RSA Conference 2016

San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

SESSION ID: CRWD-R08

CROWDPATCHING

It's Time to Take Vulnerability Fixing into Our Own Hands

Mitja Kolsek

CEO, ACROS Security
Co-founder, 0patch

@mkolsek

#RSAC

15 years of breaking in... in the same way

1. Find a public exploit for a recent vulnerability
2. Tailor exploit to work with your RAT
3. Mutate exploit until VirusTotal doesn't recognize it
4. Phish the target until you're in

EXPLOIT DATABASE + + virustotal +

But... We have all this cool technology

Beating around the bush

PATCH

We all want

...different things

We all want different things

Software vendors

- direct and opportunity costs
- deploying fixes is costly
- have better things to do

Users and administrators

- „The product should have been secure in the first place“
- hate downtime
- updating = risk breakage, not updating = risk ownage

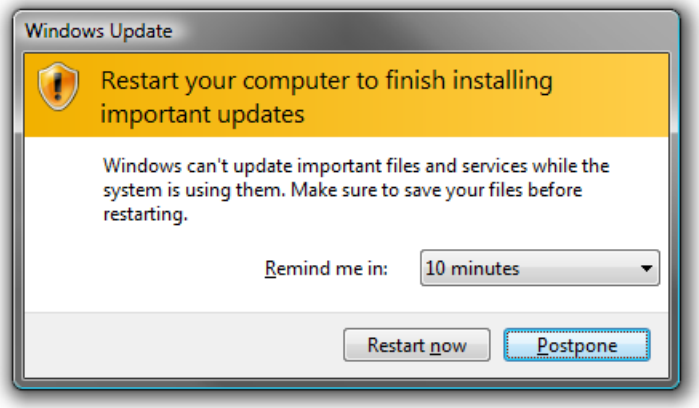
Security researchers

- Inherent conflict with vendors
- considered part of the problem

My Galaxy S4 vs. the Stagefright bug

- Reported to Google in April 2015
- Google fixed it in 2 days
- Publicly revealed in August 2015
- My Samsung Galaxy is still vulnerable today (after 10 months)

Windows updates



- 279 MS vulnerabilities in 2015*
- Computer restart always required
- February 2016 Updates
  - 33 CVEs
  - 18 Remote Code Execution bugs
  - Windows 7 ~ 200MB of changes

* Source: Secunia Vulnerability Review 2015

Updates: Days from release to install

176

* Source: NopSec, 2015 State of Vulnerability Risk Management

We couldn’t complete the updates

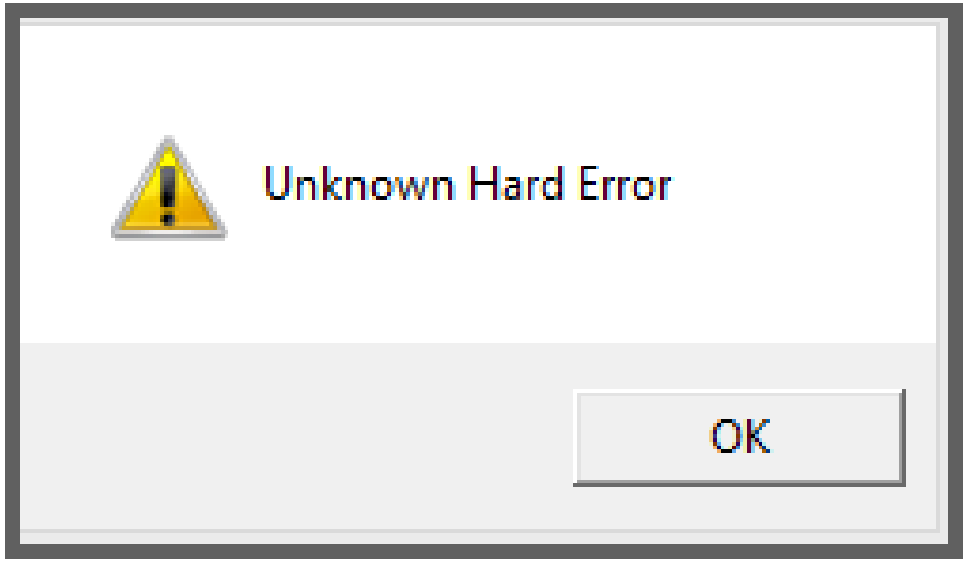
Undoing changes

Don’t turn off your computer

We're sorry, but you can't go back

The files we need to take you back to a previous version of Windows were removed from this PC.

Close

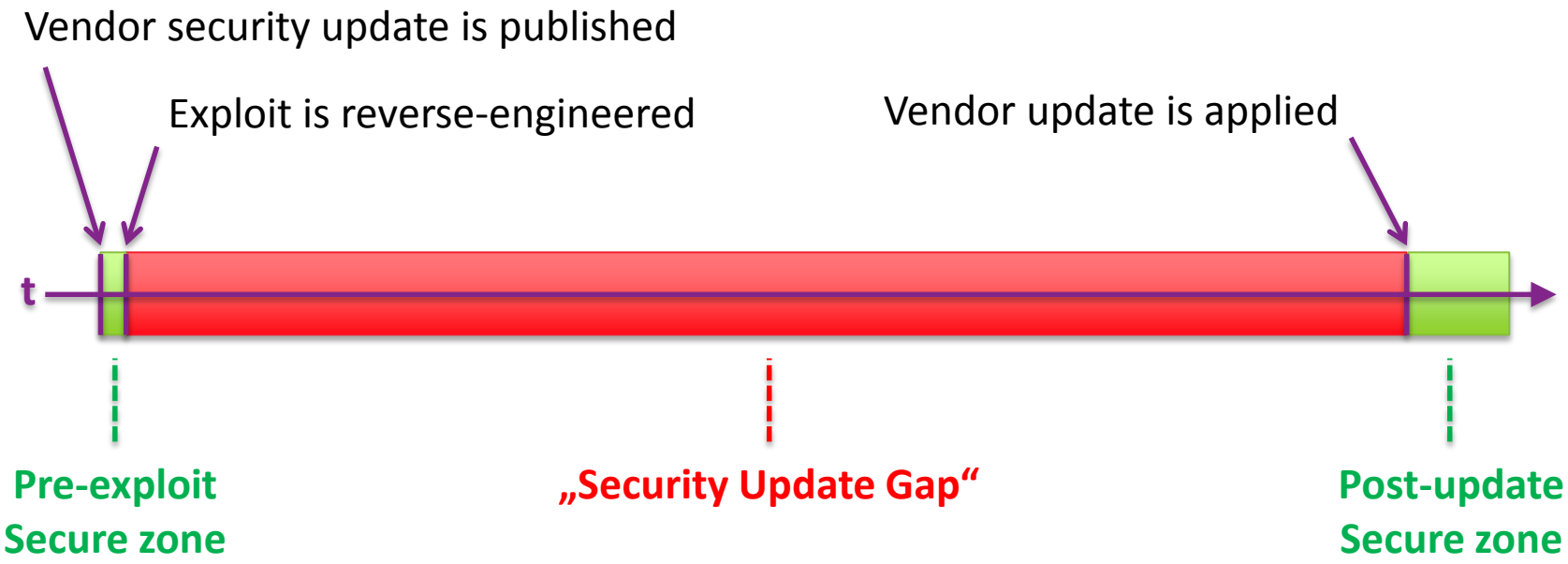

Unknown Hard Error
OK

Updates: Days from release to exploit

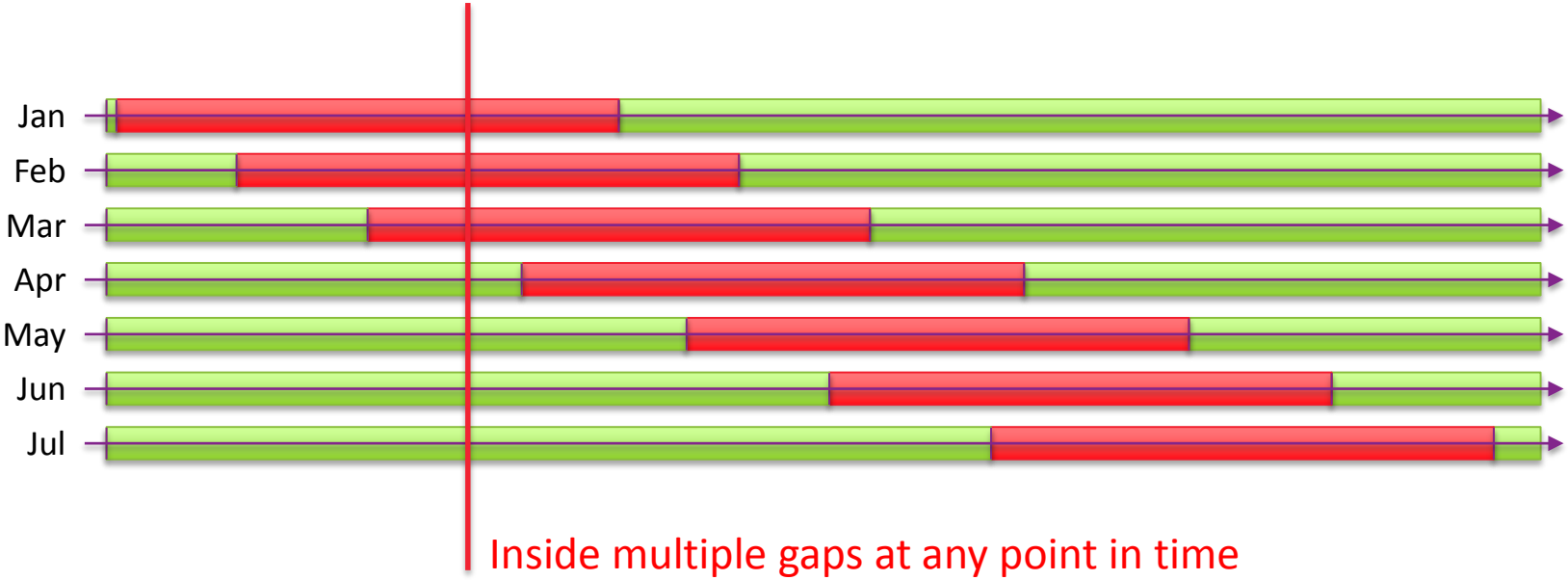
3

* Source: FireEye, Angler EK Exploiting Adobe Flash CVE-2015-0359 with CFG Bypass

Security update gap



Overlapping security update gaps



The Problem

Vulnerabilities aren't getting fixed

The main causes of this problem

- **Huge security updates** that are **risky and costly** to apply and revert (causing the security update gap)
- **Unsupported** software versions
- Software producer **does not have** a (suitable) security update process
- Software producer **does not exist** any more

We Need To…

…fix the way we fix vulnerabilities

Current state of patching

Your knee hurts?

No problem,we‘ll cut your leg off and replace it with a new one.

A different kind of patching

- **Tiny „micropatches“**
  just a few instructions to fix the vulnerability
- **Imperceptible to apply and remove**
  no restarts
- **Hot-patching**
  patching running applications without the user ever noticing
- **Hot-unpatching**
  in case something goes wrong
- **Digital microsurgery** enables **3rd party patches**

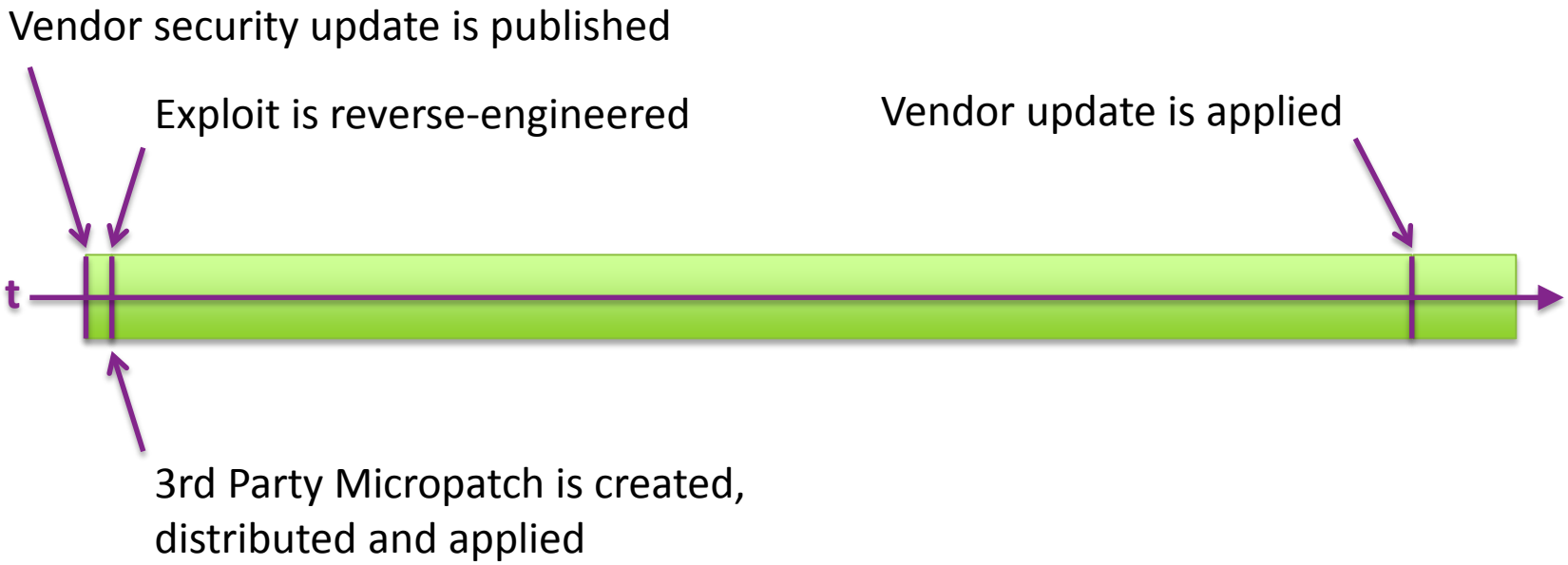
* Prior art: Determina, ZERT, eEye, PatchDroid, kSplice, ...

What types of bugs can be micropatched?

Practically all critical remote execution bugs

- Unchecked buffers
- Numeric over/underflows
- Use after free, double free
- Uninitialized variables
- Format strings
- Binary planting / DLL injection

Bridging the security update gap



Fixing unsupported software

- 3rd party patches can allow you to be safe(r) although there are no official patches for your legacy software.

Fixing security orphans

- 3rd party patches can remove vulnerabilities where the vendor is unable or unprepared to do that.
- IoCT - „Internet of crowdpatched Things“

Fixing zombies

- **3rd party patches can remove vulnerabilities in products whose vendors no longer exist.**

A lot of brain power required

- **Hunting for „proof of concept“ exploits**
  necessary to write a patch
- **Hunting for 0days**
  in malware, exploit kits, public forums
- **Analyzing open-source software**
  to create micropatches for it
- **Reverse-engineering official vendor updates**
  to create „bridge-the-gap“ replacement patches and support legacy products

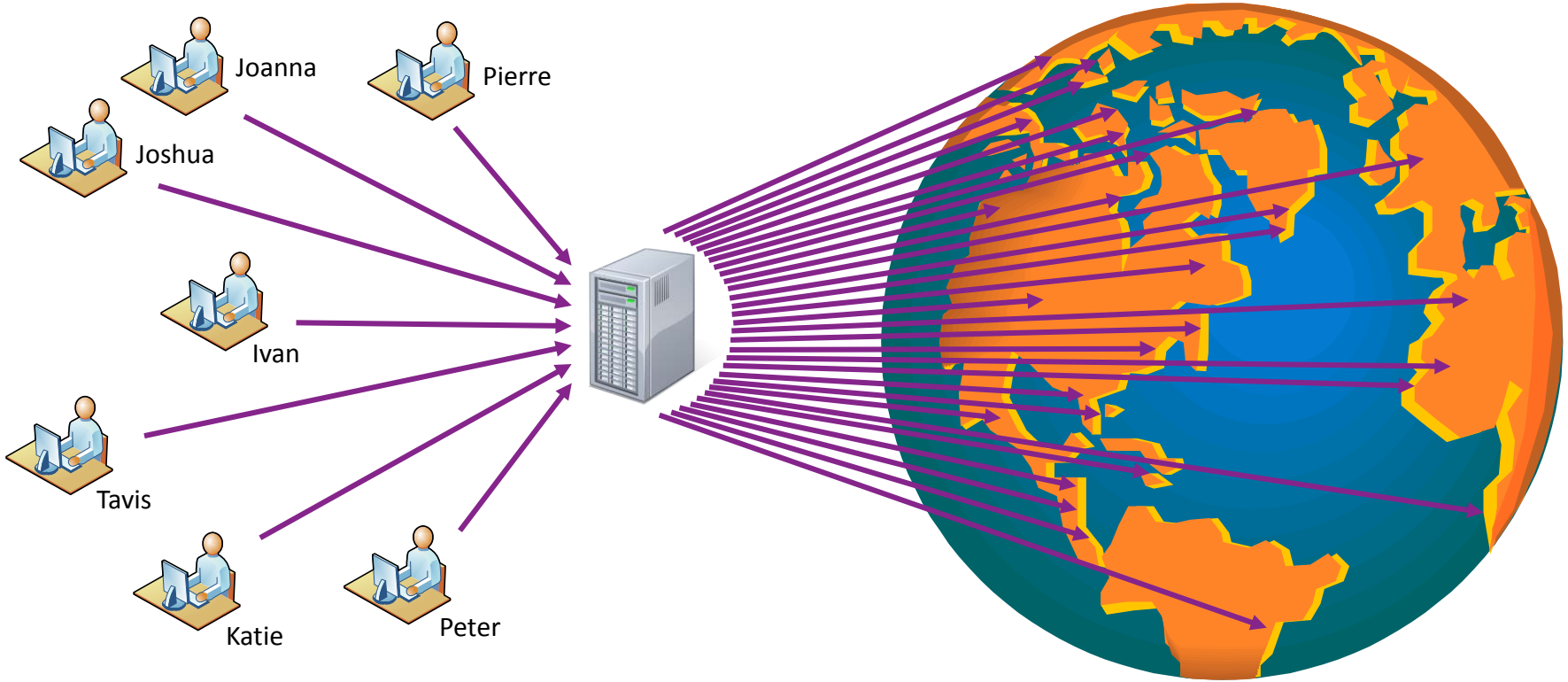
Crowdsourcing

Crowdsourcing in vulnerability discovery

- Private disclosure to vendors (for kudos)
- Bug bounties

CrowdPATCHING

Crowdpatching



Tough Questions

Reliability of crowdpatches

We need to remove the risk of breakage, remember?

- Unit testing (crowdsourced)
- Peer review (crowdsourced)
- Formal methods for validating a patch
- Telemetry
- Community feedback
- Vendor validation

Security and trustworthiness

Malicious 3rd party patch, anyone?

- It is difficult to hide malware in a 30-byte patch
- If a proposed patch is not tiny, it‘s suspect
- Peer review (crowdsourced)
- Signed by various trusted parties, you decide who to trust
- Official vendor micropatches

What will software vendors say?

- „You can‘t do that!“
- „Hmm, it‘s bad PR if we don‘t let user secure themselves.“
- „Hey, they‘re actually helping our users - kinda doing our job.“
- „Why don‘t we try micropatching ourselves?“
- „These corwdpatchers could really help us.“
- „Patch bounty!“*

* Google Patch Rewards

What YOU can do

Organizations

- Enumerate unsupported computers and apps, devices with unresponsive (or nonexisting) vendors
- Make a list of critical vulnerabilities you're exposed to right now due to the security update gap
- How long does it take you to install security updates? How many security update gaps are you in right now?
- Start bugging your software vendors to implement micropatching

Software vendors

- Consider implementing vulnerability micropatching
  - Ask your developers how they could micropatch
- Assess the benefits of:
  - avoiding out-of-band updates
  - being able to remove vulnerabilities without disturbing users
  - cheap distribution of micropatches
  - ability to quickly, cheaply revoke a micropatch if needed

Security researchers

- Start thinking about how to patch a vulnerability, not just how to exploit it
- When you write a blog about a vulnerability, also describe a micropatch for it (become part of the solution)
- Encourage your peers to do the same – creating a micropatch is quite an intellectual challenge too :)

Discussion

Mitja Kolsek

CEO, ACROS Security
Co-founder, 0patch

https://0patch.com

@mkolsek
mitja.kolsek@acrossecurity.com