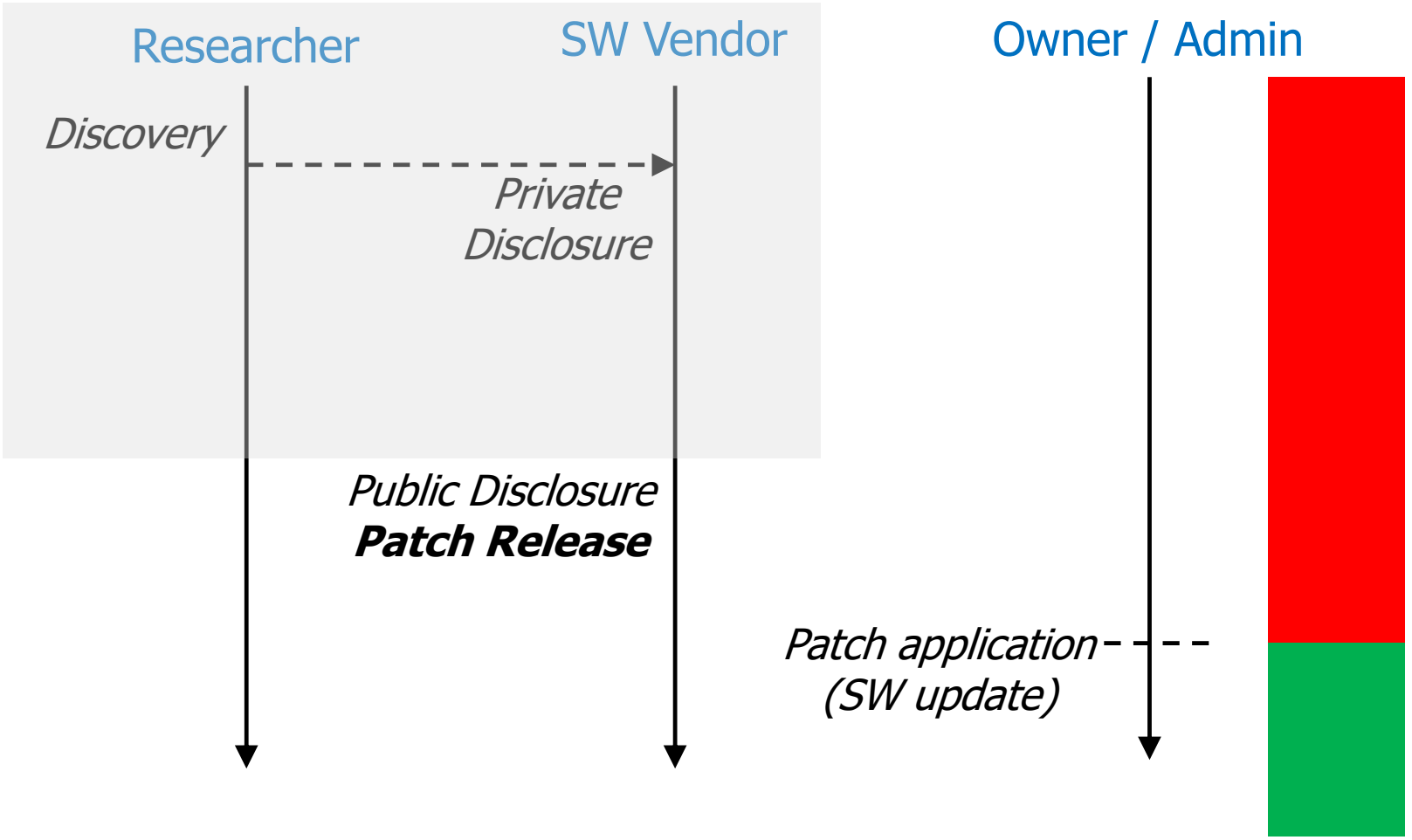
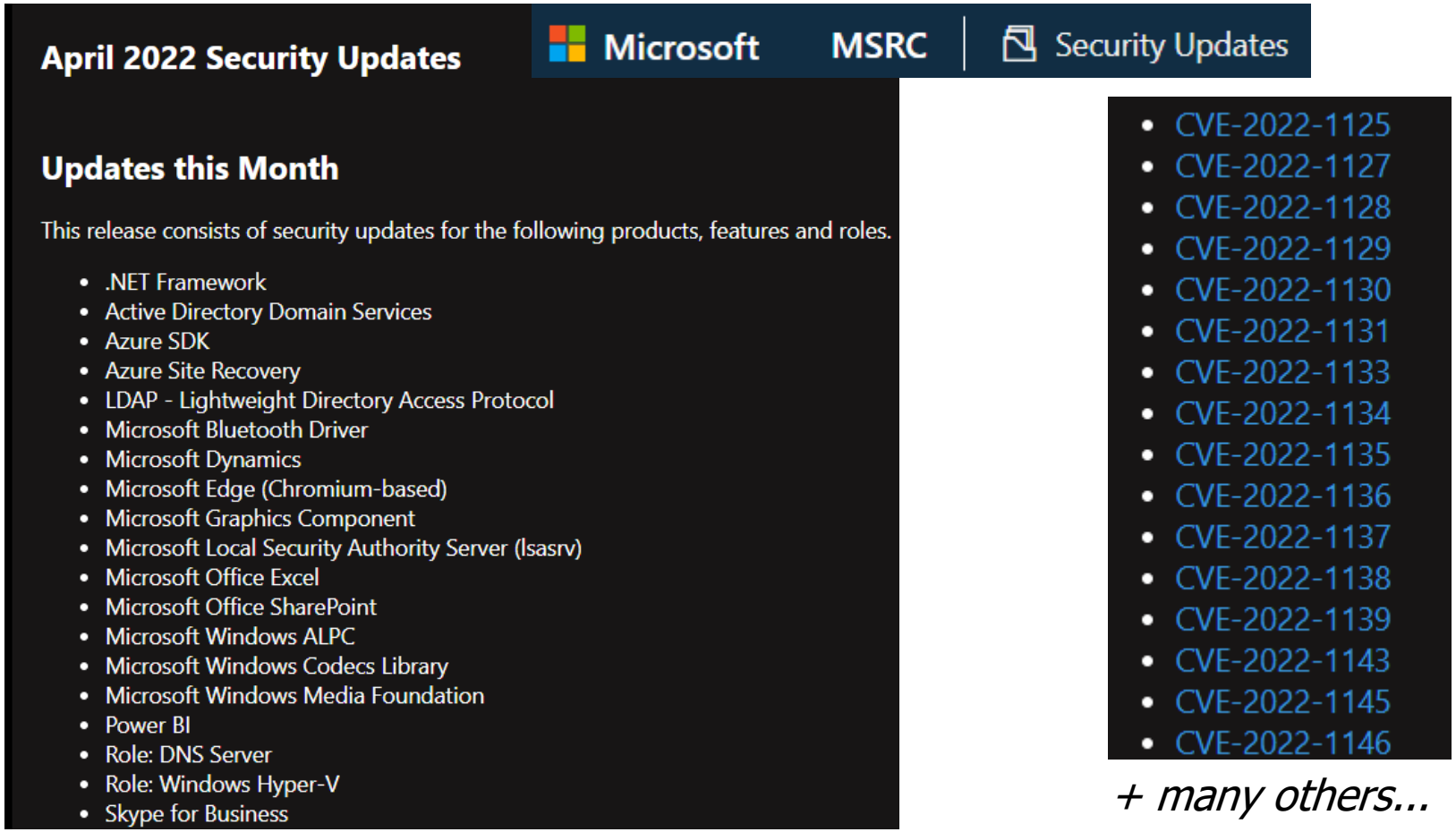
Vulnerability Lifecycle

Vulnerability Lifecycle: Ideal case



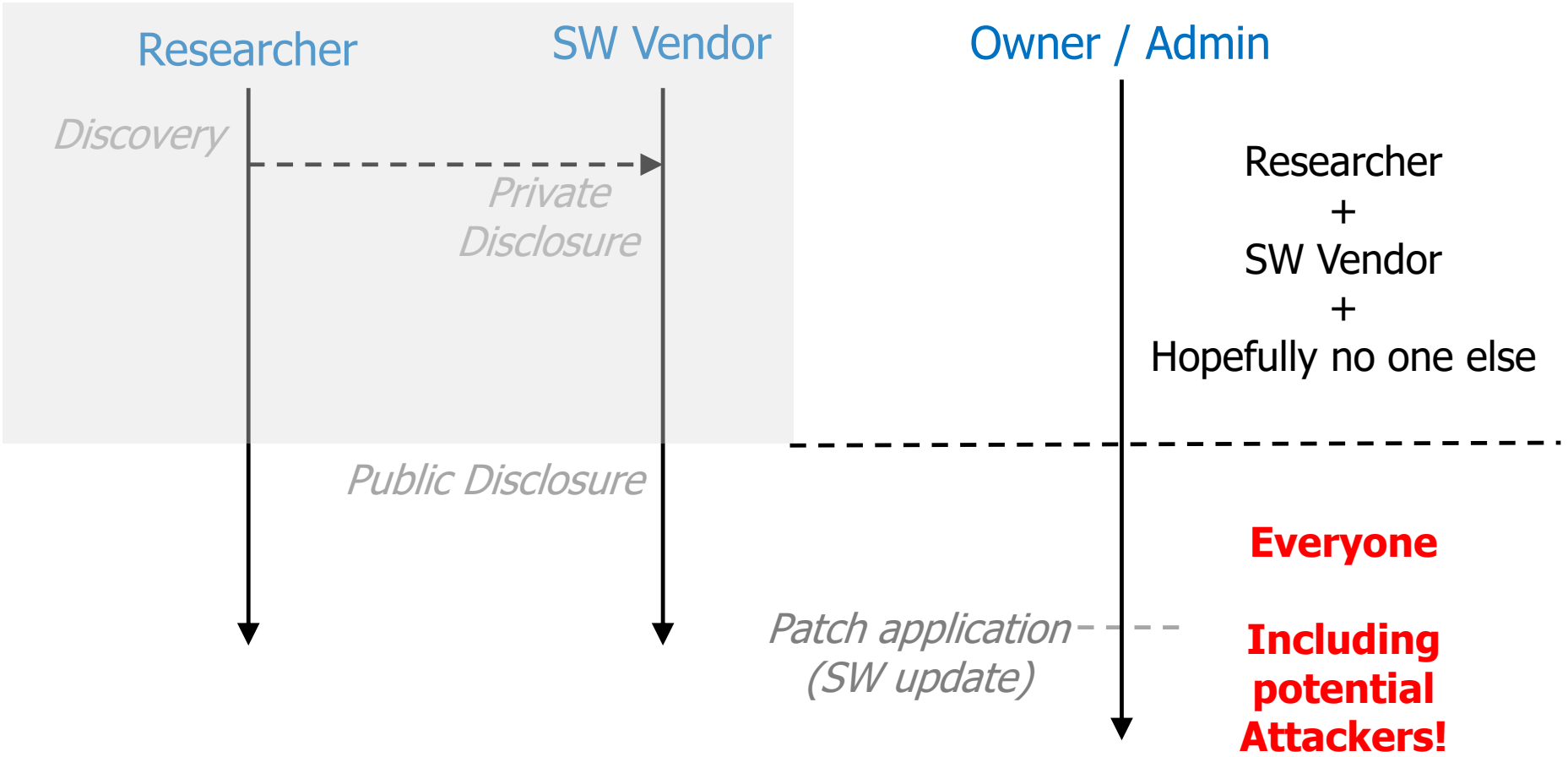
Microsoft "Patch Tuesday"



+ many others...

+ many others...

Vulnerability Knowledge



Ideal Case vs Reality

1. Researcher notifies SW Vendor
2. SW Vendor develops patch
3. Public disclosure along with patch
4. Owner/Admin applies patch

- In many cases:
  1. Researcher does **not** notify SW Vendor
  2. SW Vendor does **not** develop patch
  3. Public disclosure **without** patch available
  4. Owner/Admin does **not** apply patch

Unpatched Vulnerabilities

Ideal Case vs Reality (REMIND)

- In many cases:
  1. Researcher does **not** notify SW Vendor
  2. SW Vendor does **not** develop patch
  3. Public disclosure **without** patch available
  4. Owner/Admin does **not** apply patch

Facts about Patch Development (I)

- Patch **development** and **distribution**:
  - **Costly**
  - Occasionally **very difficult**
  - In many cases, **no contractual obligation**

- SW Vendor might not develop a patch

- Vulnerability will remain forever (or until next release, hopefully)

Facts about Patch Development (II)

- Vulnerable system has reached its **End of Life** (**EOL**)
  - Any possible contractual obligation about patches has expired

- SW Vendor **will not** develop a patch

- Vulnerability will remain **forever**

EOL Example: Struts1 (I)

Apache Struts 1 End-Of-Life (EOL) Press Release

2013-04-05 - The Apache Struts Project Team would like to inform you that the Struts 1.x web framework has reached its end of life and is no longer officially supported.

Struts 1 had its last release - version 1.3.10 - in December 2008.

In the meantime the Struts community has focused on pushing the Struts 2 framework forward, with as many as 23 releases as of this writing.

EOL Example: Struts1 (II)

Apache » Struts » 1.3.10 : Security Vulnerabilities

Cpe Name:*cpe:/a:apache:struts:1.3.10*

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication
1	CVE-2016-1182	20		DoS XSS	2016-07-04	2016-11-28	6.4	None	Remote	Low	Not required

ActionServlet.java in Apache Struts 1 1.x through 1.3.10 does not properly restrict the Validator configuration, which allows remote attackers to conduct cross-site scripting (XSS) attacks service via crafted input, a related issue to CVE-2015-0899.

2	CVE-2016-1181			DoS Exec Code	2016-07-04	2016-11-28	6.8	None	Remote	Medium	Not required

ActionServlet.java in Apache Struts 1 1.x through 1.3.10 mishandles multithreaded access to an ActionForm instance, which allows remote attackers to execute arbitrary code or cause a (unexpected memory access) via a multipart request, a related issue to CVE-2015-0899.

3	CVE-2015-0899	20		Bypass	2016-07-04	2016-11-28	5.0	None	Remote	Low	Not required

The MultiPageValidator implementation in Apache Struts 1 1.1 through 1.3.10 allows remote attackers to bypass intended access restrictions via a modified page parameter.

4	CVE-2014-0114	20		Exec Code	2014-04-30	2017-01-06	7.5	None	Remote	Low	Not required

Apache Commons BeanUtils, as distributed in lib/commons-beanutils-1.8.0.jar in Apache Struts 1.x through 1.3.10 and in other products requiring commons-beanutils through 1.9.2, doe property, which allows remote attackers to "manipulate" the ClassLoader and execute arbitrary code via the class parameter, as demonstrated by the passing of this parameter to the getC ActionForm object in Struts 1.

5	CVE-2012-1007	79		XSS	2012-02-06	2016-11-28	4.3	None	Remote	Medium	Not required

Multiple cross-site scripting (XSS) vulnerabilities in Apache Struts 1.3.10 allow remote attackers to inject arbitrary web script or HTML via (1) the name parameter to struts-examples/uplo or the message parameter to (2) struts-cookbook/processSimple.do or (3) struts-cookbook/processDyna.do.

Total number of vulnerabilities : **5** Page : 1 (This Page)

Example: Windows EOL

endoflife.date

Version	Released	Active Support	Security Support
Windows 11, version 21H2 (E)	6 months ago (04 Oct 2021)	Ends in 2 years and 6 months (08 Oct 2024)	Ends in 2 years and 6 months (08 Oct 2024)
Windows 10, version 21H1 (E)(W)	10 months ago (18 May 2021)	Ends in 8 months (13 Dec 2022)	Ends in 8 months (13 Dec 2022)
Windows 10, version 20H2 (W)	1 year and 5 months ago (20 Oct 2020)	Ends in 1 month and 6 days (10 May 2022)	Ends in 1 month and 6 days (10 May 2022)
Windows 10, version 20H2 (E)	1 year and 5 months ago (20 Oct 2020)	Ends in 1 year (09 May 2023)	Ends in 1 year (09 May 2023)
Windows 10, version 2004 (E)(W)	1 year and 10 months ago (27 May 2020)	Ended 3 months and 3 weeks ago (14 Dec 2021)	Ended 3 months and 3 weeks ago (14 Dec 2021)
Windows 10, version 1909 (W)	2 years and 4 months ago (12 Nov 2019)	Ended 10 months ago (11 May 2021)	Ended 10 months ago (11 May 2021)
Windows 10, version 1909 (E)	2 years and 4 months ago (12 Nov 2019)	Ends in 1 month and 6 days (10 May 2022)	Ends in 1 month and 6 days (10 May 2022)

EOL Consequence: Wannacry (May 2017)

Microsoft Security Bulletin MS17-010 - Critical

Security Update for Microsoft Windows SMB Server (4013389)

Published: March 14, 2017

- 300.000 infections
- Infected systems:
  1. EOL, or
  2. Had not applied a 2-months old patch

Ideal Case vs Reality (REMIND)

- In many cases:
  1. Researcher does **not** notify SW Vendor
  2. SW Vendor does **not** develop patch
  3. Public disclosure **without** patch available
  4. Owner/Admin does **not** apply patch

Practical (and Very Important) Issue

- You discover a vuln and report it to the SW Vendor
- SW Vendor downplays relevance and **does not act**
- **What do you do?**

- All instances are vulnerable…but **their owners do not know**!
- An Attacker might discover and exploit that vuln!

Vulnerability Discovery: Key Approaches (I)

1. Make vuln info **public**
   - Defenders are alerted
   - Public pressure on vendors may be the only thing that can work
   - Attackers are alerted as well
2. ...along with a Proof Of Concept (**PoC**) **exploit**
   - Stronger public pressure on vendors

- Potentially dangerous for the community

Vulnerability Discovery: Key Approaches (II)

1. Make vuln info **public**
2. Make vuln info **public** along with a **PoC**
3. **Responsible Disclosure**: Give a "reasonable deadline" to the SW vendor for acting before going public

- Considered the **best option**

Example: Google Project Zero

+14 days

Deadline adherence and fix time 2019-2021, by bug report volume

Vendor	Total bugs	Fixed by day 90	Fixed during grace period	Exceeded deadline & grace period	Avg days to fix
Apple	84	73 (87%)	7 (8%)	4 (5%)	69
Microsoft	80	61 (76%)	15 (19%)	4 (5%)	83
Google	56	53 (95%)	2 (4%)	1 (2%)	44
Linux	25	24 (96%)	0 (0%)	1 (4%)	25
Adobe	19	15 (79%)	4 (21%)	0 (0%)	65
Mozilla	10	9 (90%)	1 (10%)	0 (0%)	46
Samsung	10	8 (80%)	2 (20%)	0 (0%)	72
Oracle	7	3 (43%)	0 (0%)	4 (57%)	109
Others*	55	48 (87%)	3 (5%)	4 (7%)	44
TOTAL	346	294 (84%)	34 (10%)	18 (5%)	61

```
search "project zero deadline"
```

Remark

1. Make vuln info **public**
2. Make vuln info **public** along with a **PoC**
3. **Responsible Disclosure**
4. Keep vuln **secret** and **forget** about it
   - You do not loose time and do not fear legal actions

- Potentially dangerous for the community
- ...but sometimes more than justified in practice
  - See the course website

Emergency Disclosure (I)

- You discover a vuln and report it to the SW Vendor
- Developing and distributing a patch takes **time**
- The nature of the vulnerability creates a **huge risk** for the community

- You agree to a public disclosure **before the patch** so that Defenders become aware of the problem

- Difficult decision: potential Attackers become aware of the problem as well

Emergency Disclosure (II)

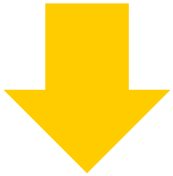
- You discover a vuln while **investigating network traffic** (or a cybersecurity incident)
- You agree to a public disclosure **before the patch** so that Defenders become aware of the problem
- Decision much easier: there is exploitation evidence already

Another Practical Issue

- You discover a vuln and decide to disclose it publicly **at a conference** before reporting it to the SW Vendor
- Public disclosure **before the patch**
- Hopefully without too many details
- Financial and personal incentives are crucial

Unpatched vulnerability

- Public disclosure **without** patch available
  - Vendors does not want to develop a patch
  - EOL
  - Responsible disclosure (and conferences)
  - Emergency disclosure



- A **publicly known** vulnerability may remain **unpatched** for some time / forever

What to do? (I)

1. Analyze vulnerability
   - Assess injection, impact, existence of exploits, ...
   - Assess **contextual risk**

2. Take action on the vulnerable system
   - **Increase monitoring** (and inform SIEM)
   - **Limit network exposure**
   - Pull the plug (temporarily)
   - Do nothing and accept the risk

What to do? (II)

1. Analyze vulnerability
   - Assess injection, impact, existence of exploits, ...
   - Assess **contextual risk**

- EOL + high contextual risk may call for a **dismission plan**

Patch Management

Ideal Case vs Reality (REMIND)

- In many cases:
  1. Researcher does **not** notify SW Vendor
  2. SW Vendor does **not** develop patch
  3. Public disclosure **without** patch available
  4. Owner/Admin does **not** apply patch

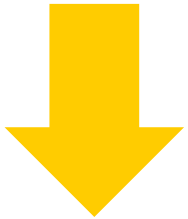
Facts about Patch Application (I)

- Applying a patch:
  - May be **complex** and **time-consuming**
  - Usually requires **service downtime**
  - May create **compatibility problems** on other systems

- **Not applying / delaying** the patch (and accepting the risk) may be the **most rational** choice

Facts about Patch Application (II)

- System owner / administrator often **is not even aware of**:
  - The specific vulnerability
  - **Which systems** are affected
  - **Who is in charge** of them
    (more details later)



- Very bad approach: you **do not even know** that you are **taking a risk**

Zero-days

Ideal Case vs Reality (REMIND)

- In many cases:
  1. Researcher does **not** notify SW Vendor
  2. SW Vendor does **not** develop patch
  3. Public disclosure **without** patch available
  4. Owner/Admin does **not** apply patch

Hhmmm…

- I have discovered an interesting exploitable vulnerability
- Why report it to the SW Vendor?
- I can **sell** vuln info to organizations interested in exploiting it!

- Illegal and **Legal** markets

Vulnerability Discovery: Key Approaches (IV)

1. Make vuln info **public**
2. Make vuln info **public** along with a **PoC**
3. **Responsible Disclosure**
4. Keep vuln **secret** and **forget** about it
5. Keep vuln **secret** and **sell info/exploit (!)**

"Zero-day" vulnerability

- **Exploited** while **unknown** to the Manufacturer (thus to **everyone else**)
- As long as injection succeeds...bingo!

Bought in **legal** markets	Discovered by Attackers (criminals, intelligence services) Bought in illegal markets

0-day vulns

- **Real problem**
- Magnitude unclear

Legal Market (I)

- Buys zero-day exploits from researchers
- Sells them to "selected organizations"

As part of this program, Crowdfense exclusively evaluates **fully functional, high-quality zero-day exploits** targeting the platforms and products listed below.

Crowdfense supplies international institutional customers, including governments, intelligence and law enforcement agencies (LEAs), and trusted system integrators with a **reliable stream of high-impact cyber capabilities**.

Access to our Client portal is available only to a carefully vetted group of Customers.

Legal Market (II)

- Mobile - up to 7M USD
- Desktop - up to 1,5M USD
- Virtualization - up to 500k USD
- Appliances & Peripheral Devices - up to 100k USD
- Enterprise - up to 500k USD
- Web Apps - up to 500k USD

! High demand

Zero Click Full Chains

- **Android Zero Click Full Chain (e.g Whatsapp, RCS)**: 5 M USD
- **iOS Zero Click Full Chain (e.g. iMessage)**: from 5 to 7 M USD

Browsers

- **Chrome (RCE + LPE)**: from 2 to 3 M USD
- **Chrome (RCE w/o SBX)**: 500k USD
- **Chrome (SBX)**: 500k USD
- **Safari (RCE + LPE)**: from 2,5 to 3,5 M USD
- **Safari (RCE w/o SBX)**: 500k USD
- **Safari (SBX)**: from 300 to 400k USD

Some 0-day stats: Legal Market

- 2004-2016 data from selected 0-day vendors

- **207** zero-days **with exploits**
- 50% **publicly unknown**
- Average life expectancy **6.9 years**

- Only 25 percent live **less** than 1.51 years
- Only 25 percent live **more** than 9.5 years.

Zero Days and Thousands of Nights
The Life and Times of Zero-day Vulnerabilities and their Exploits
RAND - 2017

Some 0-day stats: Origin unknown

- 2014-today maintained by Google Project Zero

- Public spreadsheet tracks cases of **publicly discovered** zero-days
  - Detected exploitations associated with a vulnerability
  - Association made **after** the detection
  - Vulns not publicly known at the time of detection

- April 5 2022: 211
- August 9 2023: 284

Understanding zero-days

A worrying scenario

- Population of experts **incentivized** in:
  - Discovery of exploitable vuln
  - **Keep them hidden to Vendors**
  - Diffuse knowledge in restricted circles

- Users/Organizations have more and more:
  - Vulns exploitable for a long time
  - **They do not even know about…**
  - **…but several restricted circles do**

Not to be overestimated (I)

- Attackers attempt to **avoid** usage of 0-day whenever possible

- High effectiveness
- High economic value
- **Discovery by defenders ⇒ significant loss** (no longer a zero-day)

- Minimizing the risk of its discovery is crucial

Not to be overestimated (II)

- Increasing evidence **evidence of exploitation before public disclosure**

- 1-H 2025: ≈**1/3 KEV** evidence of exploitation before CVE publication
- 1-H 2025: less than 1/4

Source: VulnCheck

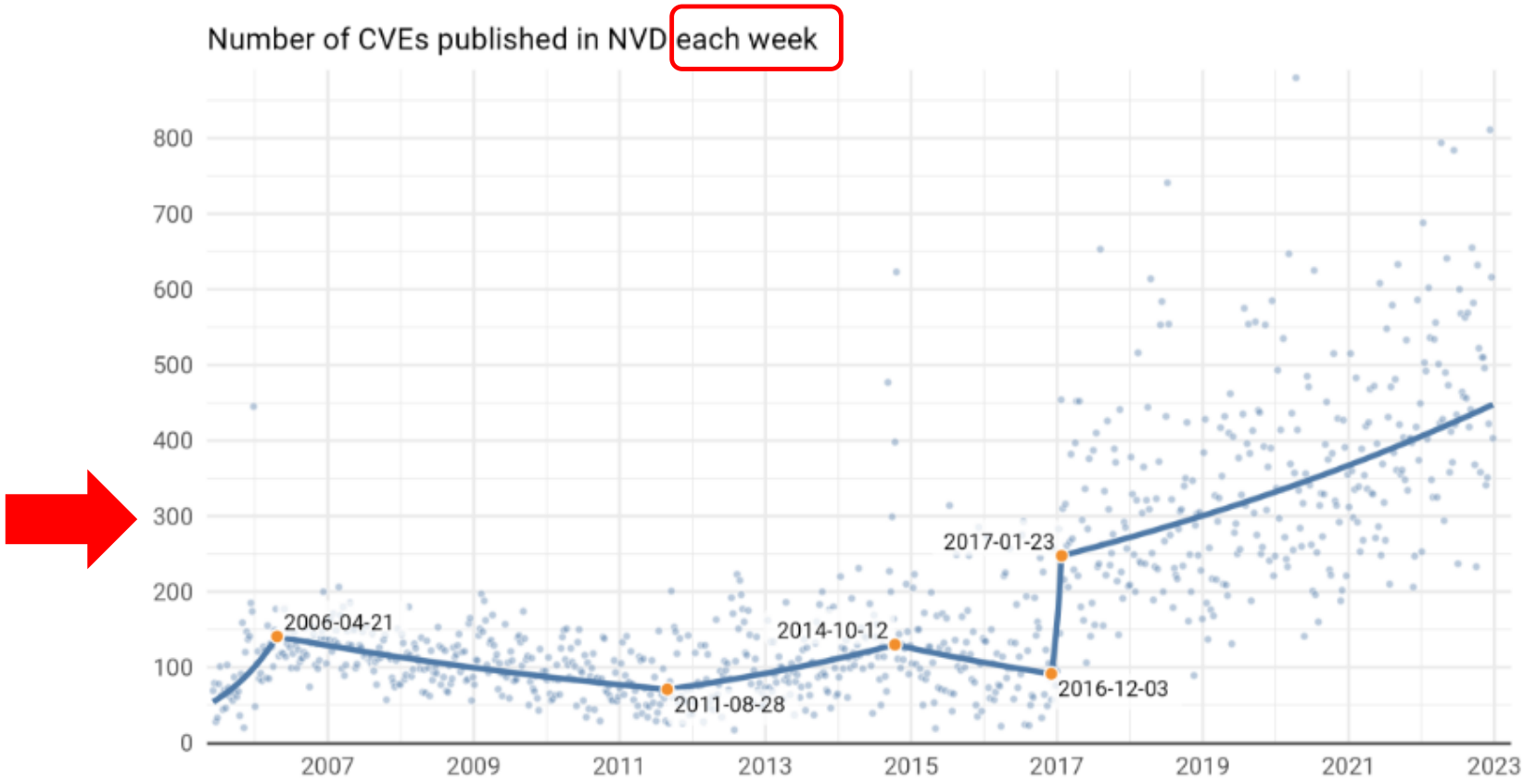
- Mostly because of plain criminal organizations
- Not secret services operating on obscure markets...

A difficult strategic problem

- You are a National Intelligence Service
- You discover or are proposed to acquire:
  - High impact vuln
  - Used by many organizations or by critical infrastructure
- Do you keep it secret or not?
  - You can use it for attacking **enemy** organizations
  - ...but you leave **your** organizations vulnerable
- What is the national interest?

Vulnerability Management in Organizations

New CVEs


Number of CVEs published in NVD each week
800
700
600
500
400
300
200
100
0
2007
2009
2011
2013
2015
2017
2019
2021
2023
2006-04-21
2011-08-28
2014-10-12
2016-12-03
2017-01-23

Fact about Vulnerabilities

- In practice, in **every** organization there are always **a lot** of vulnerabilities to worry about

Basic Fact #1 (REMIND)

- **Very few CVEs are actually exploited**
- Just to have an idea: **≈5%** of all CVEs **(!)**

Focus on those!

Basic Fact #2 (REMIND)

- CVSS is **not** a good predictor of which vulnerabilities will be **actually exploited**
- Predicting which vulnerabilities will be exploited is a **huge open problem**

- Every predictor you can think of turns out to be
  - "Low" precision:
    You worry about many vulns **unnecessarily**
  - "Low" recall:
    You wrongly **neglect** many vulns

Fact of life

- Vulnerabilities (and their associated risks):
  - **Cannot be eliminated**
  - Must be **managed**

How?

- Vulnerabilities (and their associated risks):
  - **Cannot be eliminated**
  - Must be **managed**

Common (Wrong) Approach

- **Surrender** and **cross the fingers**
- Patch (without any hurry) **extremely critical** vulns that emerge every now and then

- Many cases of total catastrophe ultimately caused by "evident" vulnerabilities "easy to patch"

Correct Approach

- Aim for **systematic** and **constant** reduction of risk, to the extent feasible with the **defensive budget** available

- Establish a **structured process** for vuln **management**
  - "Process" ≠ **Not** a one-off event
  - "Management" ≠ **Not** elimination

- Basic requirement:
  - **Sufficient human resources** allocated to this process

Vulnerability Management in Organizations (I)

- Vulnerability **Management**:
  - Who is on charge of collecting threat intelligence
  - How much effort to allocate
  - How to coordinate effort (planning, scheduling)
  - How to plan and manage downtime
  - …
- Out of scope (but **no general recipe**)

Vulnerability Management in Organizations (II)

- Vulnerability **Management**:
  - …

- Fundamental components
  - **Asset Management**
    - Which systems
    - Who is in charge
  - **Vulnerability Prioritization**
    - How to allocate defensive efforts to CVEs

Keep in mind (I)

- **No organization** patches a **significant fraction** of its vulnerabilities
- **No organization** applies patches **extremely quickly**

- Many studies:
  - **≈15%** of internal vulns patched…within **a month** from the available patch
  - Median patch rate per month: 15% of open vulns (one quarter of the orgs: 6% of open vulns)

Keep in mind (II)

- Special attention to vulns in:
  1. Devices exposed on the Internet
  2. Remote injection

- **Automated** campaigns on the **entire Internet** aimed at obtaining just **persistence** are both **feasible** and **likely**
- More details later

Keep in mind (III)

- Vulnerabilities are indeed a major issue in cybersecurity
- …but **do not overestimate** their importance

- **Credentials** are **less fascinating** and **more boring**
- …but they are **equally important** to Adversaries (if not more)

Valid Accounts (REMIND)

- Adversaries may obtain and abuse credentials of existing accounts as a means of gaining **Initial Access**, **Persistence**, **Privilege Escalation**, or **Defense Evasion**.
- Compromised credentials may be used to **bypass access controls** placed on various resources on systems within the network and may even be used for **persistent access to remote systems** and **externally available services**, such as VPNs, Outlook Web Access, network devices, and remote desktop.
- Compromised credentials may also grant an adversary **increased privilege to specific systems** or **access to restricted areas** of the network.
- Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence.

Asset Management

Asset Management: What is it? (I)

- **Accurate description of systems**
  - + software, versions, **known vulnerabilities**
  - + whether exposed **to the Internet**
  - + managed by whom

- **"Continuous** and **comprehensive** asset visibility is a **basic pre-condition for any organization** to effectively manage cybersecurity risk"

  (Dept. for Homeland Security, October 2022)

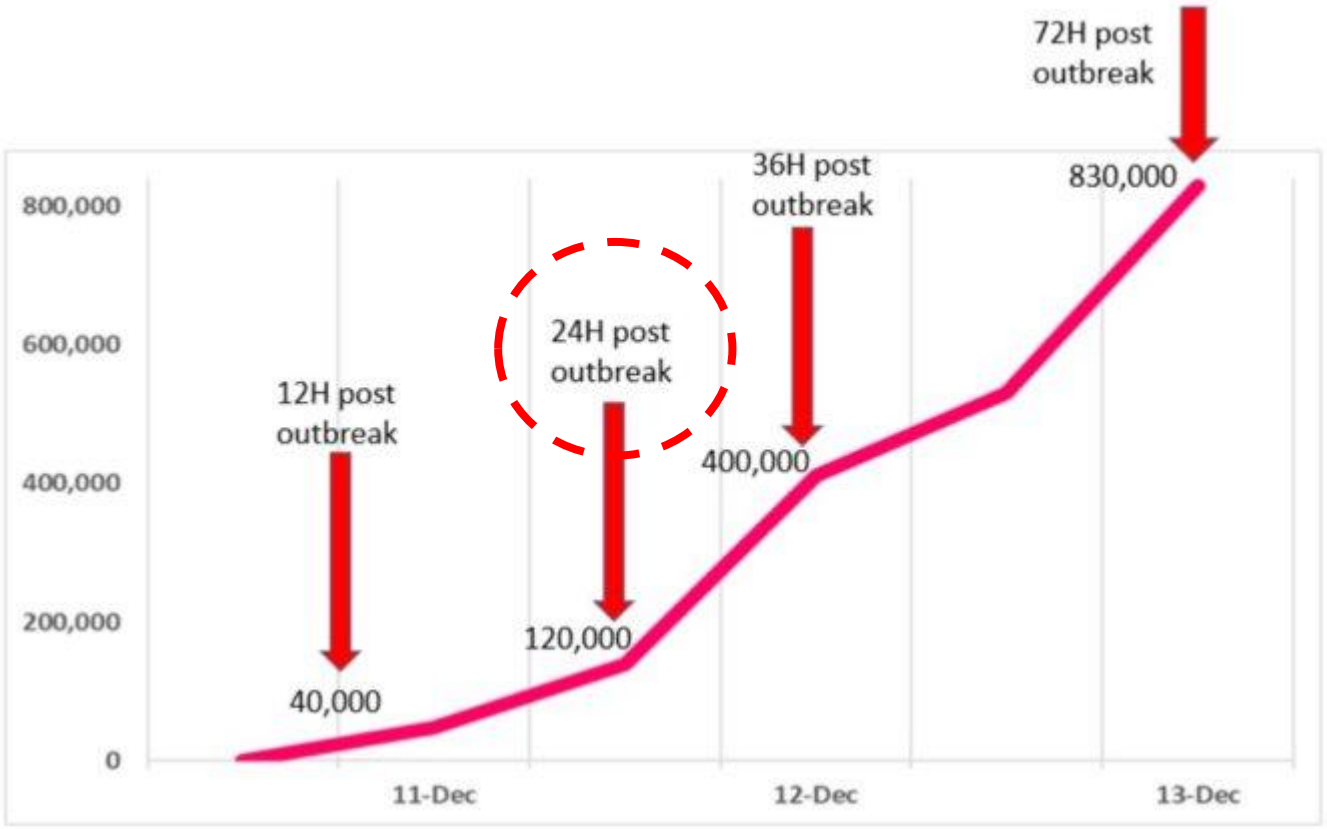
Asset Management: What is it? (II)

- "It is **not** about making lists or databases that **never get used**."

- "It means creating and **maintaining** authoritative and **accurate** information
  - **Day-to-day operations**
  - Efficient decision making **when you need them most**."

(NCSC UK)

"When you need the most": Log4j (December 2021)



SATURDAY

https://blog.checkpoint.com/2021/12/13/the-numbers-behind-a-cyber-pandemic-detailed-dive/

Asset Management: DEFENSIVE CORNERSTONE

- "**Continuous** and **comprehensive** asset visibility is a **basic pre-condition for any organization** to effectively manage cybersecurity risk"
- **Not** only for vuln management
- **How can you defend effectively if you do not even know what you are defending???**
  - E.g., what if a password has been stolen?

Many tool families

- **Asset** discovery / management
- **Vulnerability** scanners
  - Probe devices against (subset of) known vulns
- **Internet exposure** monitor
- **Attack surface** discovery / management

Keep in mind

- "**Continuous** and **comprehensive** asset visibility is a **basic pre-condition** **for any organization** to effectively manage cybersecurity risk"

- AI-powered-whatever
- Crypto-whatever
- …
- Almost pointless without effective **Asset Management**

Maturity test for organization

- Ask CTO to provide the above description
- You should not be waiting for **more than 30 minutes** to obtain a sufficiently accurate answer

Much easier said than done (I)

October 3, 2022

- *A binding operational directive is a **compulsory** direction to federal, executive branch, departments and agencies*

Much easier said than done (II)

- **By April 3, 2023**:
  1. Perform **automated asset discovery every 7 days**.
  2. Initiate **vulnerability enumeration** across all discovered assets, including all discovered nomadic/roaming devices (e.g., laptops), **every 14 days**.
  3. …

Practical Guidance

Asset management

1. Introduction
2. What is an asset?
3. The importance of asset management
4. Integrating asset management into your organisation
5. What should be included in a good asset management approach
6. Data sources
7. Validating the Asset Management system

Automated Attacks

Human-operated

- Initial Access
- Execution
- Persistence
- C&C
- Discovery
- Lateral movement
- Impact

- Human operators execute **all the steps**
- Actions can be **tailored** to the **specific** environment
- It can be very effective

Automated

- Initial Access
- Execution
- Persistence
- C&C
- Discovery
- Lateral movement
- Impact

- **Automated** tool executes **all the steps**
- Actions **cannot** be **tailored** to the **specific** environment
- Hardly very effective
  - Circumventing a broad variety of different defenses and environments with **one** tool?

Human-operated vs Automated

- Human-operated attacks are usually **very effective**
  - **Extremely** dangerous
  - Costly ⇒ Less frequent
- Automated attacks are usually **not** very effective
  - Much less dangerous
  - Cheap ⇒ Quite frequent
- We will analyze attack economics (categories) later

Automated Attacks

- Automated attacks are usually **not** very effective
  - Much less dangerous
  - Cheap ⇒ Quite frequent
- **Sometimes** they are **effective!**
- Attack tool indeed able to execute all the steps:
  - Automatically
  - With **high probability of success**
  - Quickly

- HUGE problems:
  - **Fast** propagation **across** and **within** organizations

A few examples

- Petya (2017)
  - Example of **high effectiveness** in **automation**
- NotPetya@Maersk (2017)
  - Example of **fast propagation within** organization
- WannaCry (2017)
  - Example of **fast propagation across** different organizations
- NotPetya (2017)
  - Example of **high damage** done

- References in companion website

Persistence

- Previous examples: attacks with **visible impact**
- Attacks may aim at **hidden** initial access + **persistence**
  - Backdoors for later **human-operated** exploitation

- Attack tool indeed able to execute all the steps:
  - Automatically
  - With **high probability of success**
  - Quickly

⇒ May be a **critical issue worldwide**
even for **national security**

Fact

- Exploit E with injection Remote / No user action
- Injection attempt on the **entire** IPv4 space:
  - Feasible in **less than 10 minutes**

- The predominant cost is **for developing E**
- The injection attempt on the **entire** IPv4 space **costs almost nothing**

Consequence

- Device D:
  - Public IP address and **exposed on the Internet**
  - You become aware that D has:
    - RCE vuln
    - Exploitable Remote / No user action
    - Cheap and reliable exploit

- You should assume that the D is **already under the control of an attacker**

Why?

- You should assume that the device is **already under the control of an attacker**

- Public vuln:
  - Attacker has to develop exploit and start injection scan
  - Usually a few days
  - Defender process is **certainly** much longer

- Zero day:
  - Full injection scan has probably occurred already

"High risk" Organizations

- You should assume that the device is **already under the control of an attacker**

- Depending on the risk for your organization, the most sensible action may be **disconnect D immediately**
  - …**irrespective of the operational costs**

Example (I)

CVE-2023-46805 Detail

NVD Published Date:
01/12/2024

Description

An authentication bypass vulnerability in the web component of Ivanti ICS 9.x, 22.x and Ivanti Policy Secure allows a remote attacker to access restricted resources by bypassing control checks.

CVE-2024-21887 Detail

NVD Published Date:
01/12/2024

Description

A command injection vulnerability in web components of Ivanti Connect Secure (9.x, 22.x) and Ivanti Policy Secure (9.x, 22.x) allows an authenticated administrator to send specially crafted requests and execute arbitrary commands on the appliance.

Example (II-a)

- CISA has observed **widespread** and **active exploitation …** these conditions pose an unacceptable risk … and require emergency action.
- This determination is based on:
  - the **high potential for a compromise;**
  - the **impact** of a successful compromise;
  - the **complexity** of the proposed **mitigations.**

Example (II-b)

January 31, 2024

Required Actions

Agencies running affected products—Ivanti Connect Secure or Ivanti Policy Secure solutions—are required to immediately perform the following tasks:

1. As soon as possible and no later than 11:59PM on Friday February 2, 2024, disconnect all instances of Ivanti Connect Secure and Ivanti Policy Secure solution products from agency networks.

...not enough: you then have to "Evict"

1. As soon as possible and no later than 11:59PM on Friday February 2, 2024, disconnect all instances of Ivanti Connect Secure and Ivanti Policy Secure solution products from agency networks.

   a. Continue threat hunting on any systems connected to—or recently connected to—the affected Ivanti device.

   ...

2. To bring a product back into service, agencies are required to perform the following actions:

   a. Export configuration settings.

   b. Complete a factory reset per Ivanti's instructions .

   c. Rebuild the device per Ivanti's instructions AND upgrade to one of the following supported software versions through Ivanti's download portal (there is no cost to upgrade):

   ...

BIG Example: SolarWinds

ED 21-01: Mitigate SolarWinds Orion Code Compromise

CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY

December 13, 2020

2. Affected agencies shall immediately **disconnect or power down SolarWinds Orion products, versions 2019.4 through 2020.2.1 HF1, from their network**. Until such time as CISA directs affected entities to rebuild the Windows operating system and reinstall the SolarWinds software package, **agencies are prohibited from (re)joining the Windows host OS to the enterprise domain**. Affected entities should expect further communications from CISA and await guidance before rebuilding from trusted sources utilizing the latest version of the product available.

(more info on course website)

"Very High Risk" Organizations

- Depending on the risk for your organization, the most sensible action may be **disconnect D immediately**
  - **…irrespective of the operational costs**

- Any organization with very tight security requirements should have "**people able to pull the plug purely on their own authority**"
  - Software company that develops security tools
  - Highly-sensitive government agencies
  - Control room of banks
  - …
- Risk/Cost analyses require bypassing of normal processes