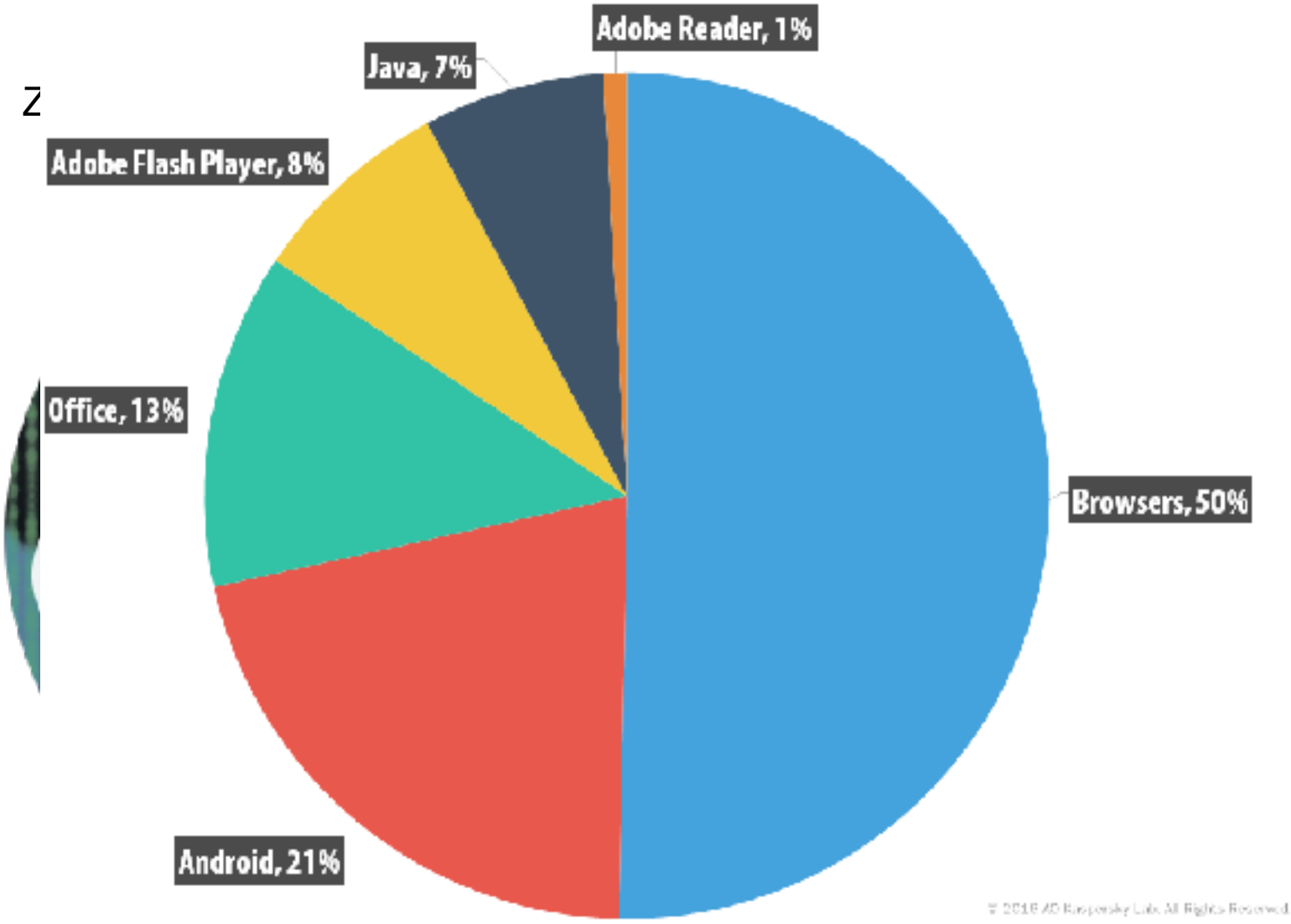
Zero Day Vulnerabilities in last 3 years

For real time map
https://cybermap.kaspersky.com/

Source: Symantec Report

Data: http://cve.mitre.org/

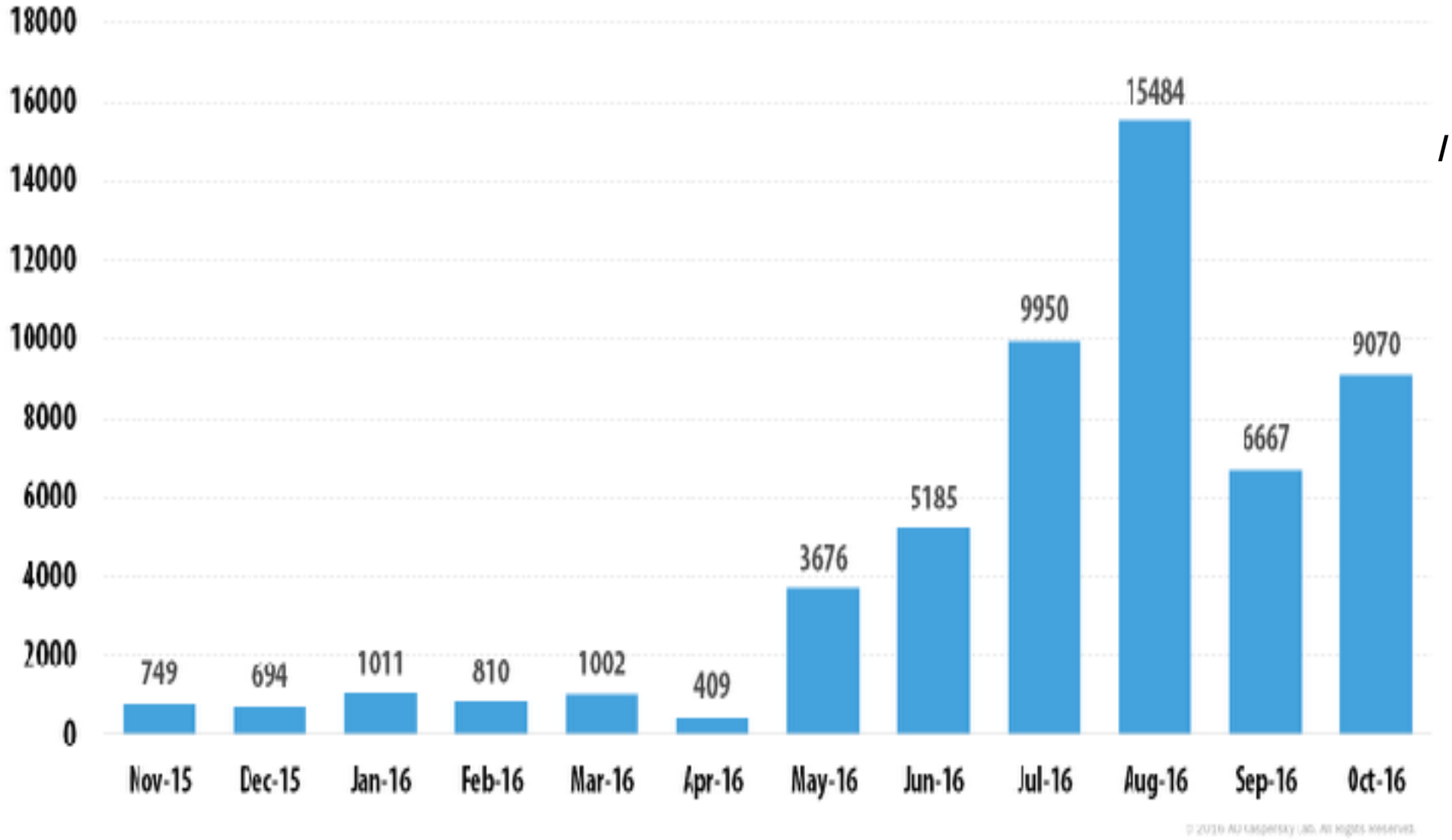
Vulnerabilities Galore



Data: http://cve.mitre.org/

Source: Symantec Report

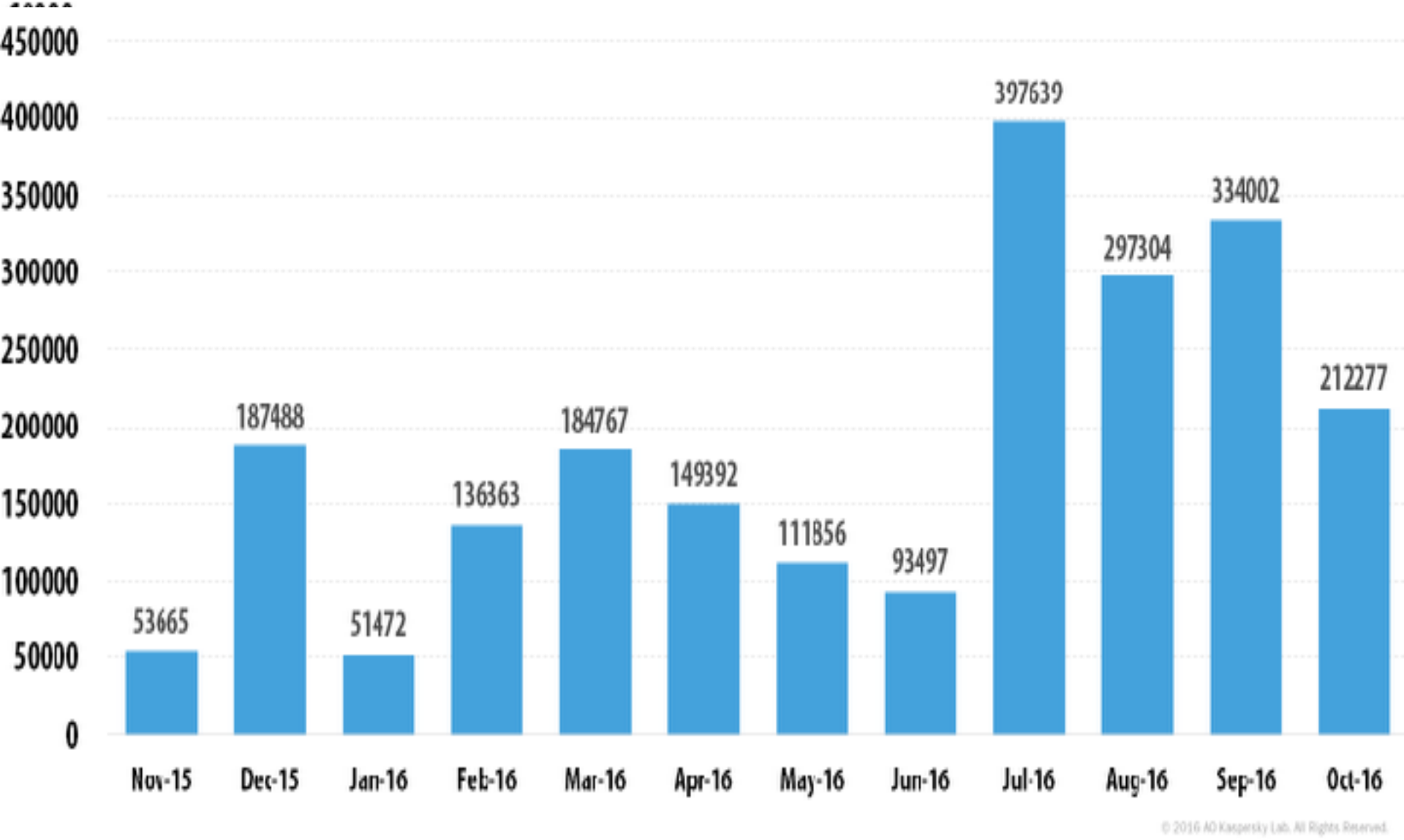
Vulnerabilities Galore



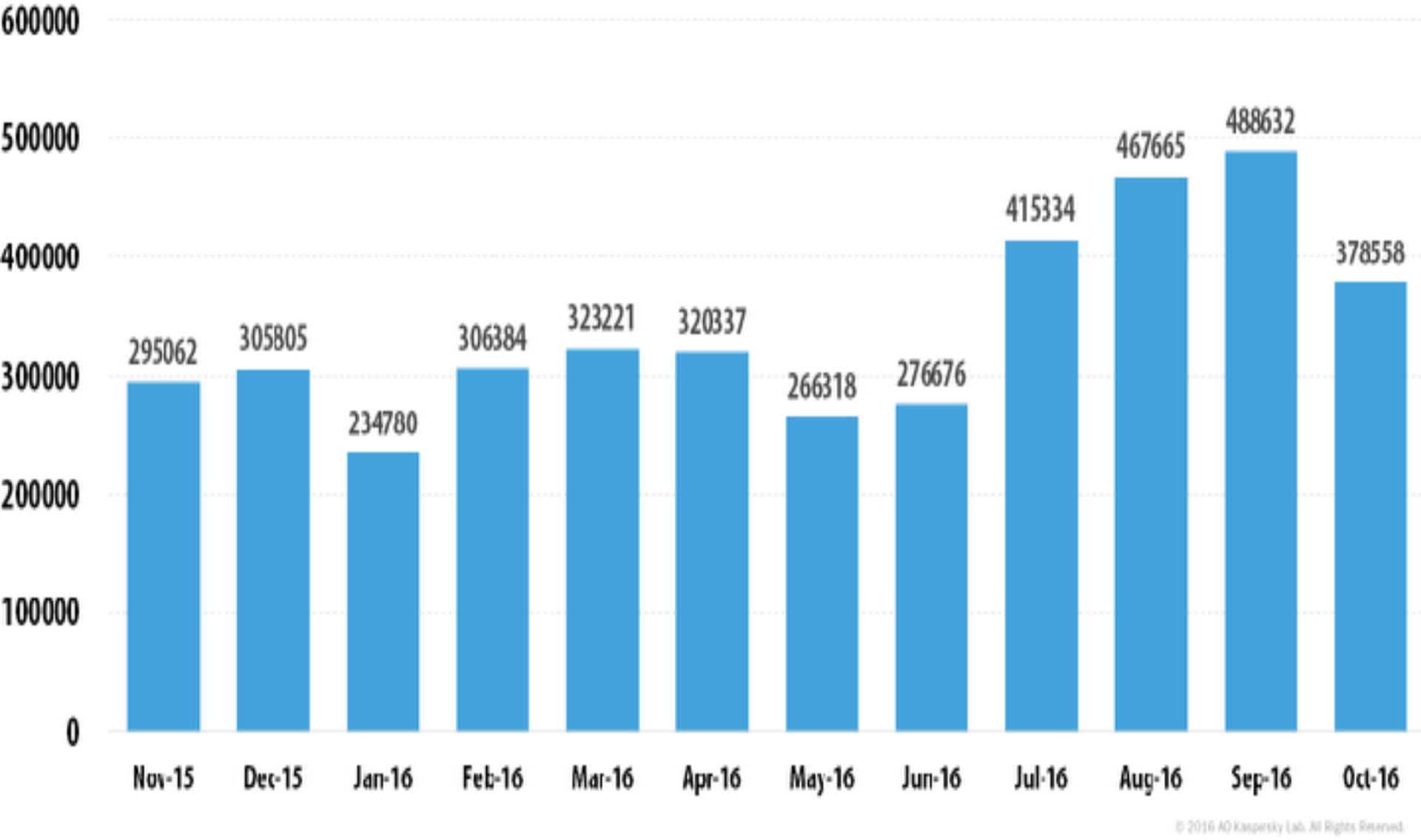
Source: Symantec Report

Data: http://cve.mitre.org/

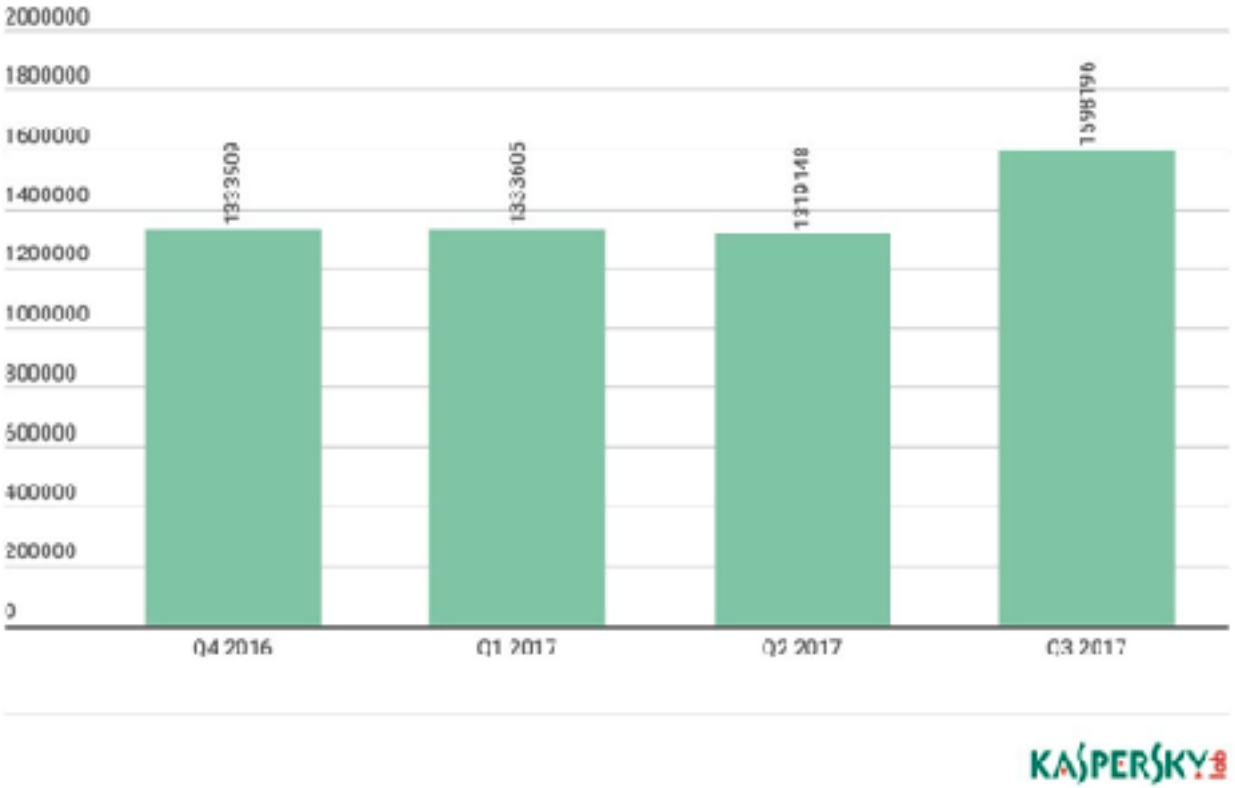
Vulnerabilities Galore



Vulnerabilities Galore



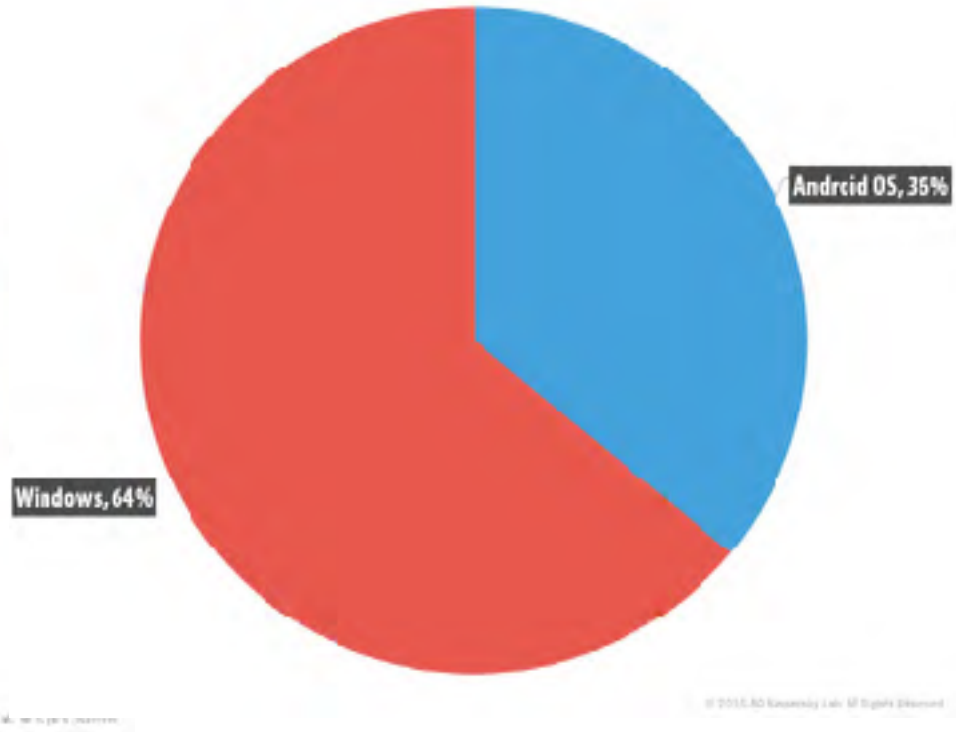
Mobile malware (Q4 2016 – Q3 2017)



date

The rise of mobile malware installation packages (Kaspersky Security Bulletin 2017)

% in Mobile OS targeted by Financial Malware 2015 vs 2016



Introduction

Sample attacks

The computer security problem

Two factors:

- **Lots of buggy software** (and gullible users)
- **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 client machines

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

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$), 24 hours (100$)

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)

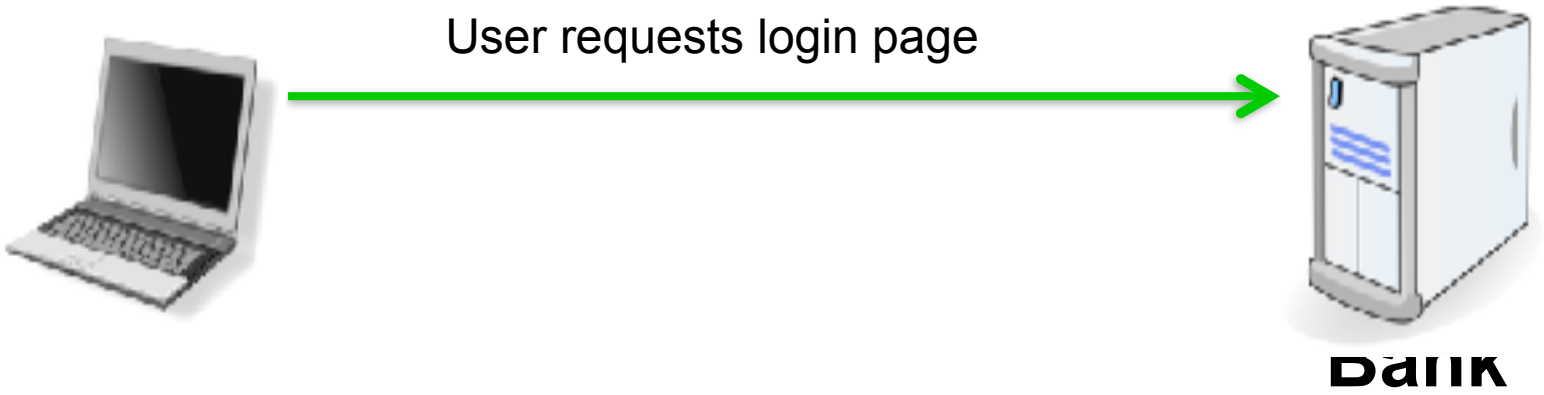
Bank

Why own machines:

2. Steal user credentials and inject ads

keylog for banking passwords, web passwords, gaming pwds.

Example: SilentBanker (and many like it)

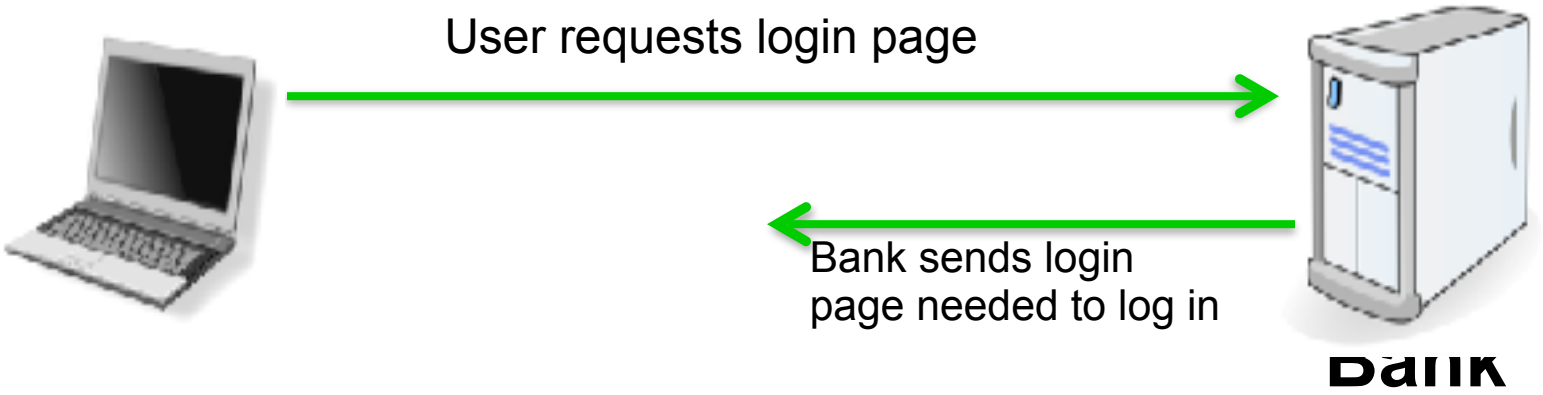


Why own machines:

2. Steal user credentials and inject ads

keylog for banking passwords, web passwords, gaming pwds.

Example: SilentBanker (and many like it)

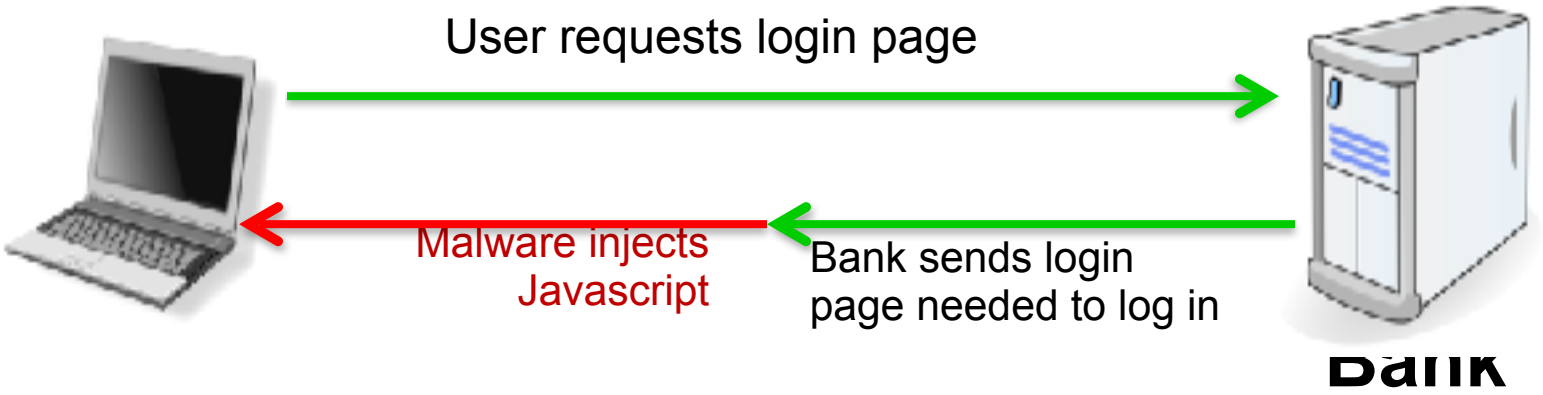


Why own machines:

2. Steal user credentials and inject ads

keylog for banking passwords, web passwords, gaming pwds.

Example: SilentBanker (and many like it)

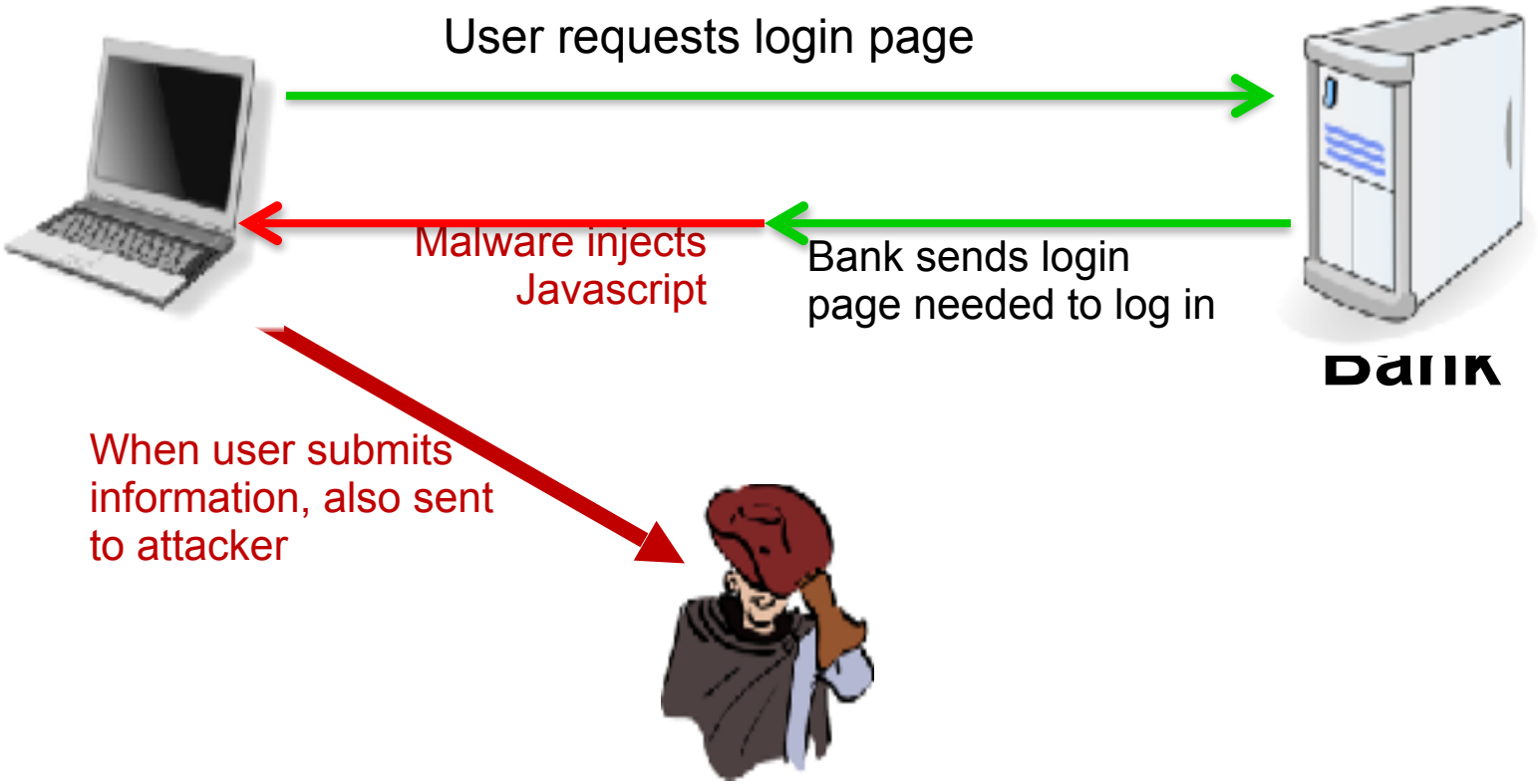


Why own machines:

2. Steal user credentials and inject ads

keylog for banking passwords, web passwords, gaming pwds.

Example: SilentBanker (and many like it)

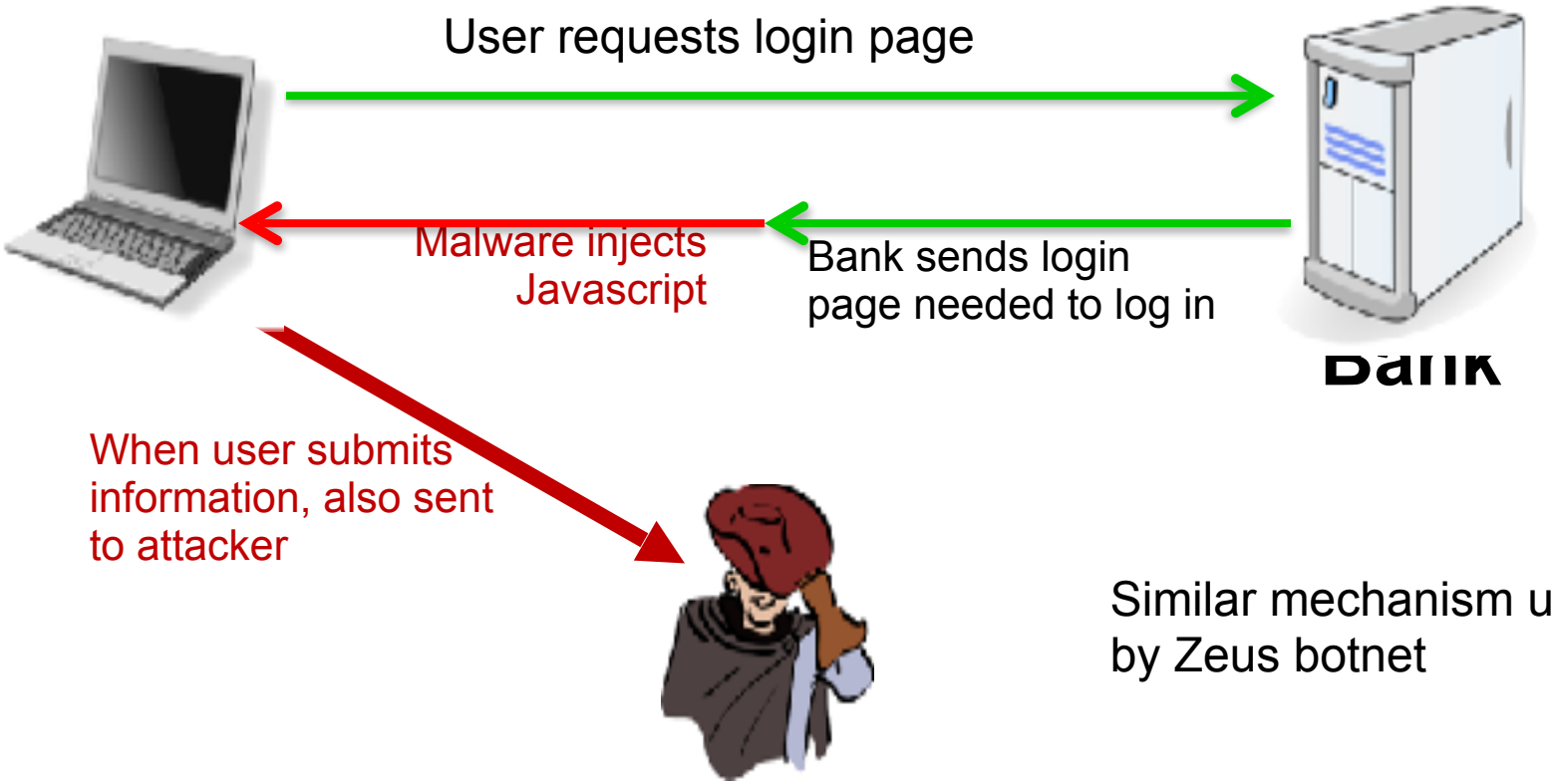


Why own machines:

2. Steal user credentials and inject ads

keylog for banking passwords, web passwords, gaming pwds.

Example: SilentBanker (and many like it)



Similar mechanism used by Zeus botnet

Why own machines:

3. Spread to isolated systems

Example: **Stuxnet**

Windows infection ⇒

Why own machines:

3. Spread to isolated systems

Example: **Stuxnet**

Windows infection ⇒

Siemens PCS 7 SCADA control software on Windows ⇒

Why own machines:

3. Spread to isolated systems

Example: **Stuxnet**

Windows infection ⇒

Siemens PCS 7 SCADA control software on Windows ⇒

Siemens device controller on isolated network

Server-side attacks

- Financial data theft: often credit card numbers
  - Example: Uber Attack, Equifax hack, Jio hack, Aadhaar Hack (2017)
  - Target attack (2013), ≈ 140M CC numbers stolen
  - Many similar (smaller) attacks since 2000
- Political motivation:
  - Aurora, Tunisia Facebook (Feb. 2011), GitHub (Mar. 2015)
  - **US Election 2016**
- Infect visiting users

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 …

See: http://lwn.net/Articles/57135/

Many more examples

- Access to SIPRnet and a CD-RW: 260,000 cables ⇒ Wikileaks

- SysAdmin for city of SF government.
  Changed passwords, locking out city from router access

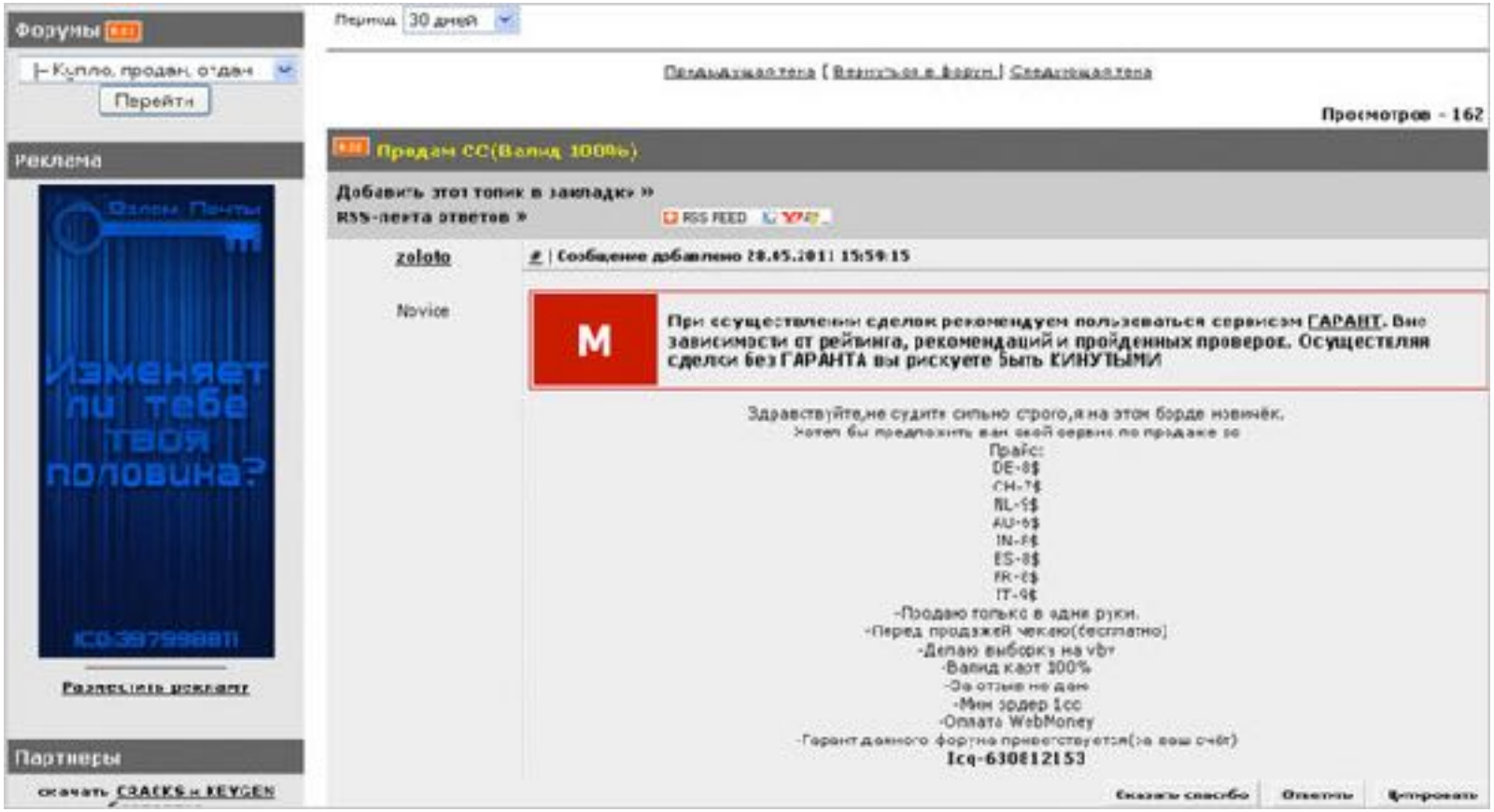
- Inside logic bomb took down 2000 UBS servers

Can security technology help?

Introduction

The Marketplace for Vulnerabilities

Hacker zoloto offered credit cards for sale on the Web site HackZone .ru.



Marketplace for Vulnerabilities

Option 1: bug bounty programs (many)

- Google Vulnerability Reward Program: up to 100K $
- Microsoft Bounty Program: up to 100K $
- Mozilla Bug Bounty program: 500$ - 3000$
- Pwn2Own competition: 15K $

Option 2:

- ZDI, iDefense: 2K – 25K $

Marketplace for Vulnerabilities

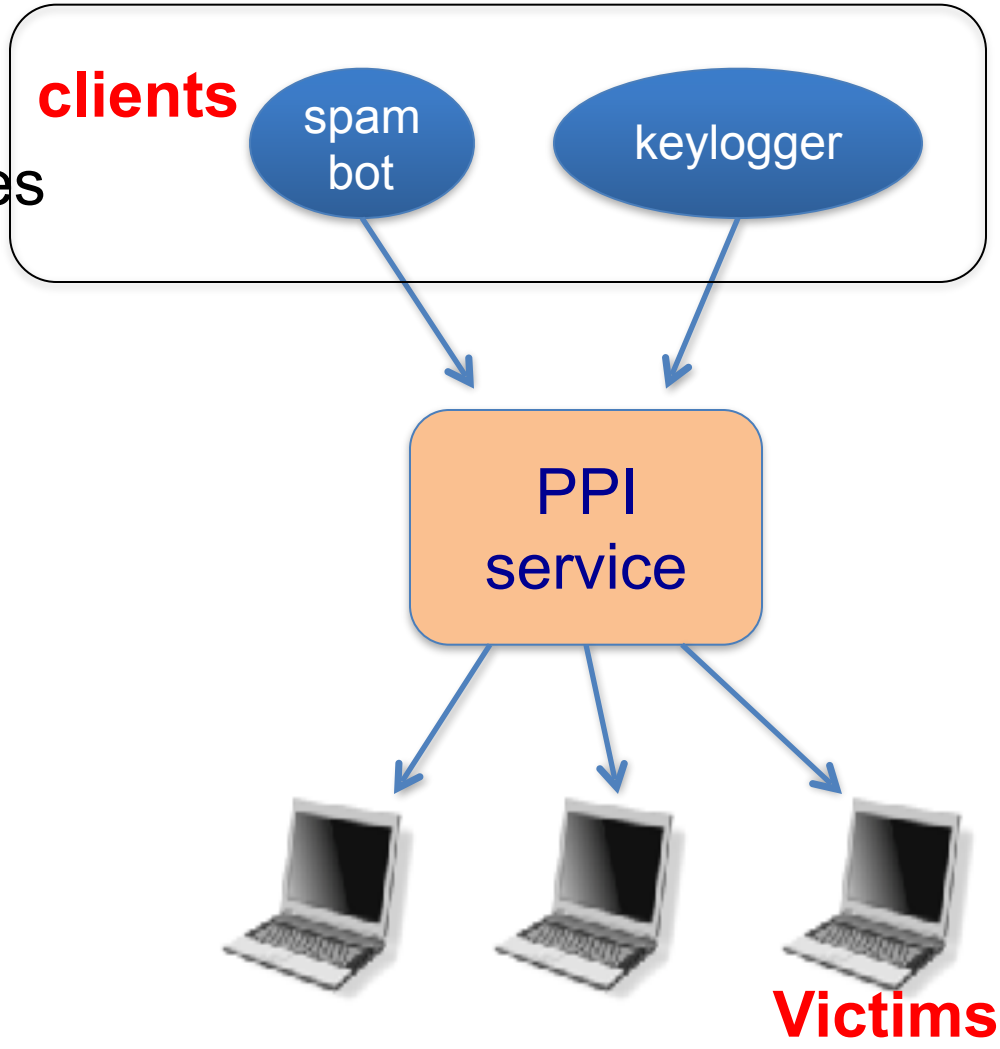
Option 3: black market

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

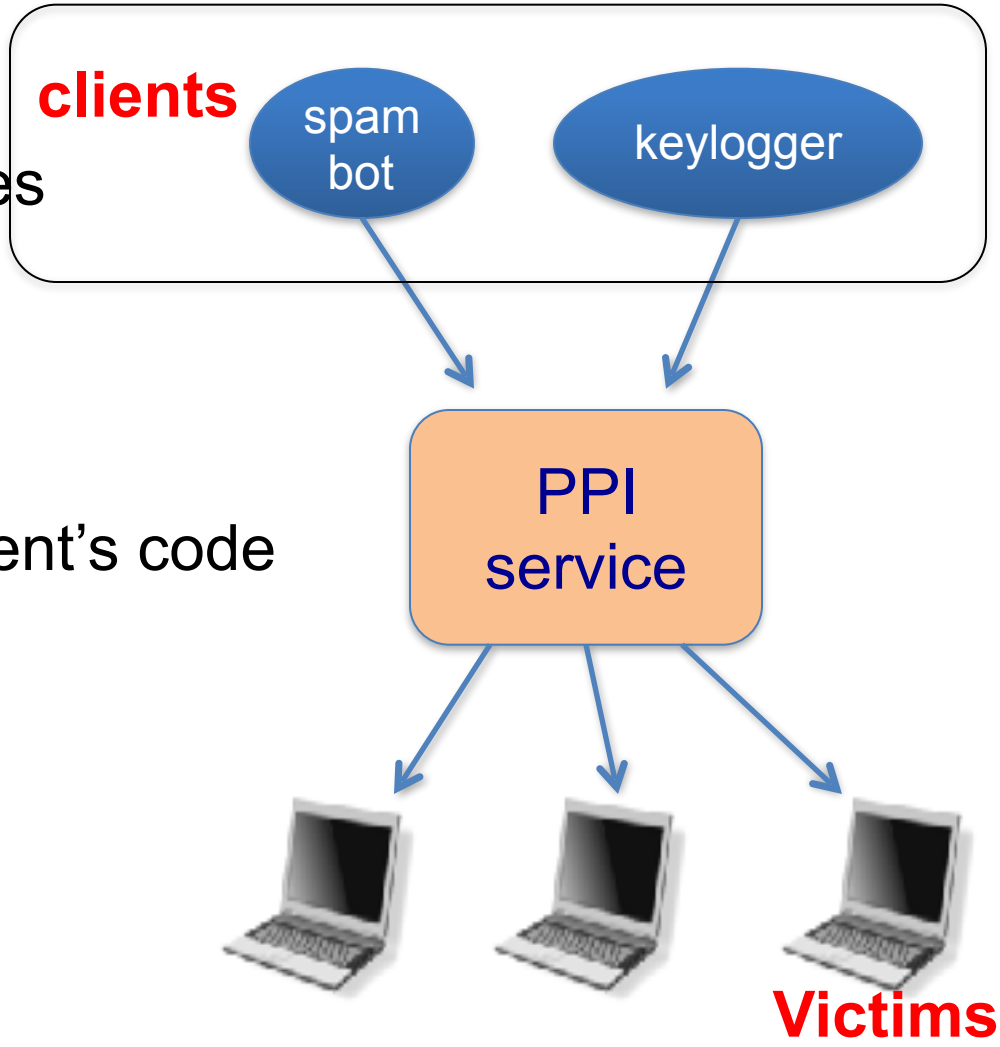
Pay-per-install (PPI) services



Source: Cabalerro et al. (www.icir.org/vern/papers/ppi-usesec11.pdf)

Marketplace for owned machines

Pay-per-install (PPI) services

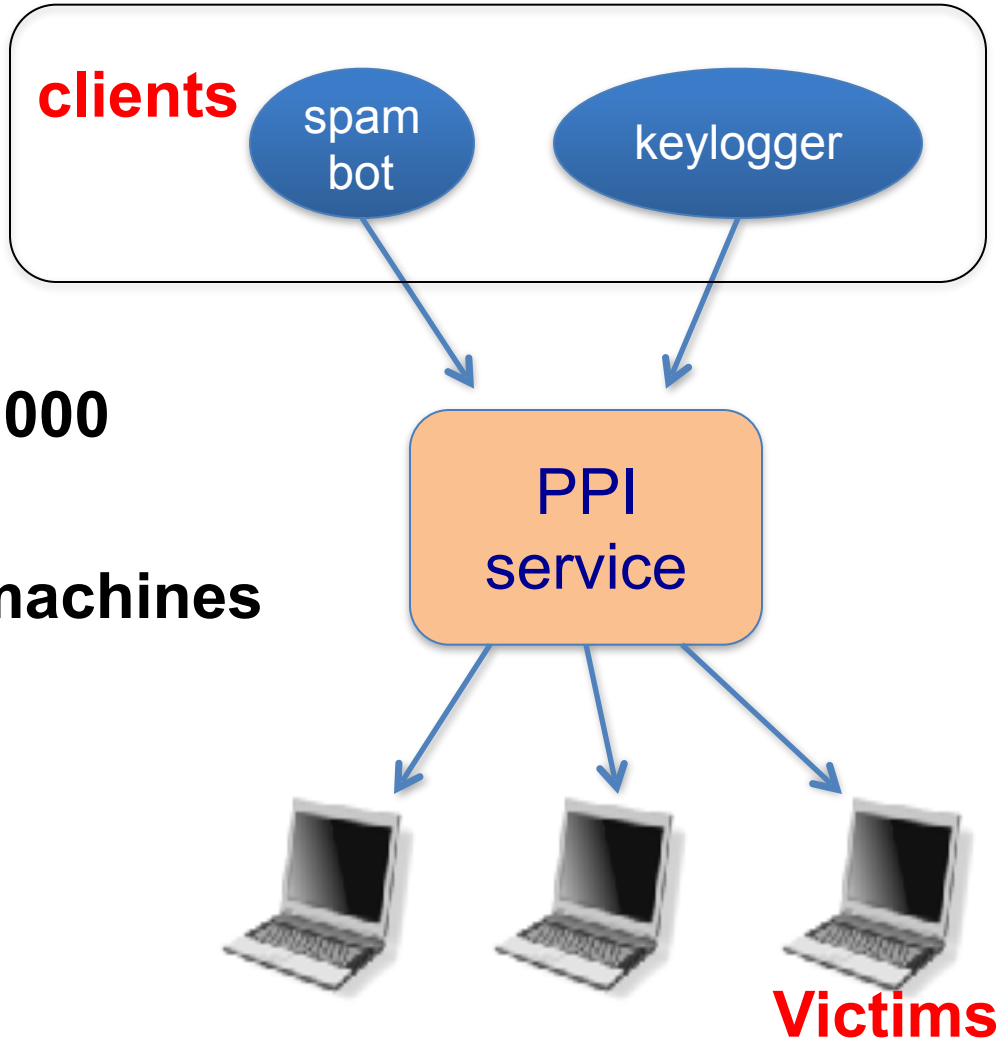


PPI operation:

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

Source: Cabalerro et al. (www.icir.org/vern/papers/ppi-usesec11.pdf)

Marketplace for owned machines



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

Asia - 7-8$ / 1000 machines

Source: Cabalerro et al. (www.icir.org/vern/papers/ppi-usesec11.pdf)

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

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

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: Mobile security

Part 5: Critical Infrastructure Security

Module 0.2

Ken Thompson’s clever Trojan

- Ken Thompson, co-author of UNIX, recounted a story of how he created a version of the C compiler that, when presented with the source code for the "login" program, would automatically compile in a backdoor to allow him entry to the system. This is only half the story, though. In order to hide this Trojan horse, Ken also added to this version of "cc" the ability to recognize if it was recompiling itself to make sure that the newly compiled C compiler contained both the "login" backdoor, and the code to insert both Trojans into a newly compiled C compiler. In this way, the source code for the C compiler would never show that these Trojans existed.

What is Security?

- Achieving something in the presence of adversaries
  - Internet is full of adversaries
  - There are insider adversaries for air-gapped systems
  - Thus design of systems need to worry about security

- A High Level Plan for Security Centric System Design
  - Policy: “Only X can access file F”
  - Common goals: Confidentiality, Integrity, Availability
  - Threat Models: “Can Y physically grab the file server?”
  - Mechanisms: The knobs that can be controlled to uphold your security policy, but also be flexible to uphold a different policy
  - Resulting Goal: “No way the adversary in the threat model to violate policy”

Why is security hard?

- Need to guarantee policy, assuming threat models
- Difficult to think of all possible ways that attacker might break in
- Realistic Threat models are open-ended (Negative models)
- Easy to check a positive goal (“X has access to File F”)
- Weakest link matters
- Iterative process: Design, Update Threat Model as necessary, assess vulnerability → Design, Update …..

What if perfect Security is not achievable?

- Best effort
- Each system will have some breaking point – need to analyze and understand – e.g., penetration testing
- Need to manage security risk vs. benefit tradeoff
- Risk based security model
- Manual auditing often can help
- Make the cost of attack high – deterrence
  - Either by law
  - Technologically

Why Policy matters in Security

- Example: Sarah Palin's email account hacked
  - Yahoo accounts have username/password and security questions
  - User can login with username/password
  - If user forgets password – can reset by answering security question
  - Security questions are sometimes easier to guess
  - Some one guessed Palin's highschool, birthday etc
  - Policy amounts to: can log in with either password or security questions

Policy Matters: Example 2

- Mat Honan's accounts at Amazon, Apple, Google etc hacked
  - Gmail password reset: send a verification link to a backup email
    - Google helpfully prints part of the backup email address
    - Mat Honan's backup address was his Apple @me.com account
  - Apple password reset: need billing address, last 4 digits of credit card
    - Address can be easily found, how do you get last 4 digits of credit card
  - Amazon: can add a credit card to an account, no password required
    - Amazon password reset: provide any of user's credit card number
    - Amazon will not print credit card number but print last 4 digits

What to do?

- Think hard about implications of policy statements
- Some policy checking tools can help – but you need to specify ‘what is bad’
- Difficult in distributed systems: don’t know what everyone is doing

What might go wrong in threat models/ assumptions?

- Human factors not accounted for: ex. Phishing attack
- Computational assumptions change over time:
  - MIT's kerberos system used 56-bit DES keys since mid 1980s
  - Now it costs about $100 to get it cracked
- All SSL certificate CAs are fully trusted
  - To connect to an SSL-enabled website, your browser verifies the certificate
    - Certificate is a combination of server's host name, and cryptographic key, signed by a trusted CA
  - 100s of CAs are trusted by most browsers
  - In 2011, two CAs were compromised – issued fake certificates for many domains (google, yahoo, tor, …)
  - http://en.wikipedia.org/wiki/Comodo_Group
  - http://en.wikipedia.org/wiki/DigiNotar

Limitations in Assumptions

- Assuming your hardware is trustworthy
  - If NSA is your adversary – it is not necessarily true
    - https://www.schneier.com/blog/archives/2013/12/more_about_the.html
- Assuming good randomness in cryptography
  - Often source of randomness may not be good, and keys may be compromised
  - https://factorable.net/weakkeys12.extended.pdf
- Assuming OS to be secure
  - Backdoors? Trojans?
- Machine is disconnected from the Network
  - Did not stop stuxnet worm

What to do to avoid limitations in threat models?

- More explicit and formalized threat models to understand possible weaknesses
- Simpler and more general threat models
- Better design may lessen reliance on certain assumptions
  - E.g., alternative trust models that does not rely on full trust in CAs
  - E.g., authentication mechanisms that aren't susceptible to phishing

Problems with mechanisms

- Bugs in security mechanism (e.g. OS kernel) lead to vulnerabilities
- If application is enforcing security, application bugs can lead to vulnerabilities
- Example: Apple's iCloud password guessing rate limits http://thenextweb.com/apple/2014/09/01/this-could-be-the-apple-icloud-flaw-that-led-to-celebrity-photos-being-leaked/
- Example: Missing access control checks in Citigroup's credit card website http://www.nytimes.com/2011/06/14/technology/14security.html?_r=0
- Example: Android's Java SecureRandom weakness leads to bitcoin theft – the randomization seed was not being changed sometimes leading to easy guess of private keys

Some implementation bugs

- Buffer overflow
- Use-after-free (e.g., dereference a already deallocated pointer)
- Double-free
- Decrementing stack pointer past the end of stack – into some other memory location
  - http://www.invisiblethingslab.com/resources/misc-2010/xorg-large-memory-attacks.pdf
- Not checking sanity of inputs
  - Sql injection
  - Command injection

Content of this Course

Part 1: Basics (architecting for security)

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

Content of 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

Content of 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

Course Content (cont.)

Course Content (cont.)

Part 4: Mobile security

Course Content (cont.)

Part 4: Mobile security

- Android Security

Course Content (cont.)

Part 4: Mobile security

- Android Security

Part 5: Critical Infrastructure Security

Course Content (cont.)

Part 4: Mobile security

- Android Security

Part 5: Critical Infrastructure Security

- Security of Industrial Systems

What is Security?

- Achieving something in the presence of adversaries

What is Security?

- Achieving something in the presence of adversaries
  - Internet is full of adversaries

What is Security?

- Achieving something in the presence of adversaries
  - Internet is full of adversaries
  - There are insider adversaries for air-gapped systems

What is Security?

- Achieving something in the presence of adversaries
  - Internet is full of adversaries
  - There are insider adversaries for air-gapped systems
  - Thus design of systems need to worry about security

Major Learning Objectives

- Discover software bugs that pose cyber security threats
  - explain and recreate exploits of such bugs in realizing a cyber attack
  - explain how to fix the bugs to mitigate such threats

Major Learning Objectives (2)

- Discover cyber attack scenarios to web browsers, and web servers
  - explain various possible exploits
  - recreate cyber attacks on browsers and servers with existing bugs
  - explain how to mitigate such threats

Major Learning Objectives (3)

- Discover and explain cyber security holes in standard networking protocols
  - both in network architecture, standard protocols (such as TCP/IP, ARP, DNS, Ethernet, BGP etc),
  - explain mitigation methods and revisions of standards based on cyber threats.

Major Learning Objectives (4)

- Discover and explain mobile software bugs posing cyber security threats
  - explain and recreate exploits
  - explain mitigation techniques.

Major Learning Objectives (5)

- Articulate the urgent need for cyber security in critical computer systems
  - explain various threat scenarios

Major Learning Objectives (6)

- Articulate the well known cyber attack incidents
  - explain the attack scenarios
  - explain mitigation techniques

Major Learning Objectives (7)

- Explain the difference between
  - Systems Cyber Security
  - Network Cyber Security,
  - Cryptography
  - crypto-protocols etc.


HACKER
ATTACK
CYBER
CRIME
HACKING
TERRORISM
NETWORK
INTERNET
GROUPS
TARGET
COMPUTER
PORNOGRAPHY
OFFENSIVE
INDIVIDUALS
CRIMINAL
HACKERS
INFORMATION
ALTERING
UNAUTHORIZED
WARFARE
COMMUNICATIONS
NETWORKS
INTERNATIONAL
SERVICE
THEFT
ILLEGAL
CONTENT
CASES
MAIL
USING
SYSTEMS
DEVICES
CRIMES
SYSTEM
DEPARTMENT
WEBSITES
USED
ACTIVITIES
CHILD
DATA