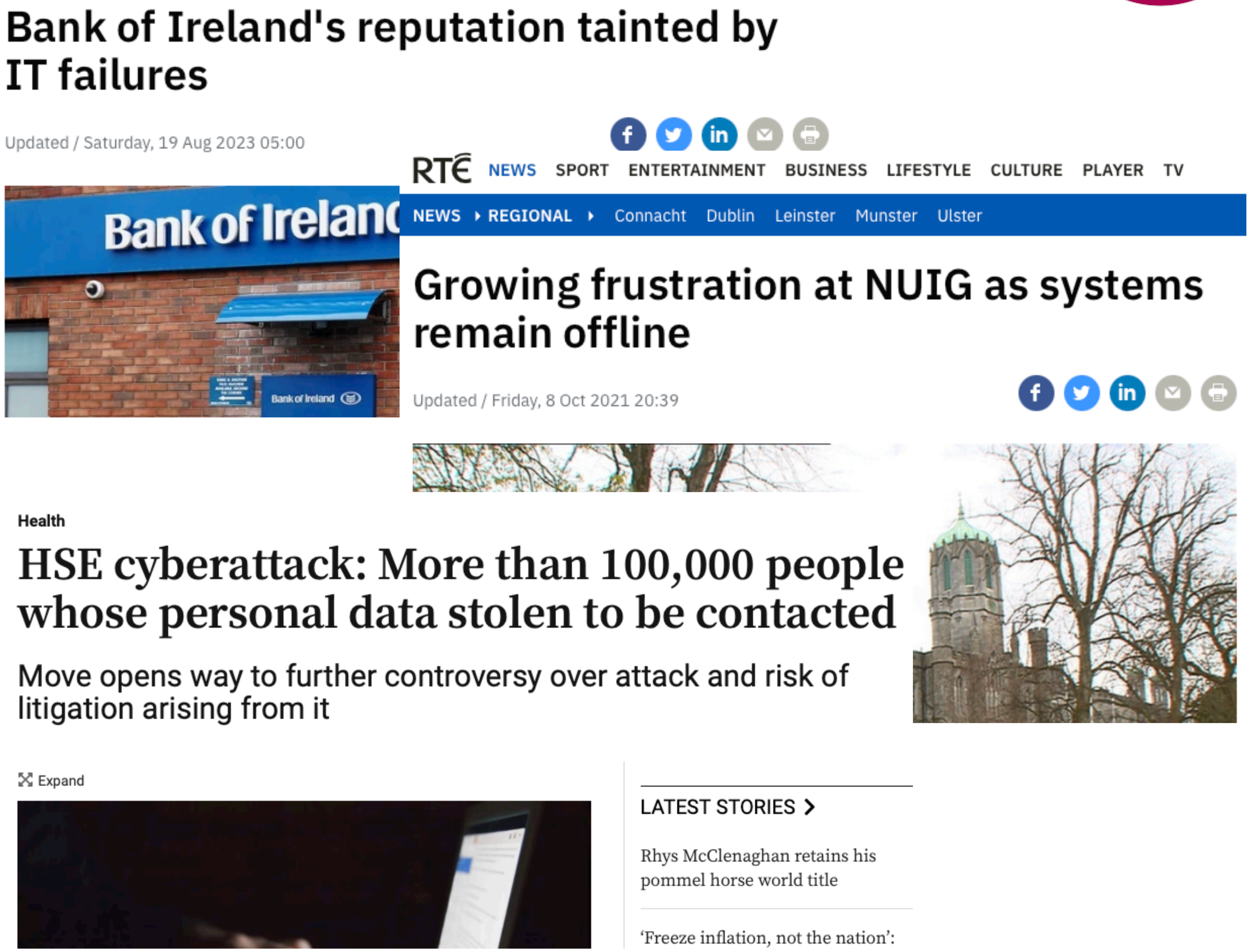
Outline

Planned topics for this lesson:

- To get a solid understanding of the underlying problems of insecure software
- To gain a practical foundation of secure / resilient software development techniques
- To get some hands-on experience to probe software for vulnerabilities



Bank of Ireland's reputation tainted by IT failures

Updated / Saturday, 19 Aug 2023 05:00

Bank of Ireland
Bank of Ireland
RTÉ NEWS SPORT ENTERTAINMENT BUSINESS LIFESTYLE CULTURE PLAYER TV
NEWS › REGIONAL › Connacht Dublin Leinster Munster Ulster

Growing frustration at NUIG as systems remain offline

Updated / Friday, 8 Oct 2021 20:39

Health

HSE cyberattack: More than 100,000 people whose personal data stolen to be contacted

Move opens way to further controversy over attack and risk of litigation arising from it

Rhys McClenaghan retains his pommel horse world title

'Freeze inflation, not the nation':

What is Software Security?

- Software security is the concept of implementing mechanisms and adopting best development practices to protect software against malicious attacks (i.e., to make it resistant to attacks, and keep it functional when attacked).

→ IT IS A CONCERN, NOT A FEATURE

- In a traditional software design and development practices, software security was almost an afterthought

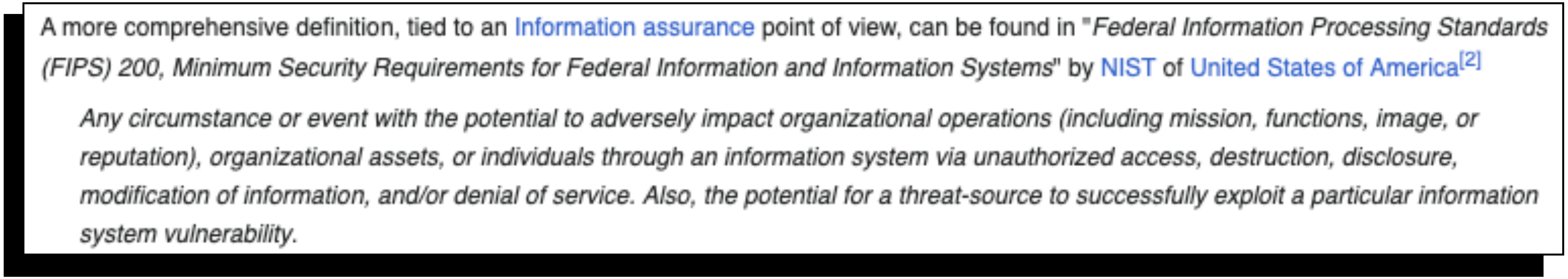
→ PROVIDING HOLISTIC SOFTWARE SECURITY IS A DIFFICULT AND TEDIOUS TASK

- Secure software is defined as software engineered in such a way that its operation and functionality continues as normal even when subjected to malicious attacks

What is a Threat?

- In computer security, a threat is a potential negative action or event facilitated by a vulnerability that results in an unwanted impact to a computer system or application

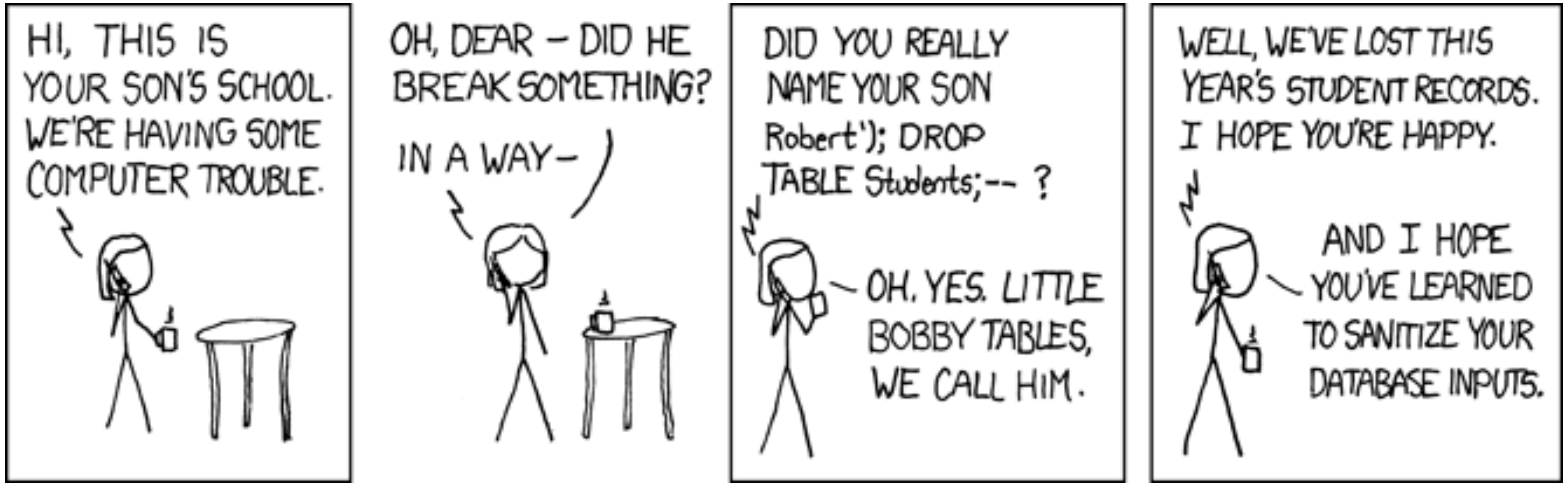
- A threat can be either:
  - a negative “intentional” event (i.e., cyberattack) or
  - an “accidental” negative event (e.g., earthquake)



A more comprehensive definition, tied to an Information assurance point of view, can be found in "*Federal Information Processing Standards (FIPS) 200, Minimum Security Requirements for Federal Information and Information Systems*" by NIST of United States of America[2]

> *Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. Also, the potential for a threat-source to successfully exploit a particular information system vulnerability.*

What is a Threat?

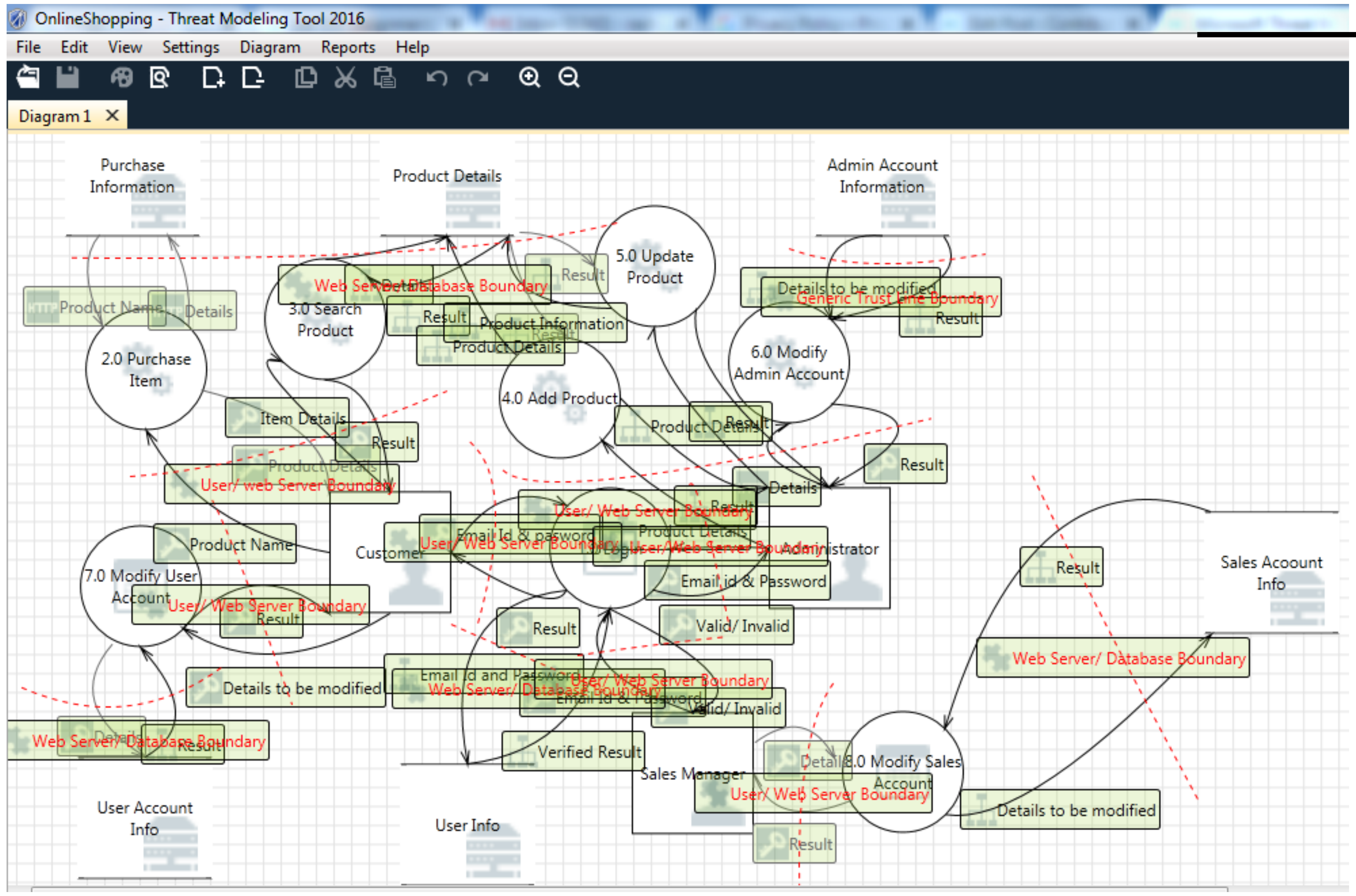

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

- **ISO27005 (International for Standardisation) - Information Security Risk Management:** A potential cause of an incident, that may result in harm of systems and organisation

- **NIST (National Institute of Standards and Technology):** Any circumstance or event with the potential to adversely impact organisational operations, organisational assets, or individuals through an information system via unauthorised access, destruction, disclosure, modification of information, and / or denial of service

- **ENISA (European Union Agency for Cybersecurity):** Any circumstance or event with the potential to adversely impact an asset through unauthorised access, destruction, disclosure, modification of data, and / or denial of service

Microsoft’s Threat Classification

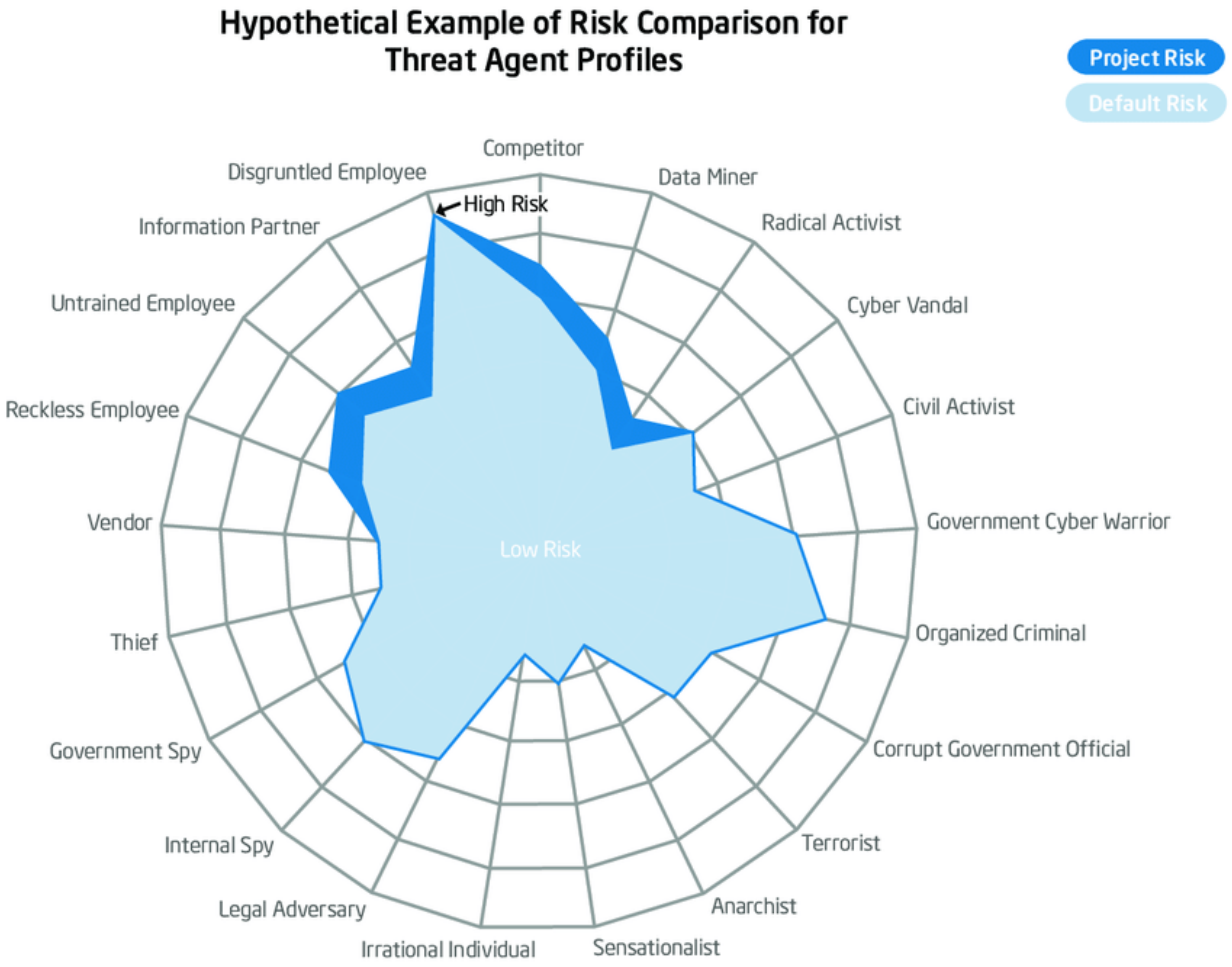


BASE OF THEIR THREAT MODELLING TOOL STRIDE

- Spoofing of user identity (e.g., an attacker takes on the identify of an administrator)
- Tampering (e.g., an attacker changes an account balance)
- Repudiation (e.g., a user denies performing an action without either parties having any way to prove otherwise)
- Information disclosure (privacy breach or data leak)
- Elevation of privilege (e.g., an attacker elevates their own security level to an administrator)
- Denial of Service (DoS)
  - A DoS attack is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the internet

Threat Agent

- The term threat agent is used to indicate an individual, thing or a group that can manifest a threat
- These include:
  - Non-target specific (e.g., computer virus, worms, trojans, and logic bombs)
  - Employees — disgruntled staff or contractors
  - Organised crime and criminals
  - Corporation (e.g., partners or competitors)
  - Human (unintentional) — accidents, carelessness
  - Human (intentional) — insider, outsider
  - Natural (e.g., flood, fire, lightning, meteor, earthquakes)



Threats

Requirement-level threats

- Expertise in requirements engineering and information system security is a rare combination:
  - Customers and users also don’t know what they want with respect to security
  - Requirement engineers don’t know what questions to ask to elicit security requirements
- This combined lack of security expertise lead to missing our or unidentified security requirements, resulting in security vulnerabilities

Hardware-level threats & countermeasures

Threat	**Countermeasure**
Eavesdropping devices (e.g., keyloggers)	Physical security
Power outage	UPS (Uninterruptible Power Supply)
Natural disasters	Geographically dispersed redundancy to avoid a single point of failure
Sabotage	Physical security

Code-Level Threats

UNINTENTIONAL

- Mainly due to the lack of secure coding knowledge
- Example: Some library functions in C are vulnerable to buffer overflow attacks
    - Here an attacker puts too much information into a buffer, so that it spills to another memory management, thereby overwriting programme code or data

→

- Software security education and training
- Automatic static and dynamic code analysis
- Peer code review

INTENTIONAL

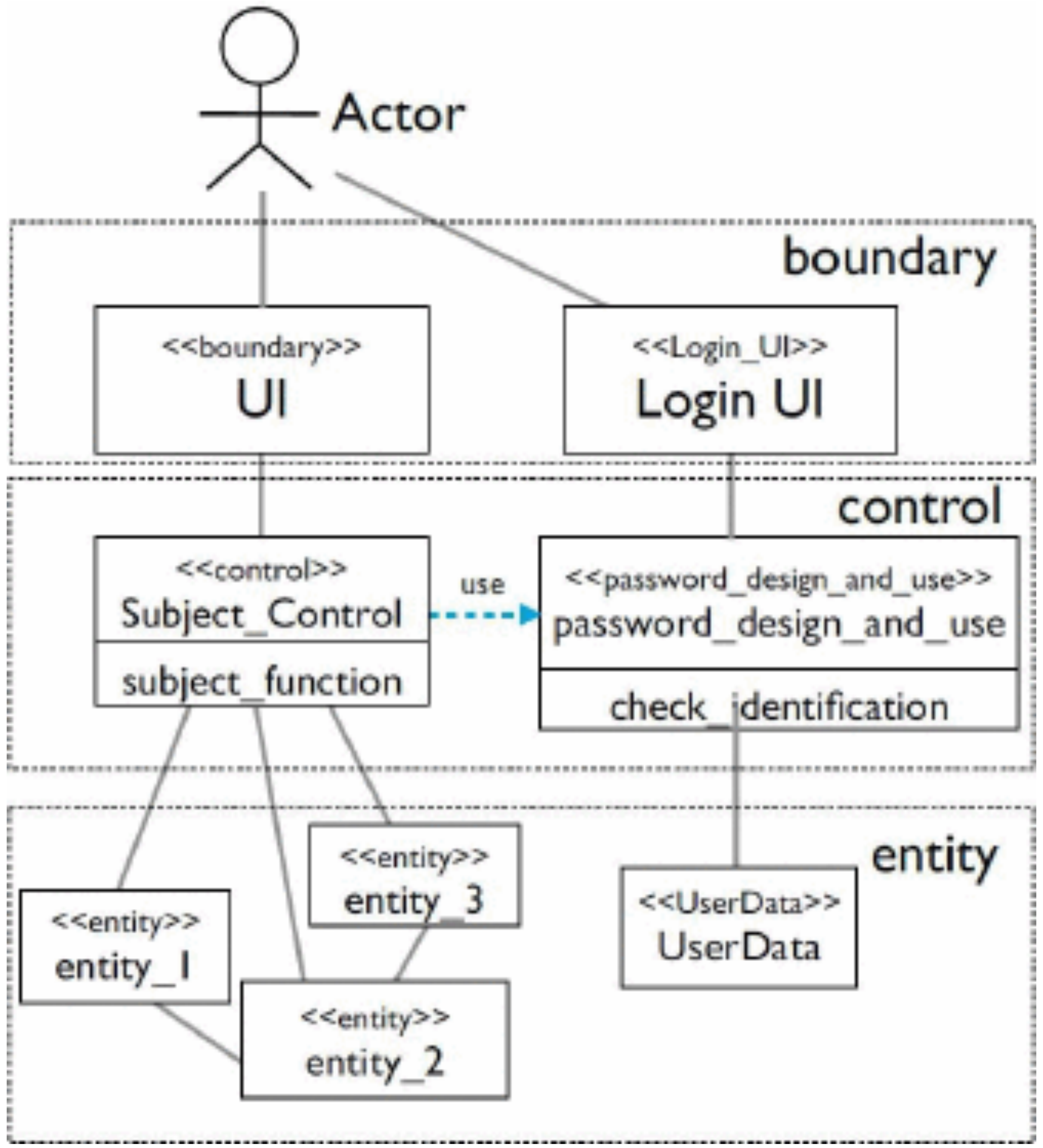
- Example: Malicious insider plants a logic bomb in the source code, which eventually makes the software misbehave or vulnerable

→

- Peer code review
- Job rotation
- Mandatory vacation ⚠️

Design-Level Threats

- Design-Level threats relate to weaknesses in principal OO design and object interaction, therefore secure design is more fundamental than secure coding
    - e.g., object attributes being public rather than private
- Best OO design practices are captures in design patterns for security
- Code implementation without a solid design is dangerous and costly ⚠



STRUCTURE OF A SECURITY PATTERN (PASSWORD DESIGN AND USE PATTERN)

Architectural-Level Design Threats

- Architectural design decisions entail overarching design decisions
- Widely accepted solutions to these recurring architectural design problems are referred to as architectural patterns
- Example:
    - A single access point is an architectural pattern
        - ✓ e.g., software has the single access point
        - ✓ Potentially single point of failure

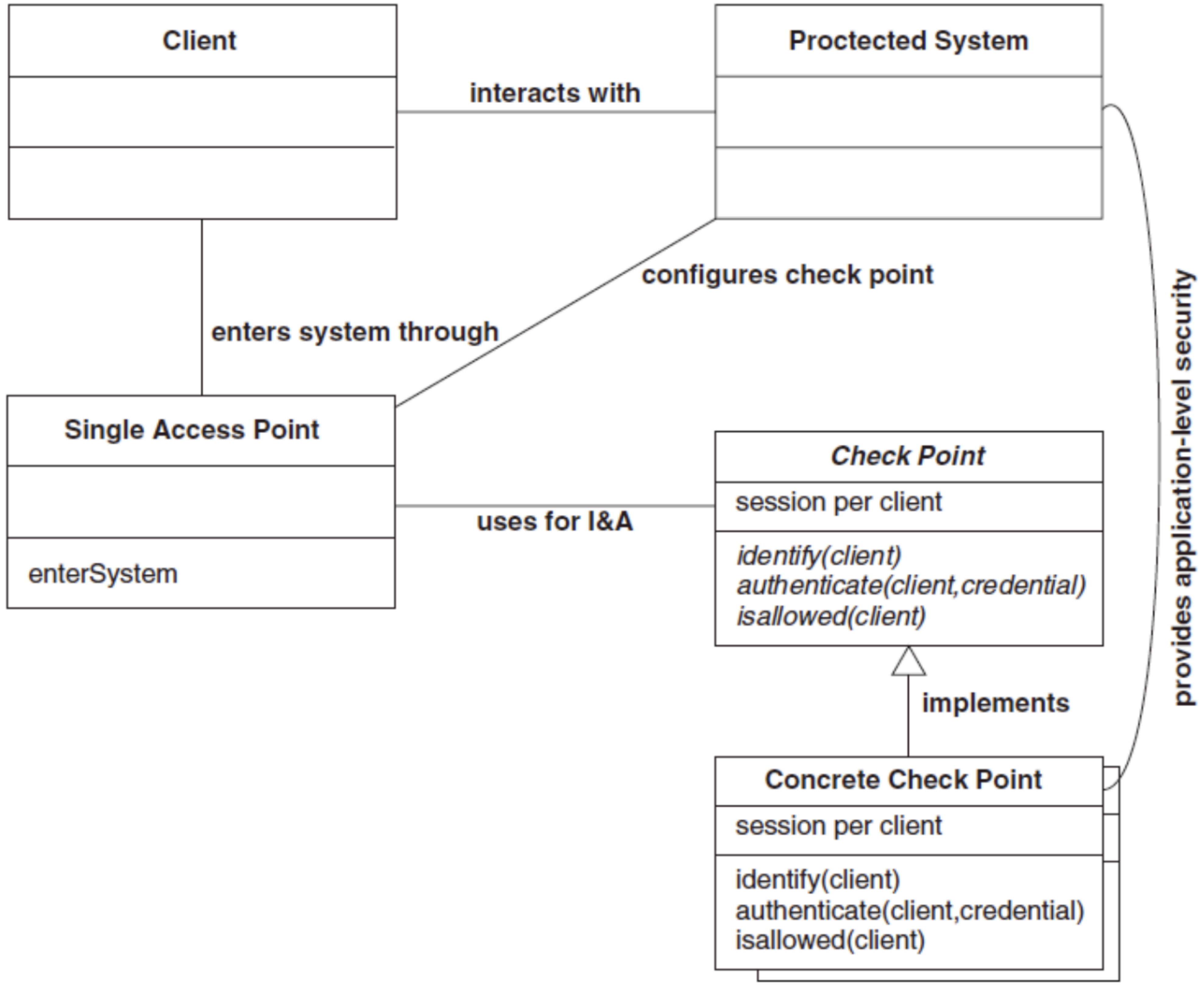
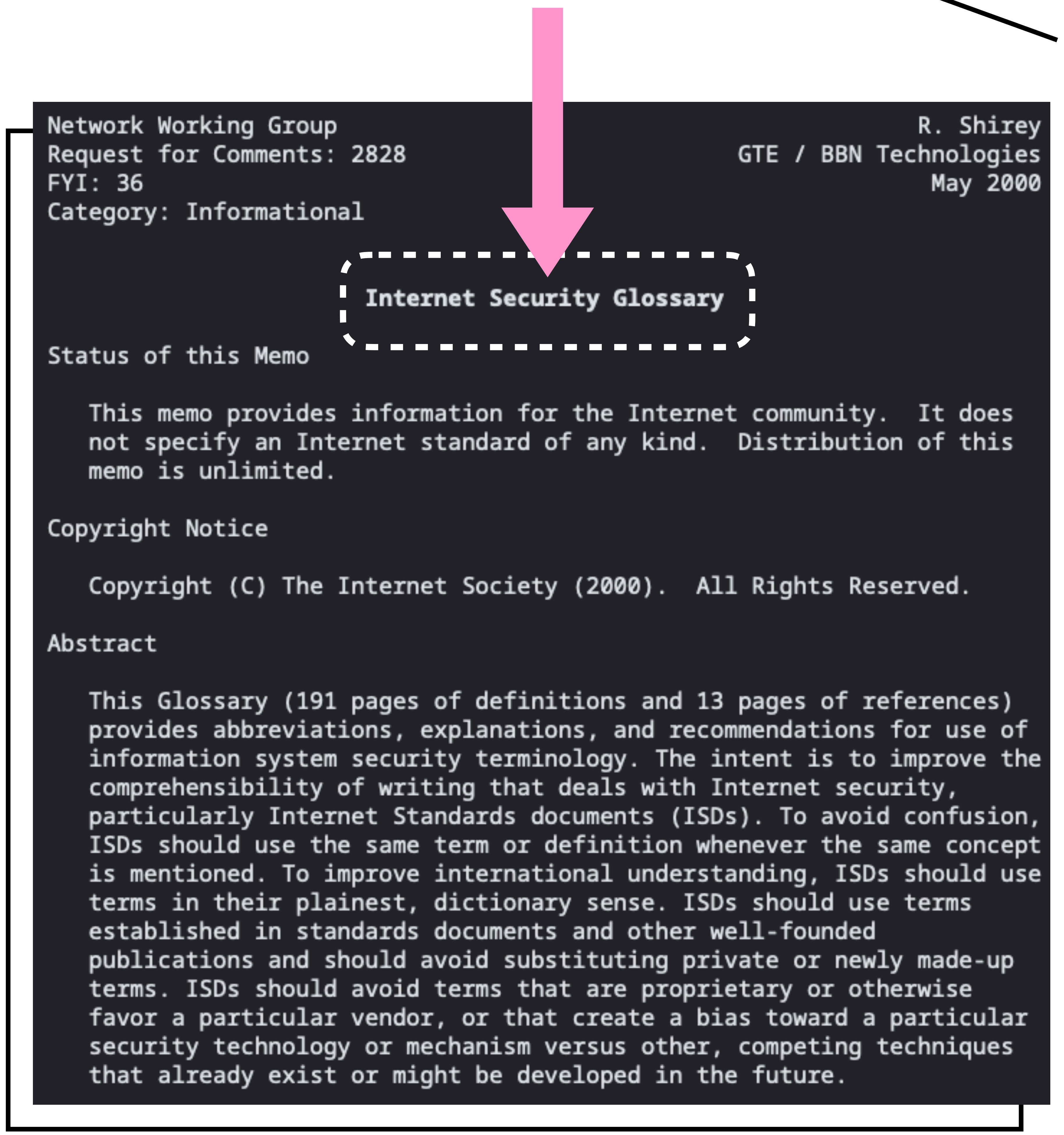


Fig. 2. The Single Access Point pattern and the Check point Pattern

Vulnerability (RFC2828)

DOCUMENT PUBLISHED BY INTERNET ENGINEERING TASK FORCE (IETF) IN MAY 2000 – PROVIDE A STANDARDISED GLOSSARY OF SECURITY-RELATED TERMS AND CONCEPTS RELEVANT TO THE INTERNET COMMUNITY.



```
Network Working Group                                          R. Shirey
Request for Comments: 2828                        GTE / BBN Technologies
FYI: 36                                                         May 2000
Category: Informational


                       Internet Security Glossary

Status of this Memo

   This memo provides information for the Internet community.  It does
   not specify an Internet standard of any kind.  Distribution of this
   memo is unlimited.

Copyright Notice

   Copyright (C) The Internet Society (2000).  All Rights Reserved.

Abstract

   This Glossary (191 pages of definitions and 13 pages of references)
   provides abbreviations, explanations, and recommendations for use of
   information system security terminology. The intent is to improve the
   comprehensibility of writing that deals with Internet security,
   particularly Internet Standards documents (ISDs). To avoid confusion,
   ISDs should use the same term or definition whenever the same concept
   is mentioned. To improve international understanding, ISDs should use
   terms in their plainest, dictionary sense. ISDs should use terms
   established in standards documents and other well-founded
   publications and should avoid substituting private or newly made-up
   terms. ISDs should avoid terms that are proprietary or otherwise
   favor a particular vendor, or that create a bias toward a particular
   security technology or mechanism versus other, competing techniques
   that already exist or might be developed in the future.
```

- A flow of weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy
- Most system have vulnerabilities of some sort, but this does not mean that the systems are too flawed to use.
- Not every attack results in an attack, and not every attack succeeds.
- Success depends on the degree of vulnerability, the strength of attacks, and the effectiveness of any countermeasures in use
  - e.g., if the attacks needed to exploit a vulnerability are very difficult to carry out, then the vulnerability may be tolerable
- Vulnerabilities in software arises mainly due to **defects**

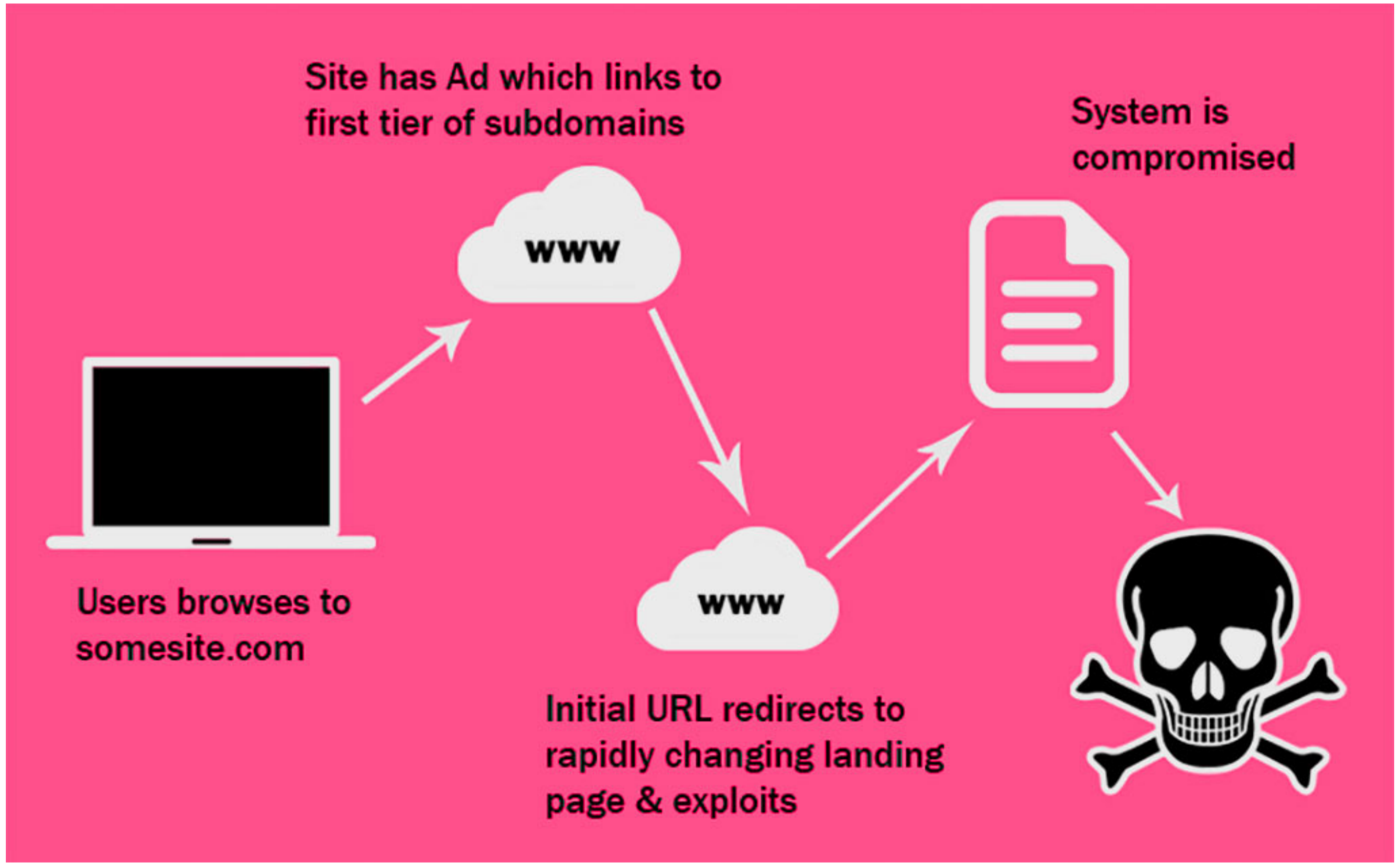
Exploit

- An exploit is a piece of software, data, or a sequence of commands that takes advantage of a vulnerability to cause unintended or unanticipated behaviour to occur on computer software or hardware

- Such behaviour frequently includes things like gaining control of a computer system, allowing privilege escalation, or a denial-of-service attack

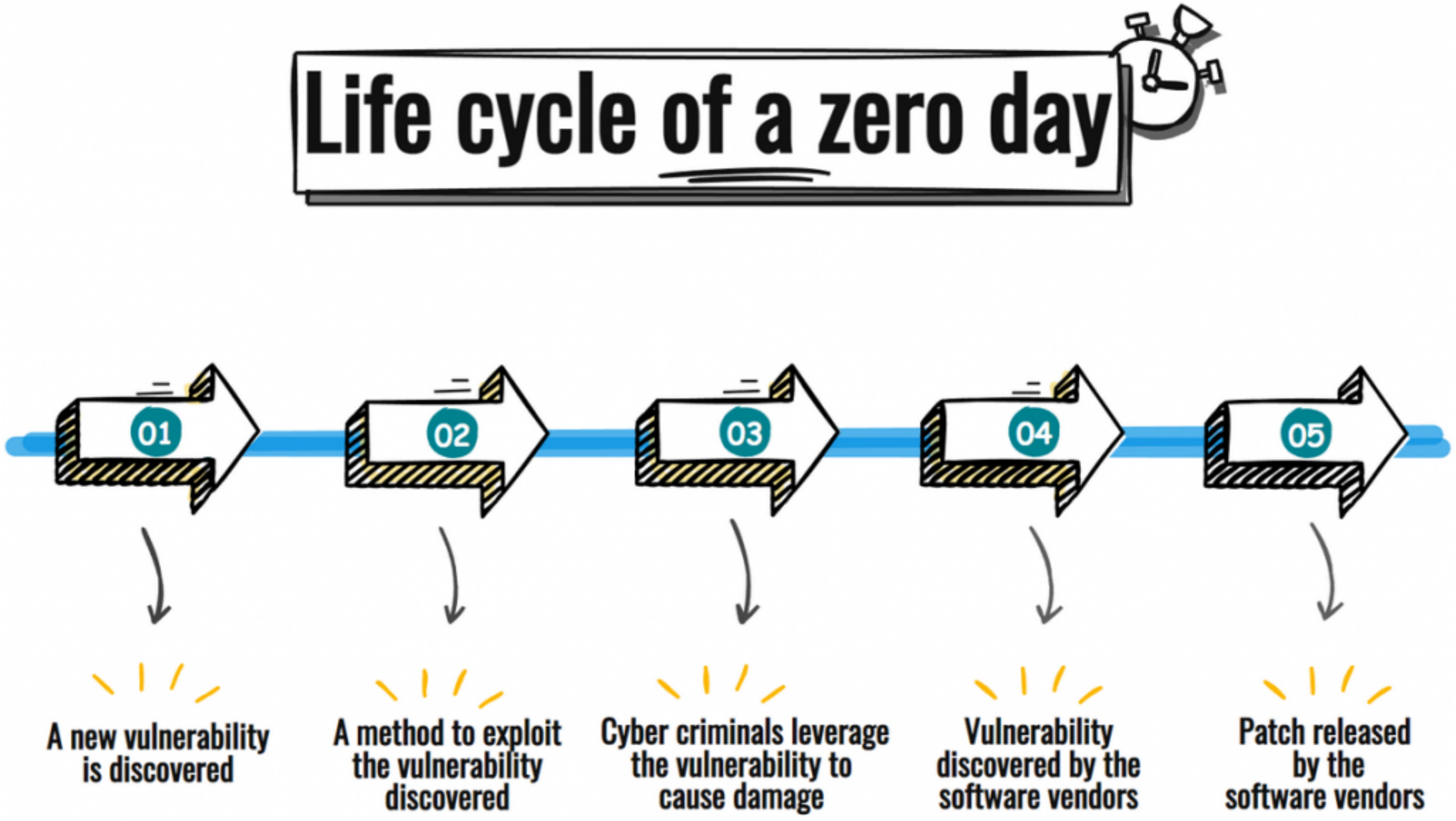
- A remote exploit works over a network and exploits the security vulnerability without any prior access to the vulnerable system

- A local exploit requires prior access to the vulnerable system and usually increases the privileges of the person running the exploit past those granted by the system administrator


Site has Ad which links to first tier of subdomains
System is compromised
www
Users browses to somesite.com
www
Initial URL redirects to rapidly changing landing page & exploits

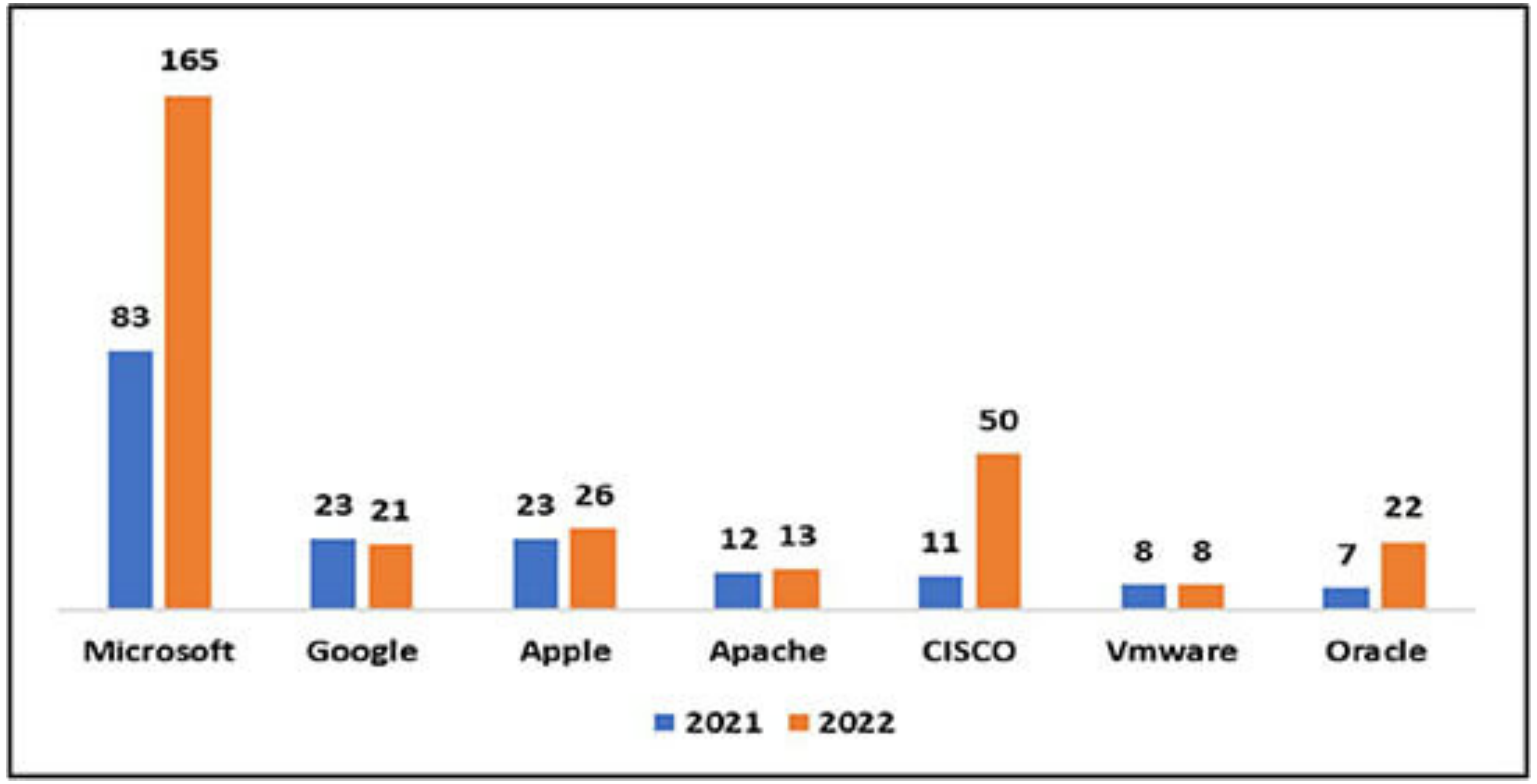
Zero-Day Vulnerability

- A zero-day vulnerability is a computer-software vulnerability that is unknown to those who should be interested in mitigating the vulnerability (including the vendor of the target software)
- Until the vulnerability is mitigated, hackers can exploit it to adversely affect computer programs, data, additional computers or a network
- An exploit directed at a zero-day is called a zero-day exploit, or a zero-day attack


Life cycle of a zero day
01
A new vulnerability is discovered
02
A method to exploit the vulnerability discovered
03
Cyber criminals leverage the vulnerability to cause damage
04
Vulnerability discovered by the software vendors
05
Patch released by the software vendors

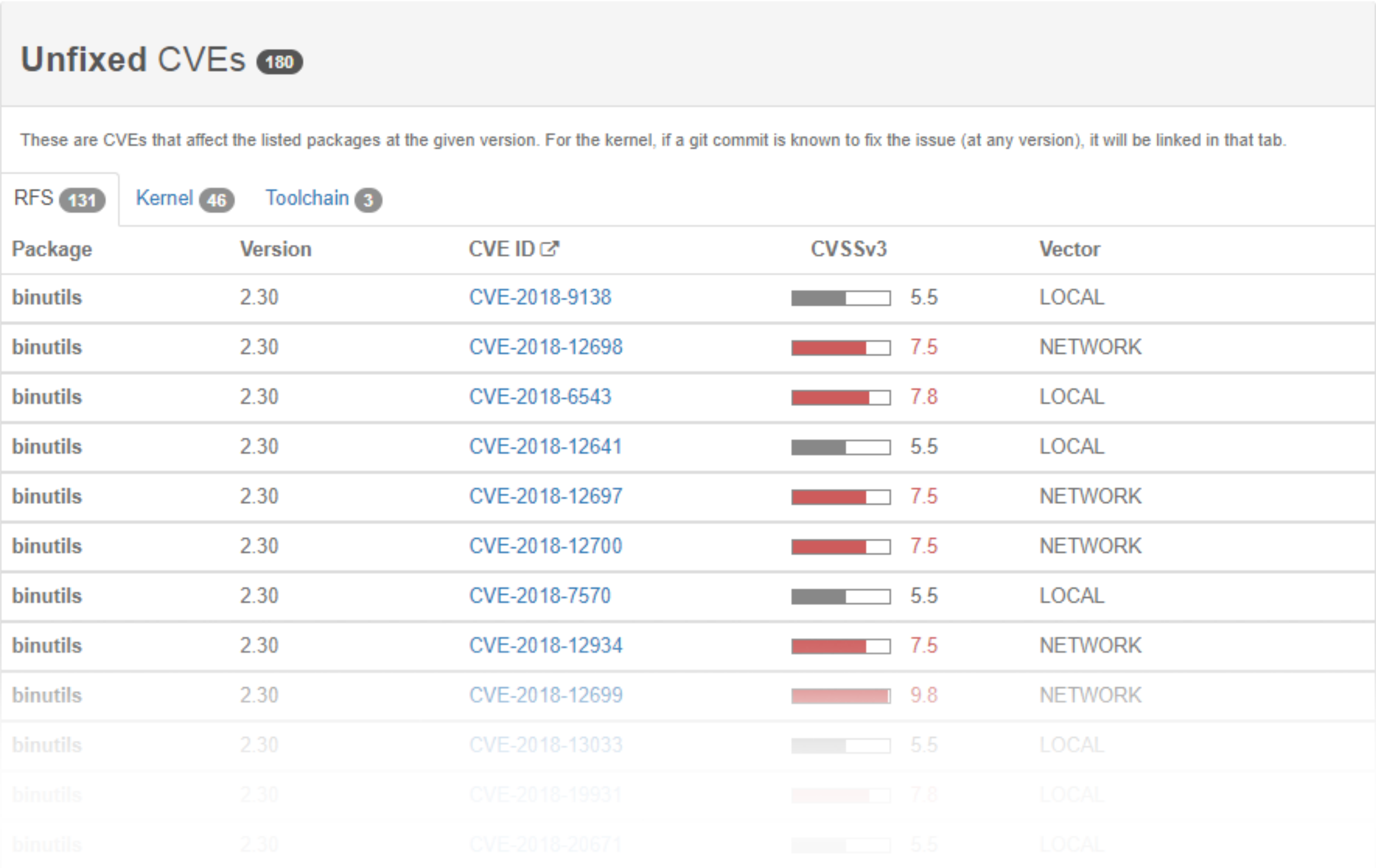
Zero-Day Vulnerability

- Zero-day vulnerabilities can effect:
  - Libraries used across multiple products (e.g., OpenSSL)
  - Generically used software (e.g., MSOffice)
  - Proprietary software (e.g., XSS vulnerability in smartphone app)
- However, vulnerabilities will become public eventually, and have to be communicated to affected users, before being patched (e.g., MS Update and Security)



CVE Systems

- CVE = Common Vulnerabilities and Exposure Systems
- Security vulnerabilities need to be managed systematically to help identify weaknesses in the affected source code of a software system
- CVE provides a reference-method for publicly known information-security vulnerabilities
- The National Cybersecurity FFRDC (NCF), operated by the MITRE Corporation (a non-profit organisation that manages federally funded research and development centres) maintains the system, with funding from the National Cyber Security Division of the United States Department of Homeland Security



Unfixed CVEs 180

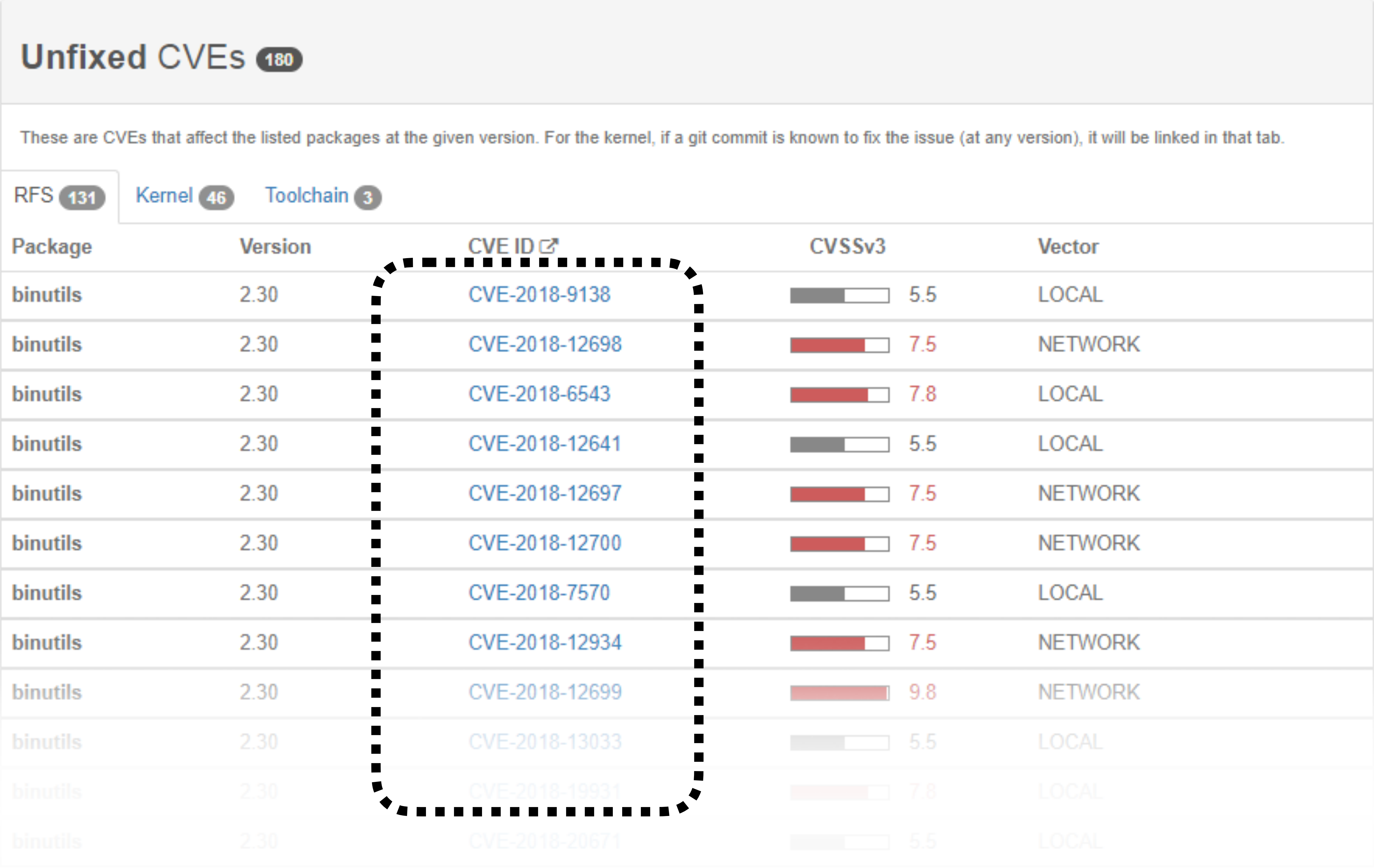
These are CVEs that affect the listed packages at the given version. For the kernel, if a git commit is known to fix the issue (at any version), it will be linked in that tab.

Package	Version	CVE ID	CVSSv3	Vector
binutils	2.30	CVE-2018-9138	5.5	LOCAL
binutils	2.30	CVE-2018-12698	7.5	NETWORK
binutils	2.30	CVE-2018-6543	7.8	LOCAL
binutils	2.30	CVE-2018-12641	5.5	LOCAL
binutils	2.30	CVE-2018-12697	7.5	NETWORK
binutils	2.30	CVE-2018-12700	7.5	NETWORK
binutils	2.30	CVE-2018-7570	5.5	LOCAL
binutils	2.30	CVE-2018-12934	7.5	NETWORK
binutils	2.30	CVE-2018-12699	9.8	NETWORK
binutils	2.30	CVE-2018-13033	5.5	LOCAL
binutils	2.30	CVE-2018-19931	7.8	LOCAL
binutils	2.30	CVE-2018-20671	5.5	LOCAL

CVE Systems

- CVE (https://cve.mitre.org/) is a central repository of all the reported security vulnerabilities associated with a specific software system
  - Each CVE entry has a unique identifier which is commonly used by many commercial vulnerability management systems to refer to a specific vulnerability.
- CWE (common weakness enumeration) -

  https://cwe.mitre.org/ categorises the vulnerabilities identified in CVE
  - CWE has much fewer DB entries than the CVE



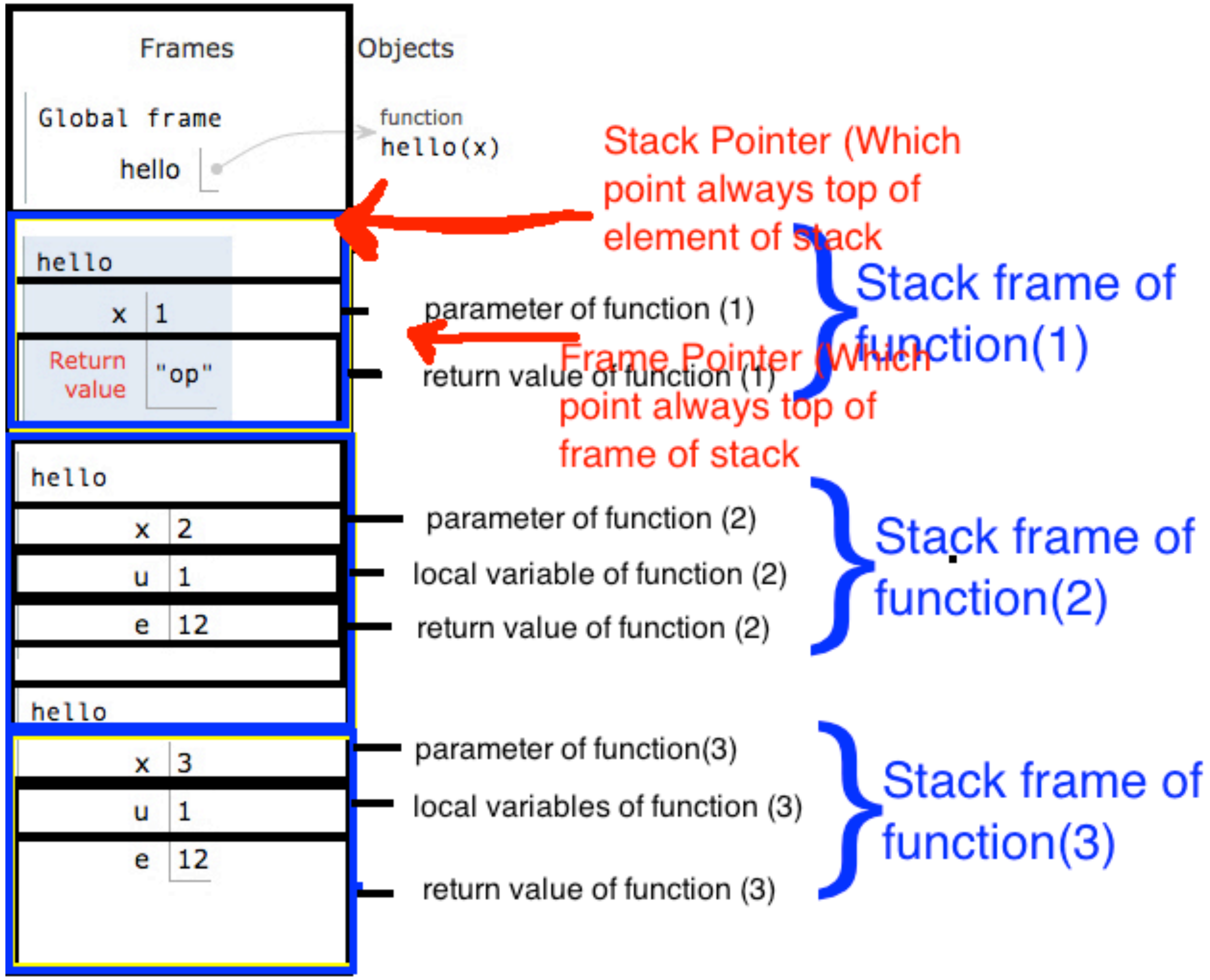
Unfixed CVEs 180

These are CVEs that affect the listed packages at the given version. For the kernel, if a git commit is known to fix the issue (at any version), it will be linked in that tab.

RFS 131 | Kernel 46 | Toolchain 3

Package	Version	CVE ID	CVSSv3	Vector
binutils	2.30	CVE-2018-9138	5.5	LOCAL
binutils	2.30	CVE-2018-12698	7.5	NETWORK
binutils	2.30	CVE-2018-6543	7.8	LOCAL
binutils	2.30	CVE-2018-12641	5.5	LOCAL
binutils	2.30	CVE-2018-12697	7.5	NETWORK
binutils	2.30	CVE-2018-12700	7.5	NETWORK
binutils	2.30	CVE-2018-7570	5.5	LOCAL
binutils	2.30	CVE-2018-12934	7.5	NETWORK
binutils	2.30	CVE-2018-12699	9.8	NETWORK
binutils	2.30	CVE-2018-13033	5.5	LOCAL
binutils	2.30	CVE-2018-19931	7.8	LOCAL
binutils	2.30	CVE-2018-20671	5.5	LOCAL

The Call Stack



Stack Pointer (Which point always top of element of stack

parameter of function (1)

return value of function (1)

Frame Pointer (Which point always top of frame of stack

Stack frame of function(1)

parameter of function (2)

local variable of function (2)

return value of function (2)

Stack frame of function(2)

parameter of function(3)

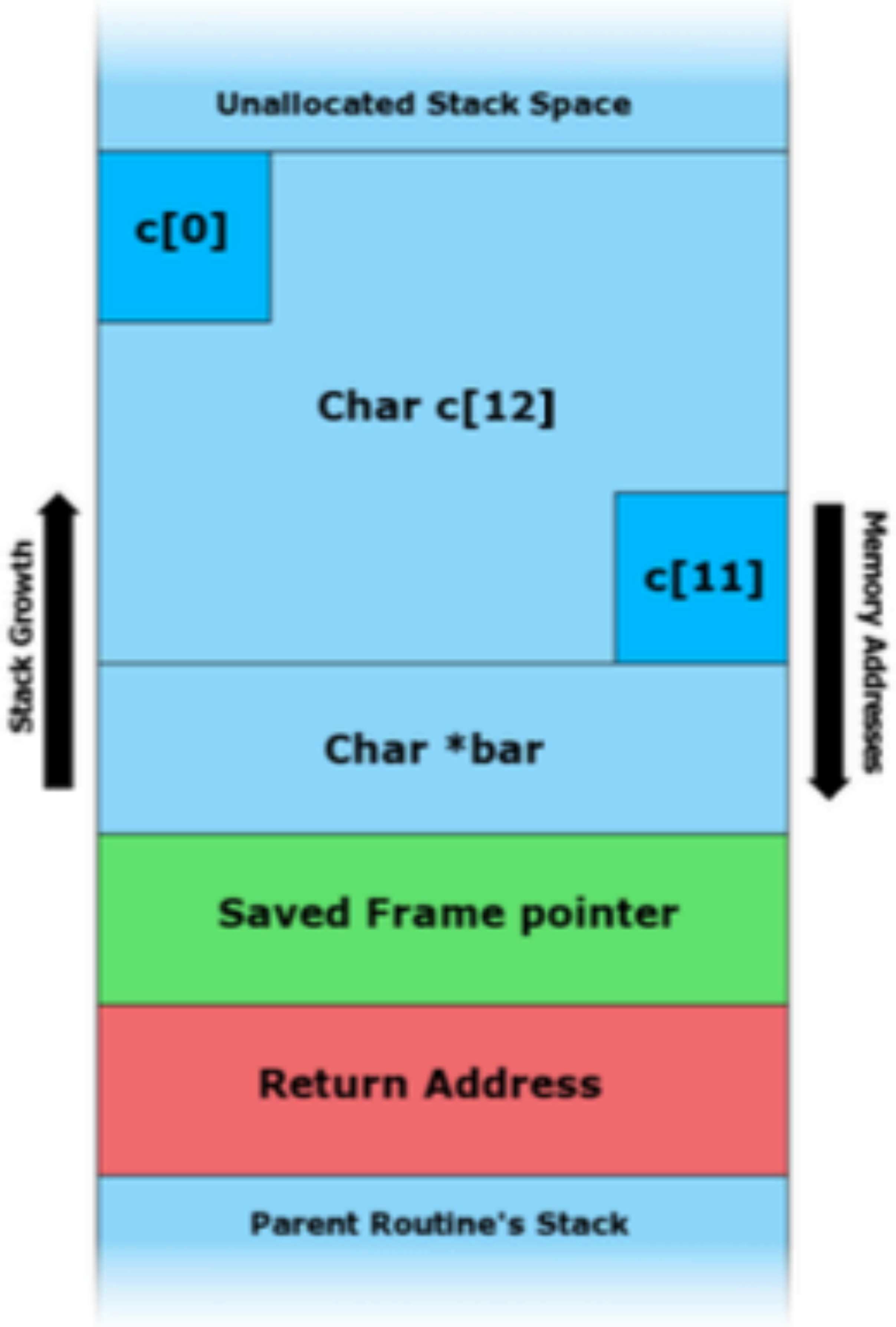
local variables of function (3)

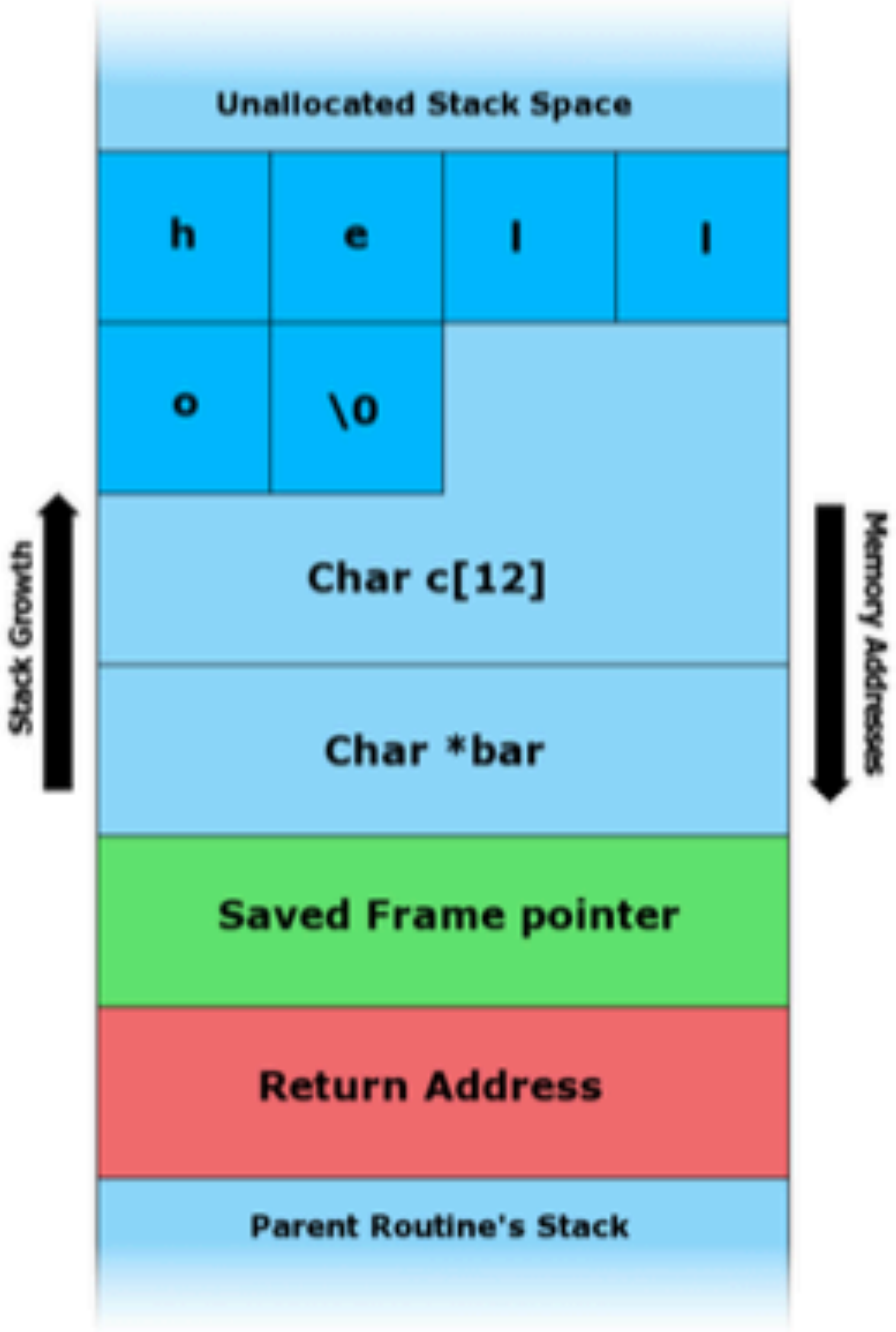
return value of function (3)

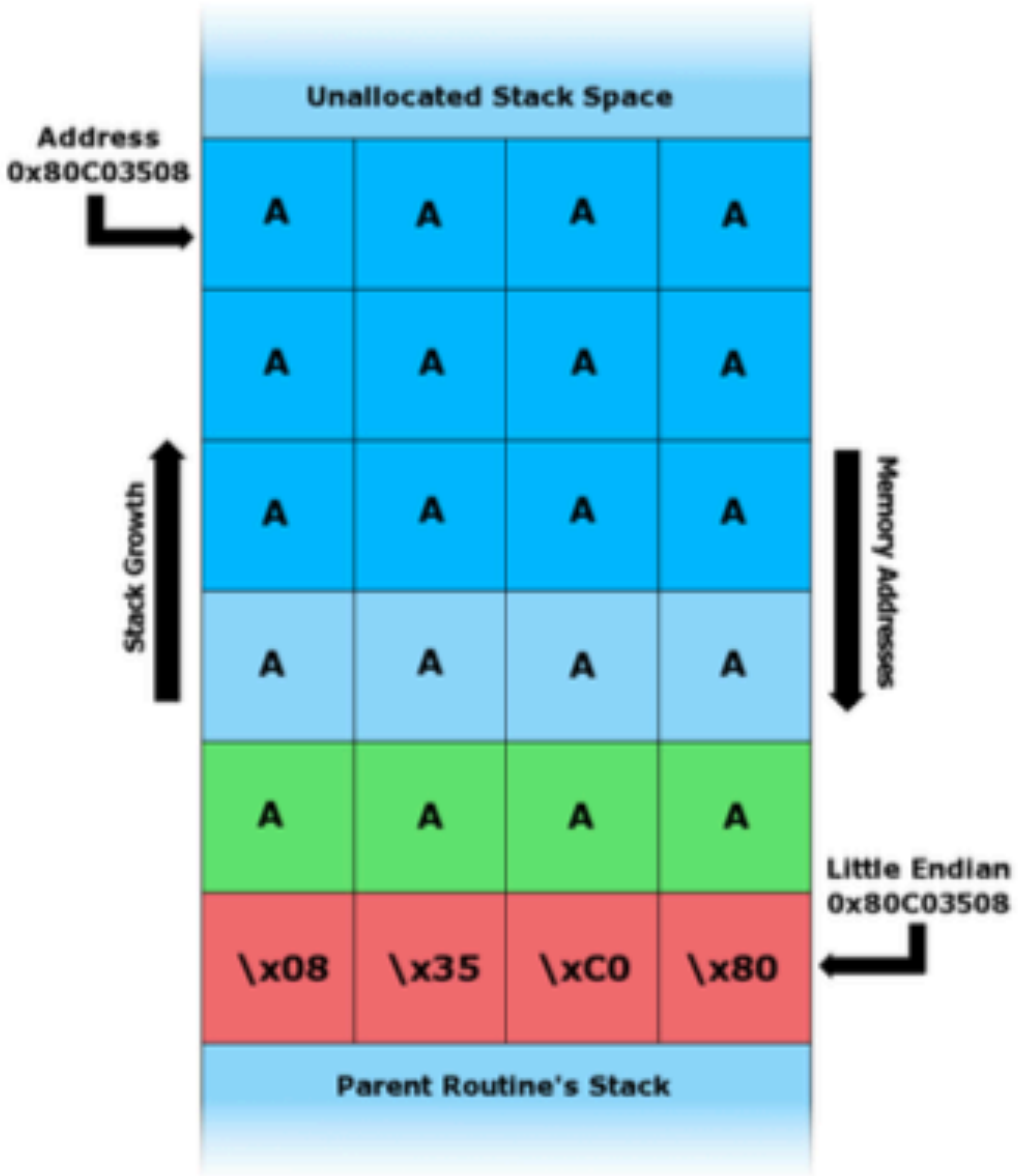
Stack frame of function(3)

- Each stack frame contains a stack pointer to the top of the frame immediately below
    - The stack pointer is a mutable register
- The stack frame is the collection of all data on the stack associated with one subprogram call, The stack frame generally includes the return address, argument variables passed on the stack, and local variables
- A frame pointer of a given invocation of a function is a copy of then stack pointer as it was before the function was invoked
- If a stack is corrupted, i.e., overwritten, arguments, variables and or return address do change

Stack Overflow







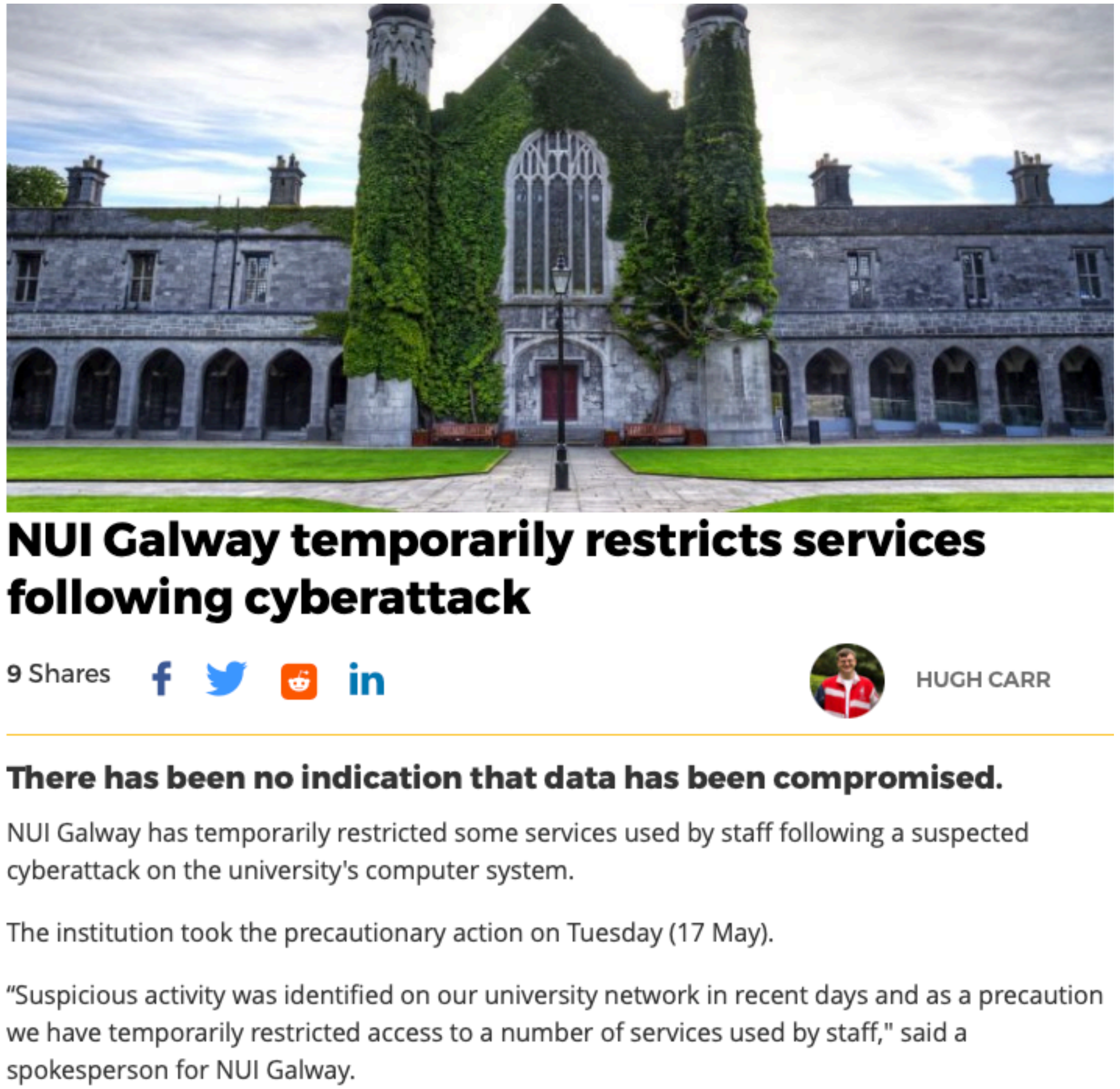
Buffer-Overflow Countermeasures

- Use a programming language that supports automatic bound checking of buffers
  - Java or Python, but NOT C
- Use a language specific library module that implements info validation in the form of safe buffer handling
- Compilers can produce a warning when an unsafe function call is made, or can add code for buffer overflow detection
- An Operating System can enforce more stringent memory access control so that buffer overflows cannot infringe into the protected areas of the main memory

Security Tactics

- A security tactic is a global design concept that addresses a security problem at the architectural design level
- There are four main categories of security tactics:
  - Tactics to help detect attacks, e.g., intrusion detection system
  - Tactics that are used to resist attacks, e.g., single access point, authenticate users
  - Tactic to react to attacks
  - Tactic to recover from attacks

WAS IT CORRECT ?



NUI Galway temporarily restricts services following cyberattack

9 Shares

HUGH CARR

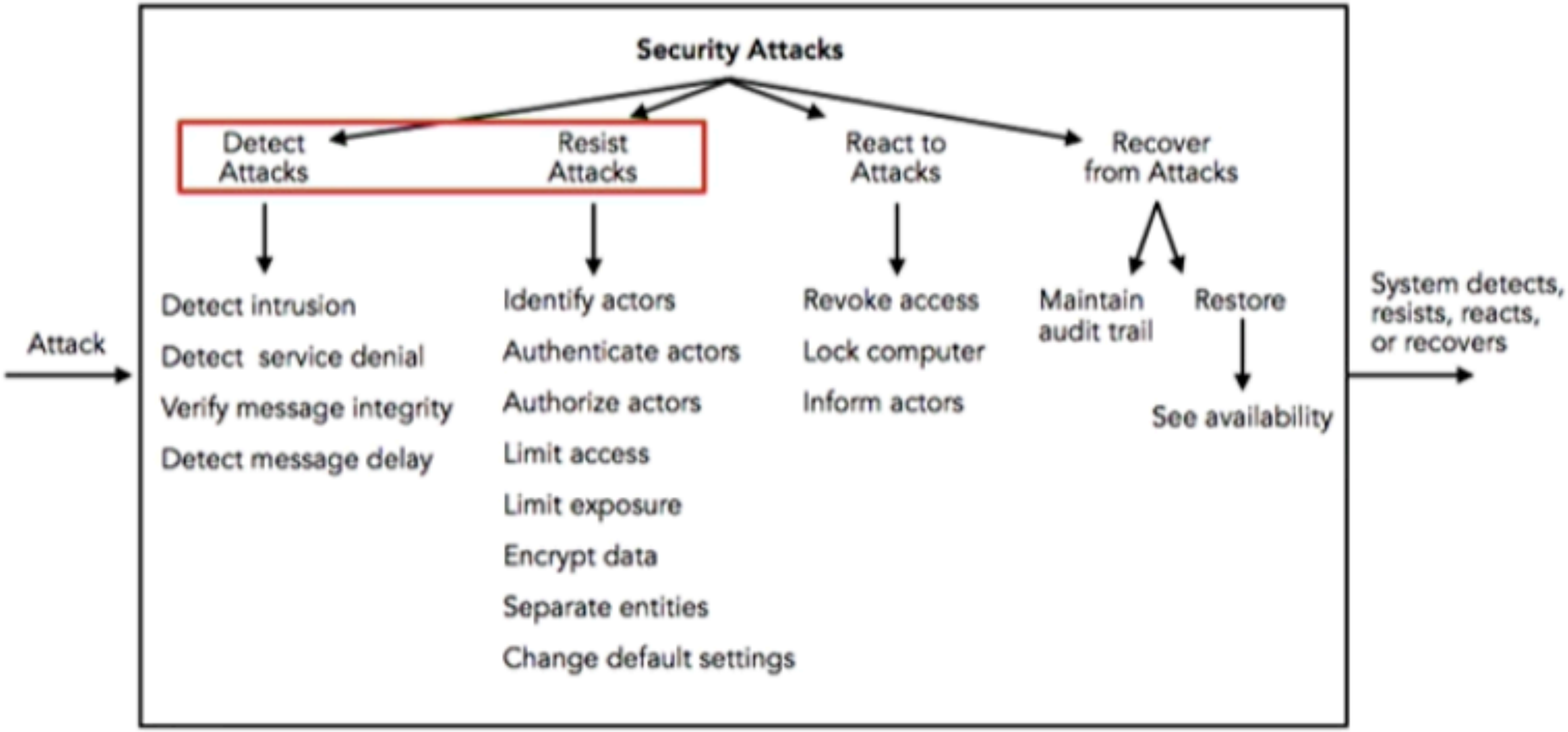
There has been no indication that data has been compromised.

NUI Galway has temporarily restricted some services used by staff following a suspected cyberattack on the university's computer system.

The institution took the precautionary action on Tuesday (17 May).

"Suspicious activity was identified on our university network in recent days and as a precaution we have temporarily restricted access to a number of services used by staff," said a spokesperson for NUI Galway.

Security Tactics



Attack

Security Attacks

Detect Attacks

- Detect intrusion
- Detect service denial
- Verify message integrity
- Detect message delay

Resist Attacks

- Identify actors
- Authenticate actors
- Authorize actors
- Limit access
- Limit exposure
- Encrypt data
- Separate entities
- Change default settings

React to Attacks

- Revoke access
- Lock computer
- Inform actors

Recover from Attacks

- Maintain audit trail
- Restore
  - See availability

System detects, resists, reacts, or recovers