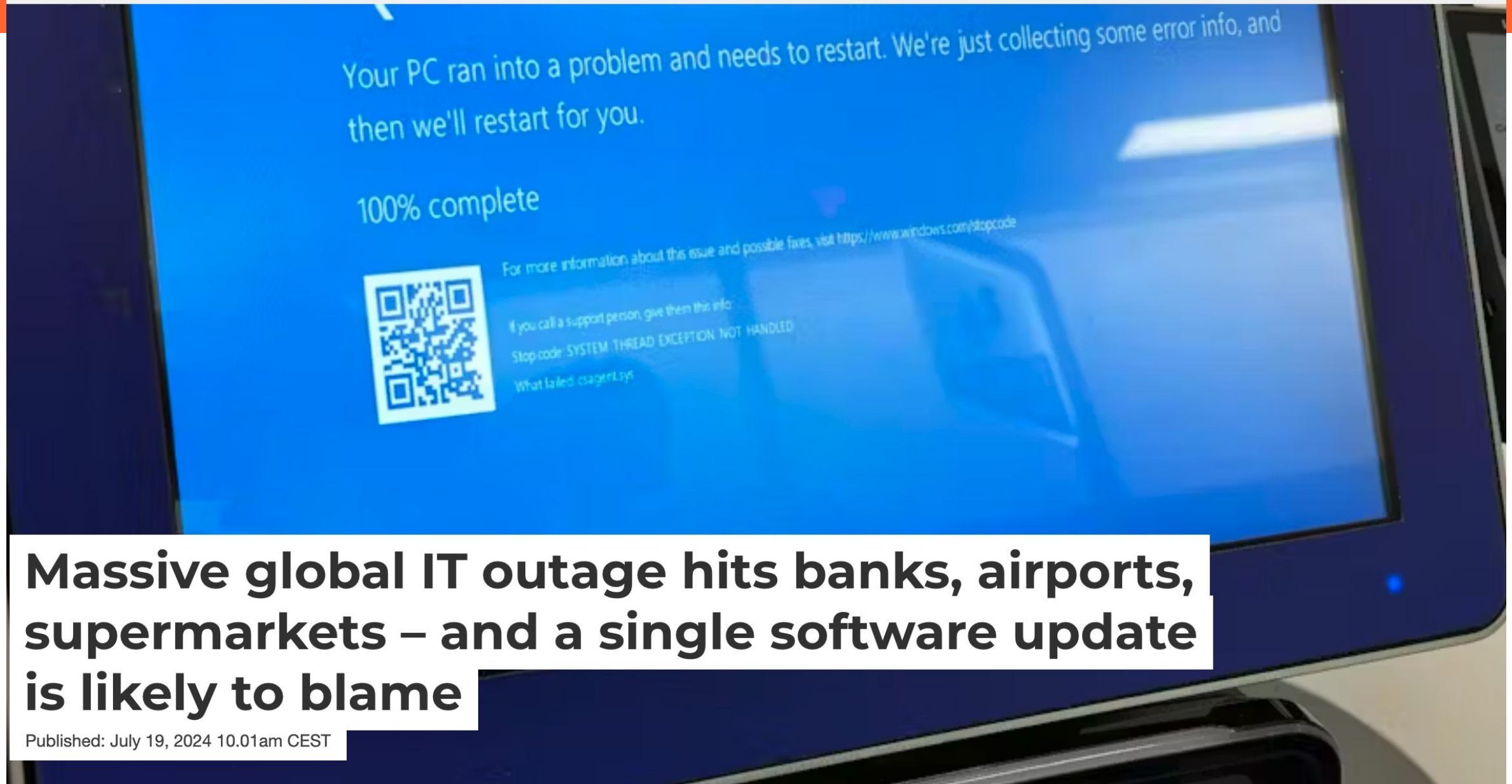
DataFlex

SECURITY MATTERS

BRAM NIJENKAMP

THE CONVERSATION
Academic rigour, journalistic flair

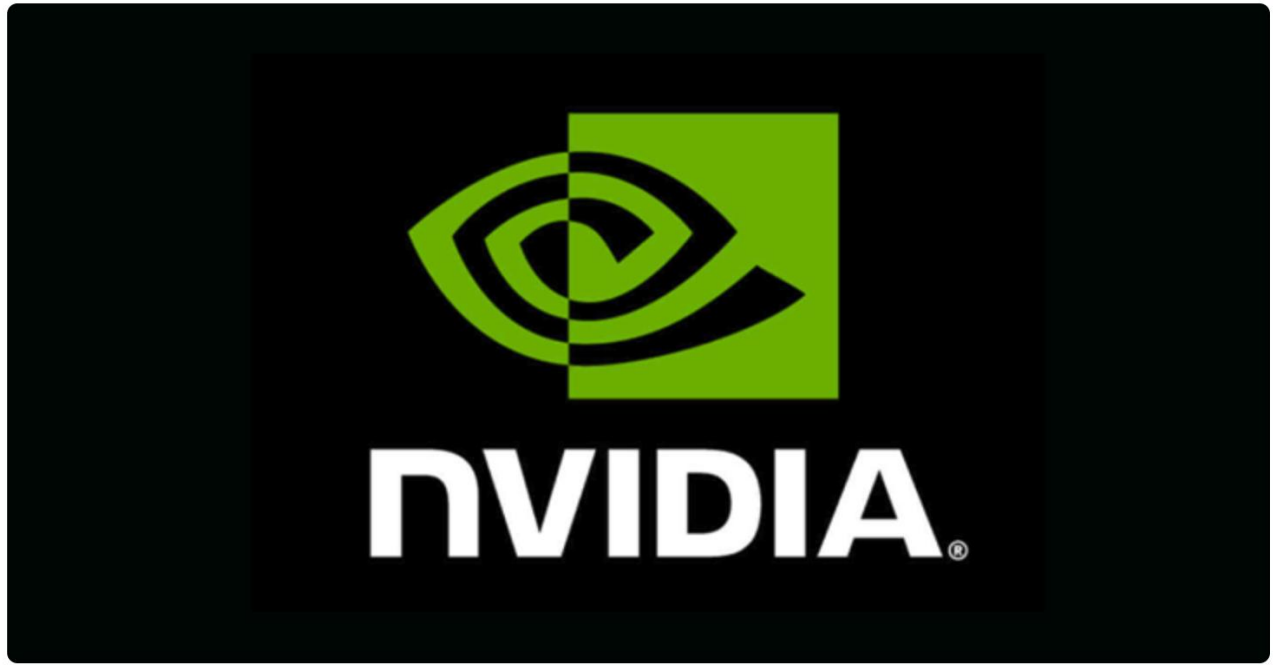


Massive global IT outage hits banks, airports, supermarkets – and a single software update is likely to blame

Published: July 19, 2024 10.01am CEST

Critical NVIDIA Container Toolkit Vulnerability Could Grant Full Host Access to Attackers

Sep 27, 2024 Ravie Lakshmanan Container Security / Cloud Computing



A critical security flaw has been disclosed in the NVIDIA Container Toolkit that, if successfully exploited, could allow threat actors to break out of the confines of a container and gain full access to the underlying host.

The vulnerability, tracked as **CVE-2024-0132**, carries a CVSS score of 9.0 out of a maximum of 10.0. It has been addressed in NVIDIA Container Toolkit version v1.16.2 and NVIDIA GPU Operator version 24.6.2.

"NVIDIA Container Toolkit 1.16.1 or earlier contains a Time-of-Check Time-of-Use (TOCTOU)

— Trending News

The Facts About Continuous Penetration Testing and Why It's Important

Healthcare's Diagnosis is Critical: The Cure is Cybersecurity Hygiene

Critical Ivanti Cloud Appliance Vulnerability Exploited in Active Cyberattacks

Google's Shift to Rust Programming Cuts Android Memory Vulnerabilities by 52%

Hacktivist Group Twelve Targets Russian Entities with Destructive Cyber Attacks

Phony Call Centers Tricking Users Into Installing Ransomware and Data-Stealers

Hackers Could Have Remotely Controlled Kia Cars Using Only License Plates

Passwordless AND Keyless: The

Cyber Security News | Vulnerability

VLC Player Vulnerability Let Attackers Execute Malicious Code, Update Now

By **Guru Baran** - September 27, 2024

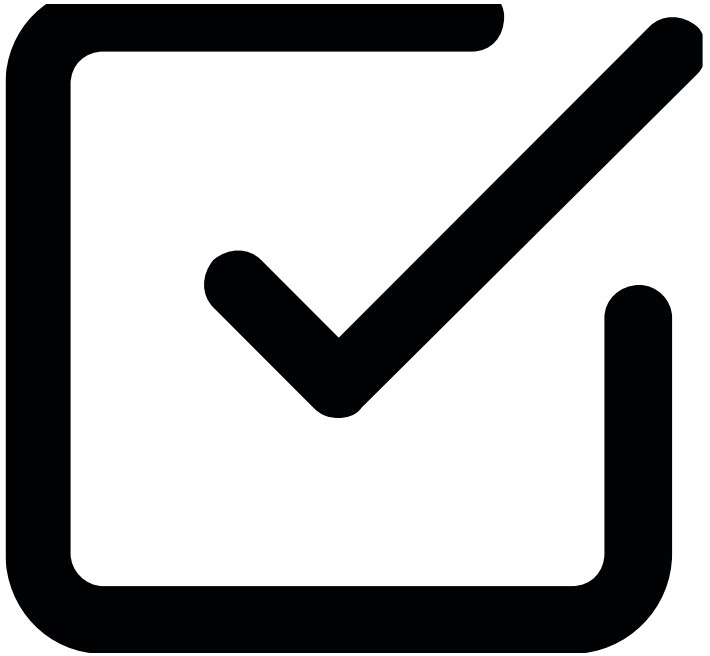
Latest News

- September 21, 2024

SECURITY COMES IN MANY DIFFERENT SHAPES AND SIZES…

BUT SO DO VULNERABILITIES…

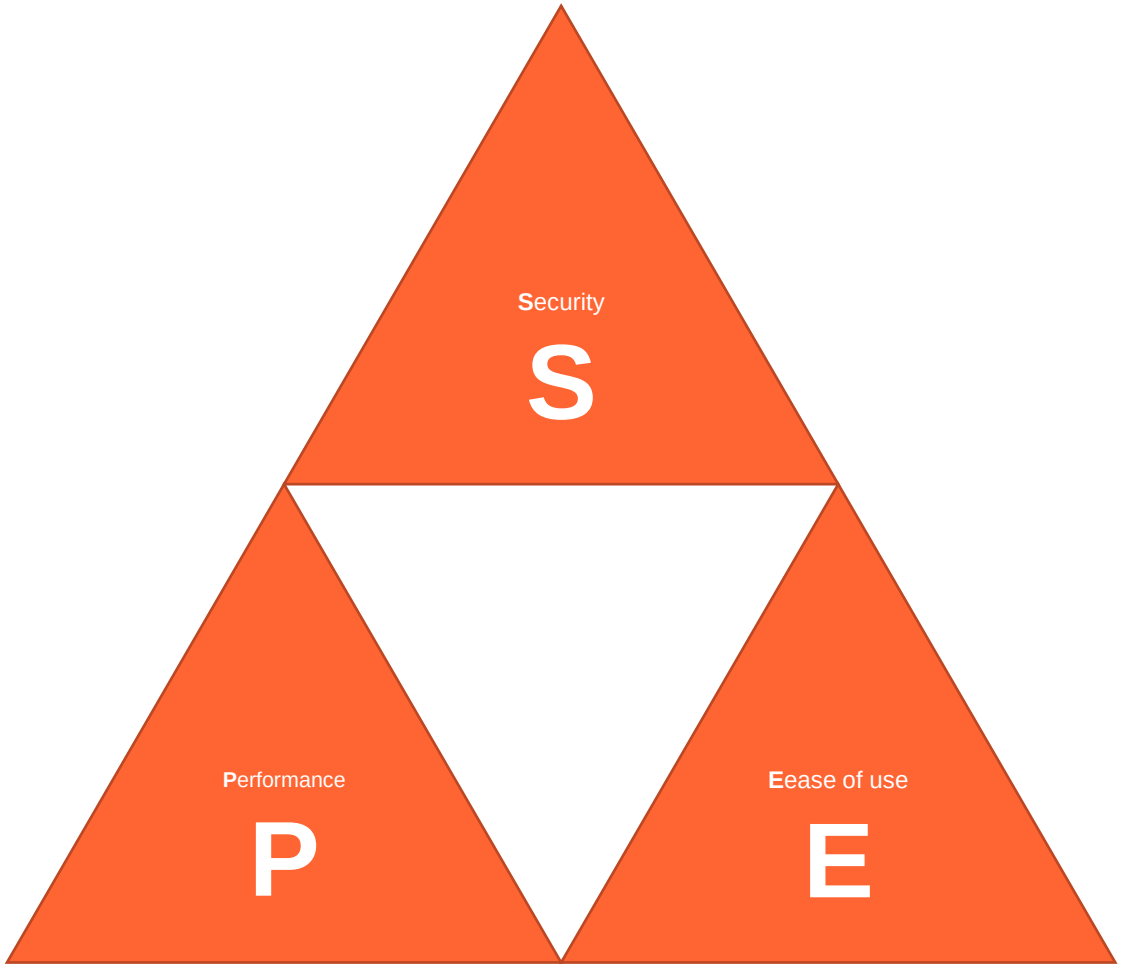
FRIENDLY REMINDER
CAN I CHECK IT OFF?



FRIENDLY REMINDER
NOT A CHECKBOX!

FRIENDLY REMINDER
IT IS…

- A continuous cycle
- Shared responsibility
  - But **you** are the driver!
- Being able to quickly respond
- As a business you balance risk
  - Performance
  - Security
  - Manageability/Ease of use

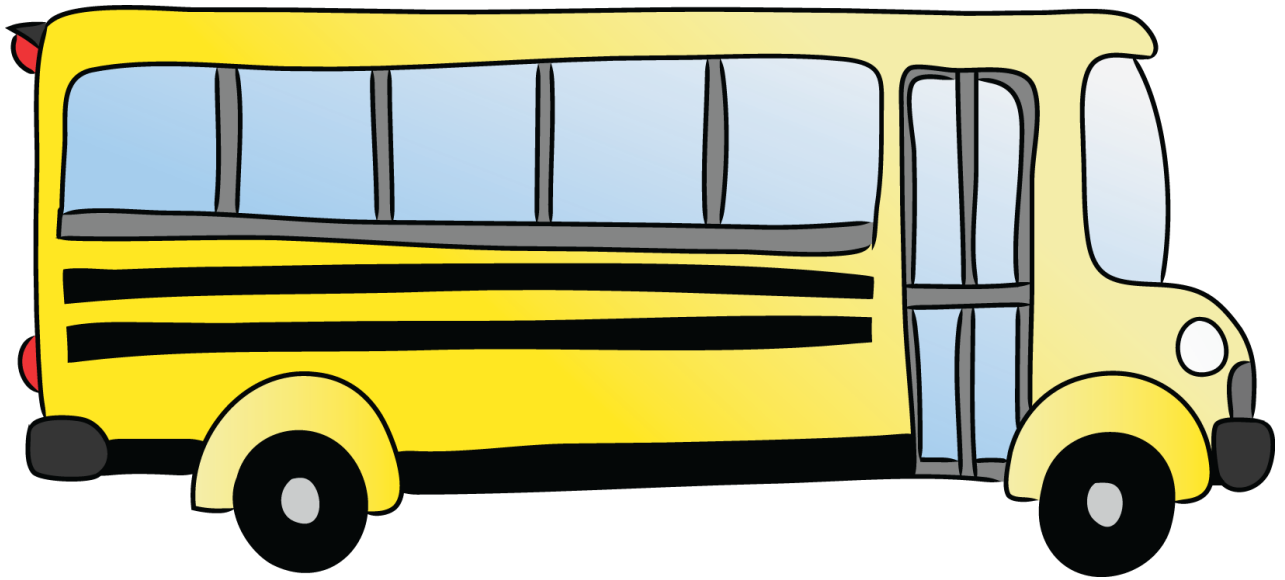


BUT THAT PART OF BEING THE DRIVER IS IMPORTANT!

DATAFLEX IS A FRAMEWORK!

FRIENDLY REMINDER
DATAFLEX IS A FRAMEWORK!

- We provide you with the engine of the bus you are going to drive...



FRIENDLY REMINDER
DATAFLEX IS A FRAMEWORK!

- But if one of your passengers doesn’t have their seat belts on... And you drive on a rough road…

IT IS IMPORTANT TO MAINTAIN YOUR BUS

MEANING THAT YOU NEED TO KEEP UP TO DATE WITH THE LATEST LTS VERSIONS OF FOR EXAMPLE DATAFLEX!

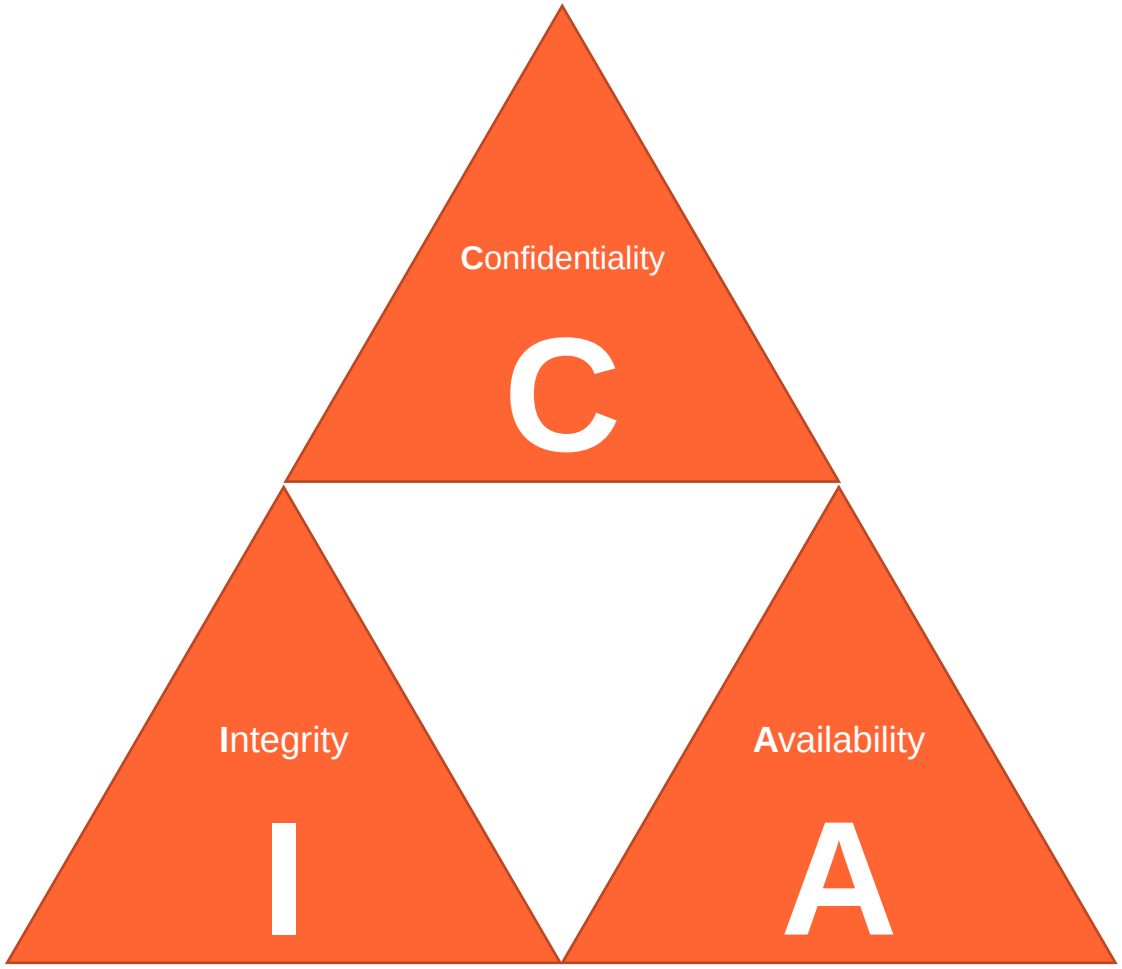
FRIENDLY REMINDER
DATAFLEX IS A FRAMEWORK!

- As such when build your applications make sure you **use, and use** the right tools

FRIENDLY REMINDER
OUR RESPONSIBILITY

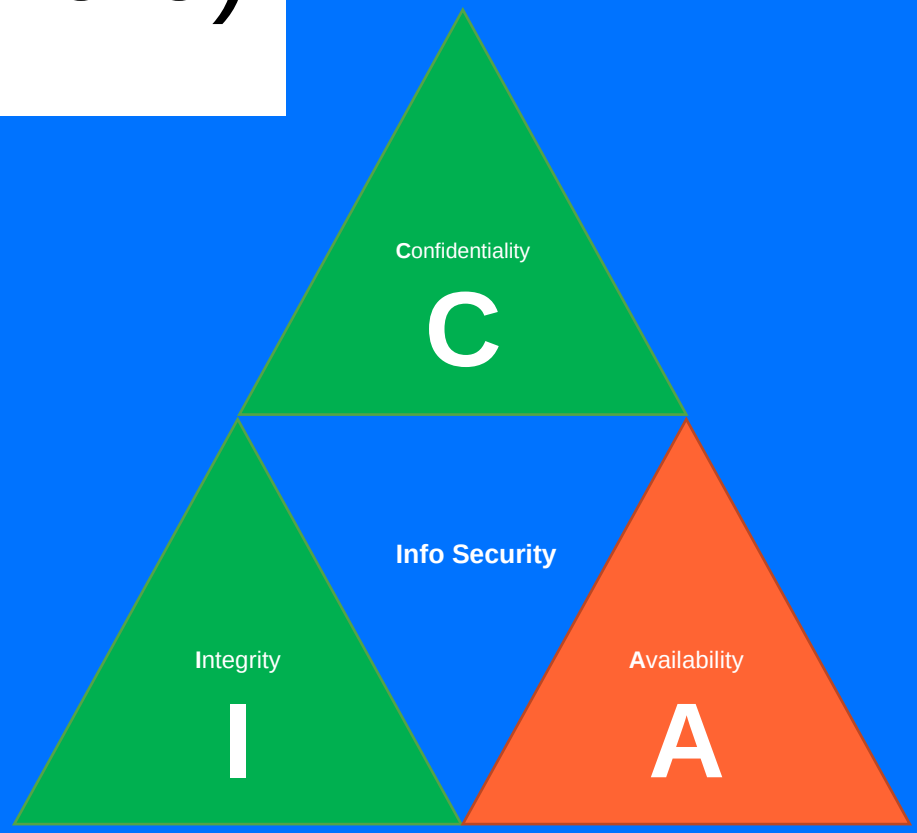
- We must implement security measures and best practices to ensure that our software is resilient against potential threats, safeguarding the confidentiality, integrity, and availability of sensitive data and functionalities

- And it is becoming more and more important, especially on the Web!

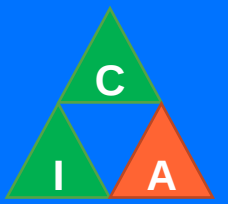


LIBRARIES

REDACTION LAYER (<20.0)

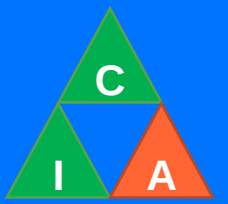


LIBRARIES
REDACTION LIBRARY



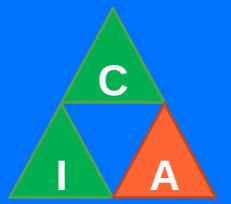
- Pre 20.0 data-aware controls could leak data or be vulnerable to data injection
- 20.0+ fixes this issue by default and older version get the redaction library
- Prevents unintended exposure of data
  - Hidden/Disabled Controls
  - Restrict Data Operations using properties

- The redaction library can fix most if not all issues but, as it is very developer-error prone, it is highly recommended to upgrade to at least 20.0!

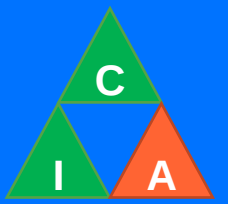


- { WebProperty=ClientProtected }, Provides protection against client-sided attacks, no write allowed (20.0+)
- Also makes sure properties like pbRender, pbVisible, and pbEnabled cannot be changed

- Redaction Library users are ought to implement IsRedacted (<19.0), whereas 19.0+ users can set pbRedact to True
- Therefore, it is even recommended for the 18.2 to upgrade to at least 19.0!



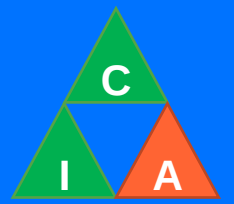
- Function IsControlAccessible (IsRedacted)
  Visible
  Rendered
  Enabled
- Property Boolean pbNoAccessibilityCheck (pbRedact/IsRedacted)
  - Specifies whether to apply the IsControl-API
- Procedure AllowServerAction
  - Checks before a Server Action is called whether the call is allowed



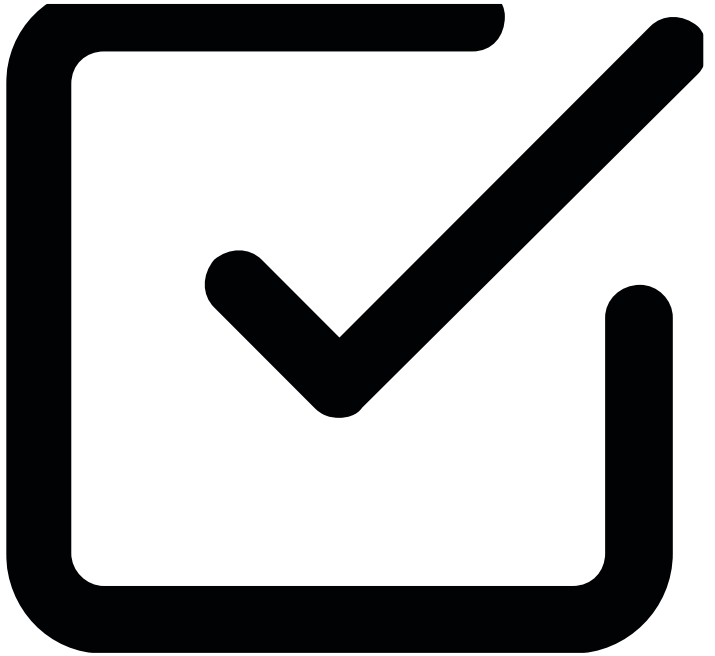
- pbNoUpdateIfHidden (cWebBaseDEO) (pbNoUpdateIfRedacted)
  - Controls if hidden DEO's write values to the DD
- pbNoFillIfHidden (cWebBaseDEO) (pbNoFillIfRedacted)
  - Controls if hidden DEO's read values from the DD

- AllowUpdateDD (cWebBaseDEO)
- AllowFillDEO (cWebBaseDEO)

LIBRARIES
REDACTION LIBRARY

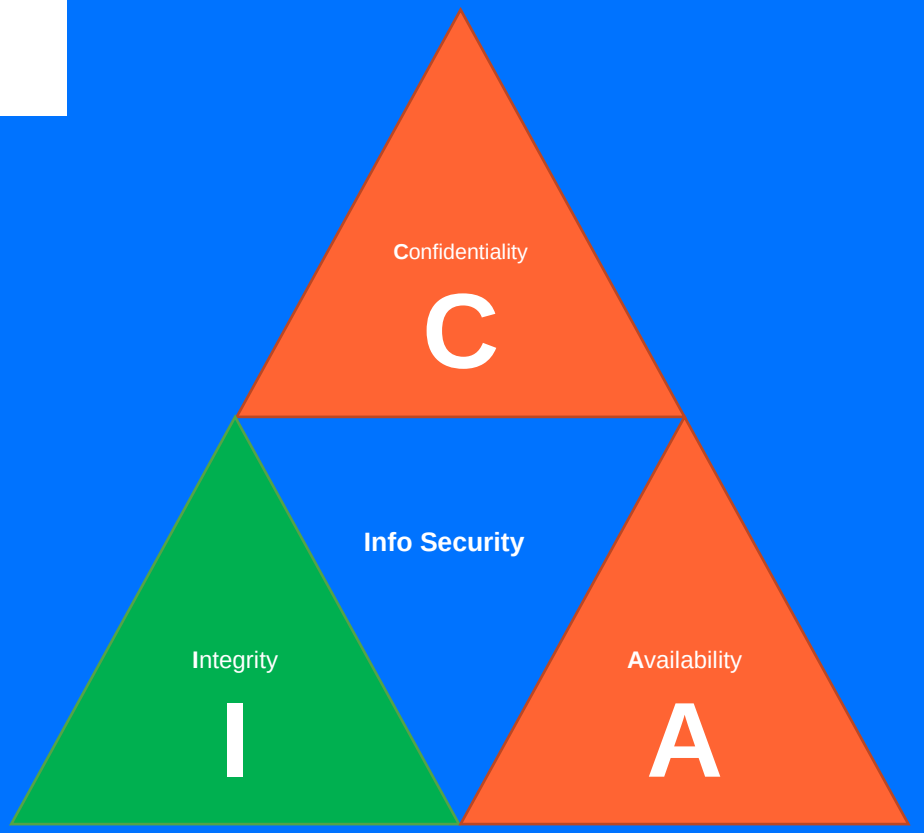


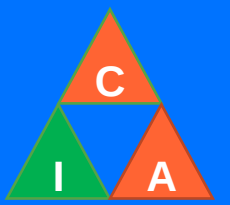
- In conclusion, not using this library is very serious as the trainees can attest to, as they have exploited the vulnerability themselves
- 18.2 should upgrade to 19.x to avoid developer-errors
- 20.0+ secures by default



LIBRARIES

XSS SANITIZER





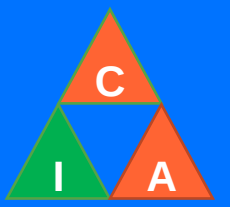
- Can be great for styling and customizability, But also one of an application's biggest security holes
- Cause you can inject code into HTML and is much like SQL injection:

```
<div>{{html}}</div>

Could become:

<div><script>alert('hello world')</script></div>
```

XSS SANITIZER - WHAT CONTROLS USE HTML?

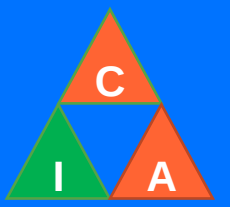
- cWebHtmlBox
- cWebHtmlList
- And any pbAllowHtml-enabled control:
  - cWebColumn
  - cWebTreeView

LIBRARIES
XSS SANITIZER - HOW DO I SANITIZE THESE?



- Use pbAllowHtml only in controlled situations
- HtmlEncode values if you get HTML from the client

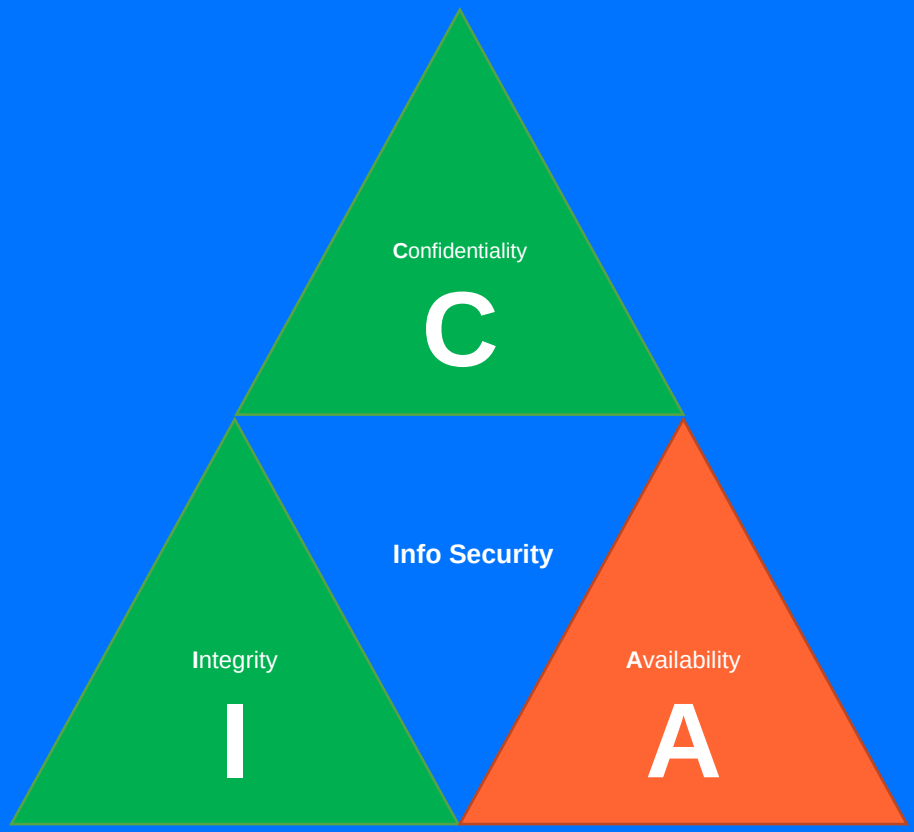
LIBRARIES
XSS SANITIZER - AND IN CUSTOM CONTROLS?



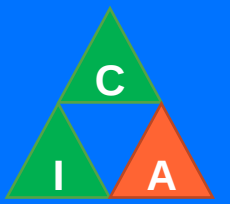
- Use the cXssSanitizer to sanitize just some elements or attributes, like in an HTML editor for example
- Available as a library already!

LIBRARIES

SECURITY LIBRARY

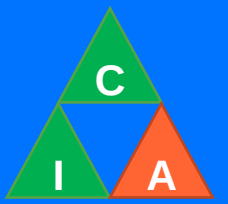


LIBRARIES
SECURITY LIBRARY



- Security algorithms
- Hashing
- Encryption
- Password hashing for DataFlex WebApps
- Two factor authentication for DataFlex WebApps

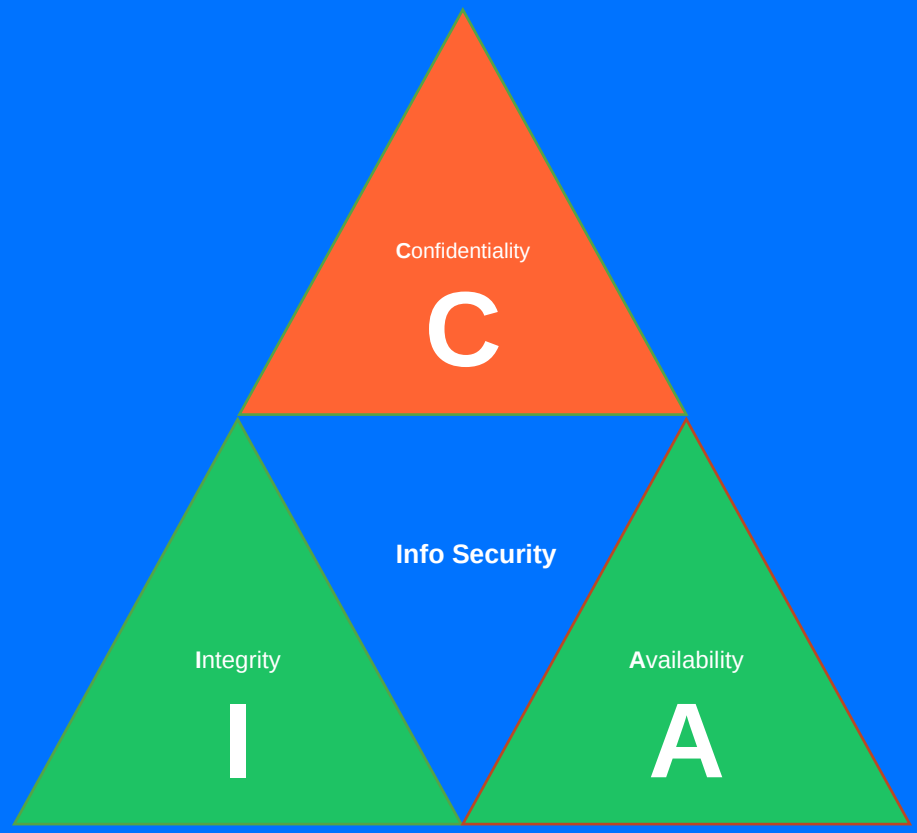
LIBRARIES
SECURITY LIBRARY



- DfSecurity-CNG Library
  - Replaces the old Crypto API (cCryptographer)
- DfSecurity-Libsodium
  - Open-source cryptography library
  - Comes as DLL with the library
- Both engines can be mixed in a single application

NEW LIBRARIES TO COME

SECURE UPLOAD



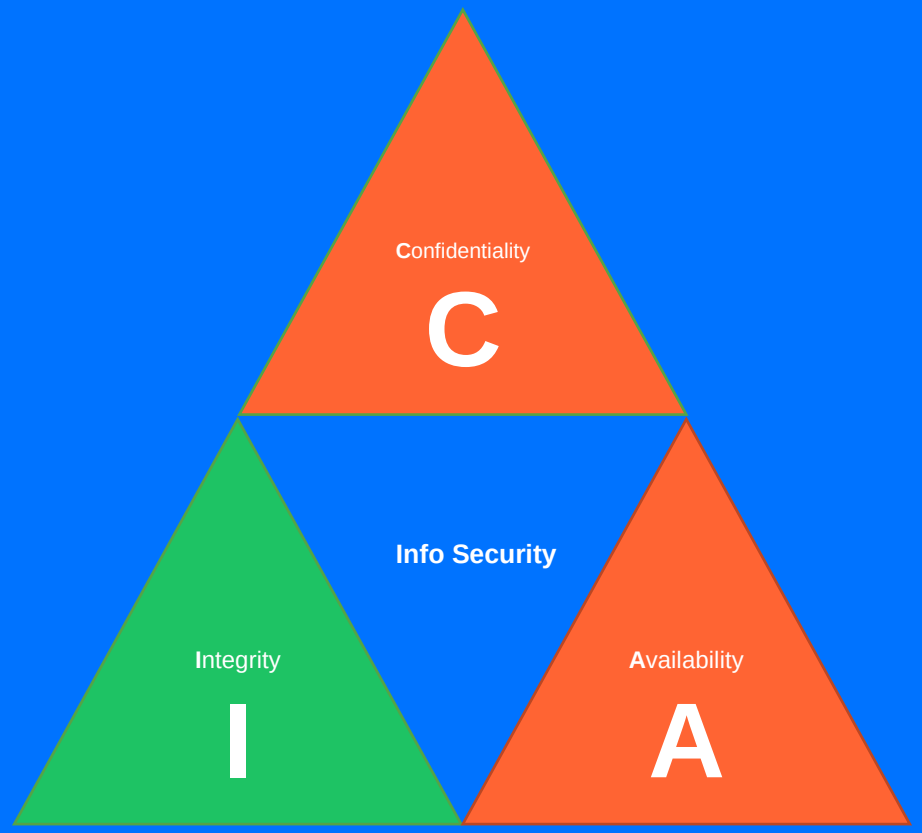
NEW LIBRARIES TO COME
SECURE UPLOAD

- File uploading can be a tricky business
- While this might not be too big an issue for an internal applications
- As you wish to scale or build a SaaS-product, you need to take more into account
- You generally only want to allow a specific type of file, and off course need some kind of protection against viruses, malware, and other threats

THAT’S WHY WE ARE WORKING ON SOME ADDITIONAL SECURITY FOCUSED LIBRARIES!

NEW LIBRARIES TO COME

LIBMAGIC WRAPPER



NEW LIBRARIES TO COME
LIBMAGIC

- When a file is being uploaded, you can never be sure of what you will be receiving as a server
- This is because browsers are the only ones enforcing http rules, but as you know http requests are merely some bytes flying over the line


```
POST /upload HTTP/1.1
Host: www.example.com
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Length: 12345

------WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="file"; filename="example.png"
Content-Type: image/png

[This can be a virus or .EXE file]

------WebKitFormBoundary7MA4YWxkTrZu0gW--
```


```
POST /upload HTTP/1.1
Host: www.example.com
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Length: 12345

------WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="file"; filename="example.png"
Content-Type: image/png

[This can be a virus or .EXE file]

------WebKitFormBoundary7MA4YWxkTrZu0gW--
```

- That means that both the psAccept property:

```
Set psAccept to "image/*,.jpg,.png,.bmp,.gif"
```

- As well as the sMime parameter:

```
Function OnFileUpload String sFileName Integer iBytes String sMime Returns String
```

- Can help, but should never be trusted


```
POST /upload HTTP/1.1
Host: www.example.com
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Length: 12345

------WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="file"; filename="example.png"
Content-Type: image/png

[This can be a virus or .EXE file]

------WebKitFormBoundary7MA4YWxkTrZu0gW--
```

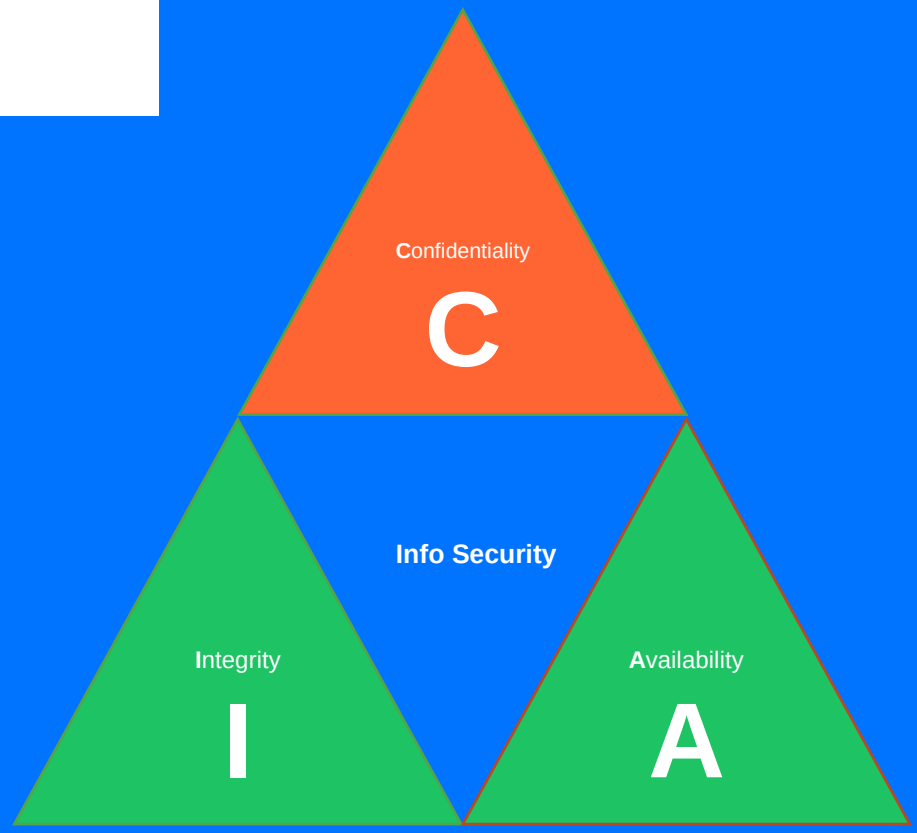
NEW LIBRARIES TO COME
LIBMAGIC

- This is what libmagic is for; it is a library that determines on multiple markers, based on the content, what type a given file is
- It is the library that operating systems like MacOS, Linux, and Unix use to determine which application to start
- As opposed to the .extention technique that Windows uses
- This means that when a file is uploaded you can check whether said file contains the content that you expect

DEMO!

NEW LIBRARIES TO COME

ANTI-VIRUS INTERFACE



NEW LIBRARIES TO COME
AV INTERFACE

- Before we open a file ourselves however, it would be better to let our AV look first
- We have created a small interface that allows an AV to be called and by default will call the often-built-in Windows Defender suite

DEMO!

NEW LIBRARIES TO COME
AV INTERFACE

- What we must remember is that touching AVs is and will remain tricky
- One thing that is important is to mark your upload directories as exceptions to default scanning and watching rules in Windows Defender
- Why? Because otherwise Windows Defender might find WebApp to be suspicious, kills it, and quarantines it
- This is something that happened to us during our first versions of this small library, it is important that the application takes full responsibility for managing its own uploads and data

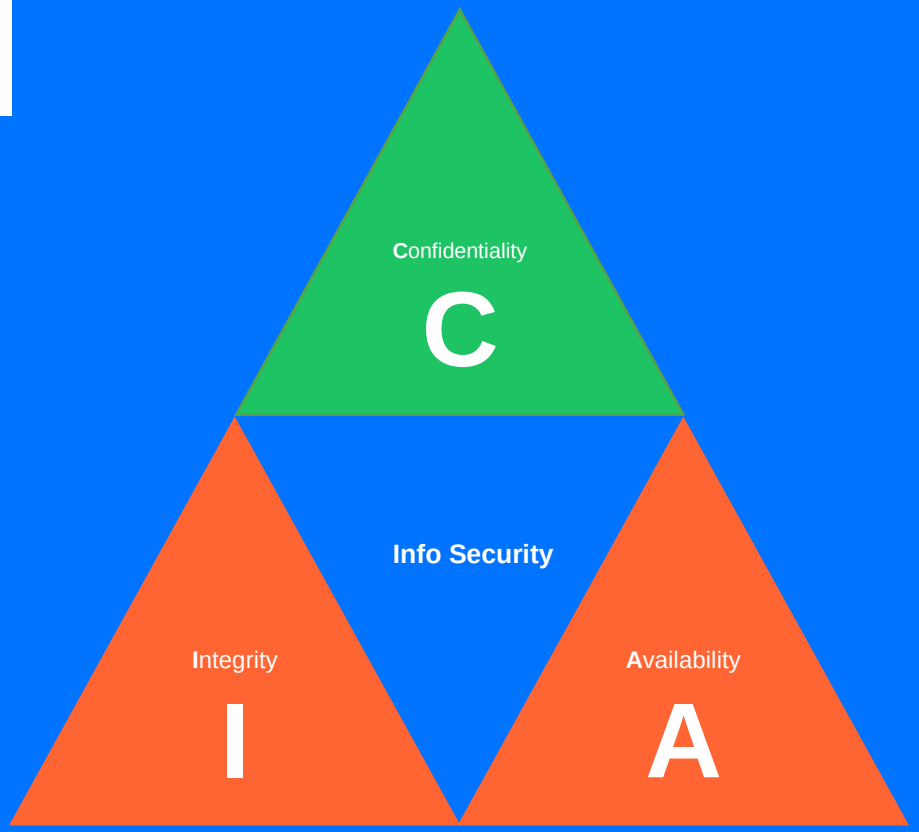
NEW LIBRARIES TO COME
SECURE UPLOAD

- The secure upload will be a combination of these libraries with some prefab upload components but with the option to implement them yourself as well
- Still finalizing this, as the depth and total amount of automatization

```
Procedure OnFileFinished String sFileName String sLocalPath
    Forward Send OnFileFinished sFileName sLocalPath

    Integer iErr
    Get AVCheck of ghoAVCheck sLocalPath to iErr
    If (iErr <> C_AV_Ok) Begin
        Send AVThrow of ghoAVCheck iErr sFileName
        EraseFile sLocalPath
        Procedure_Return
    End

    Get MagicMatchesPattern (psAccept(Self)) sLocalPath to iErr
    If (iErr <> C_MagicMatch_Match) Begin
        Send MagicThrow iErr sFileName
        EraseFile sLocalPath
        Procedure_Return
    End
    EraseFile sLocalPath
End_Procedure
```

SECURE BY DEFAULT

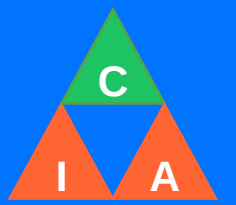
PASSWORD HASHING



SECURITY BY DEFAULT
PASSWORD HASHING

- Passwords and hashing are probably one of the biggest problems on this earth. Mainly because most programmers don’t know why they are using it. And, even if you do, are you sure?

SECURITY BY DEFAULT
PASSWORD HASHING



- A random string of characters, numbers, and whatnot.
- Ascii: 95 usable characters (66 commonly used)

CAN ANYBODY GUESS AS TO WHY PASSWORDS DON’T HAVE UNICODE SUPPORT

SECURITY BY DEFAULT
PASSWORD HASHING

- Cause Unicode encoding is a mess. There are multiple ways to get to a character. How it is encoded etc. Language usage and last but not least, what encoding the browser and system use!

Which Characters Are Most Popular in Passwords?

Based on most common characters within passwords in a breach of over 30 million accounts.

SecurityScorecard

Alphanumeric

0	1	2	3	4	5
6	7	8	9	A	B
C	D	E	F	G	H
I	J	K	L	M	N
O	P	Q	R	S	T
U	V	W	X	Y	Z

Symbols

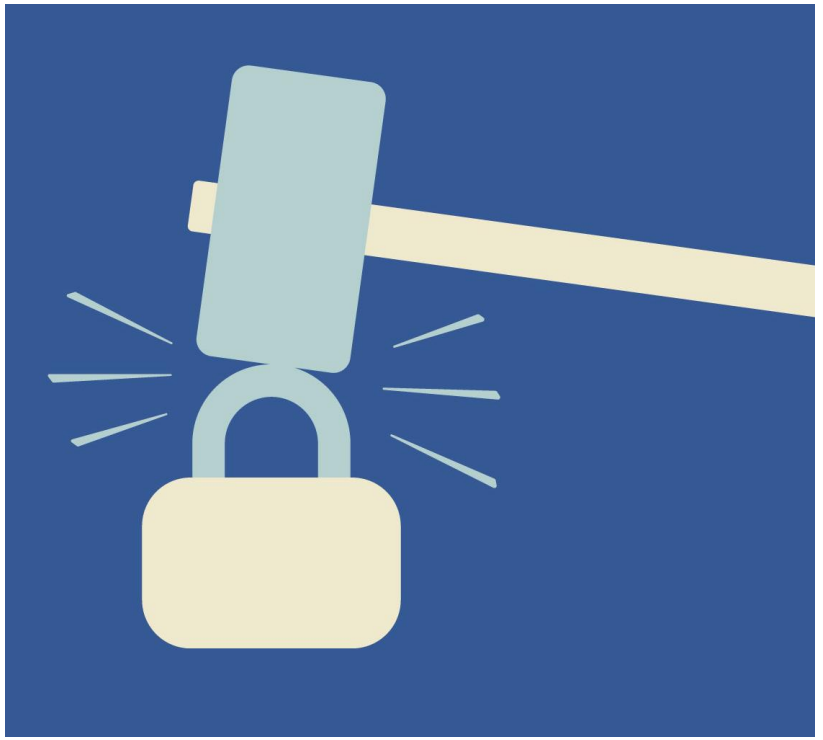
`	~	!	@	#	$
%	^	&	*	(	)
-	_	=	+	[	{
]	}	\	|	;	:
‘	“	<	.	>	/
?					

SECURITY BY DEFAULT
PASSWORD HASHING

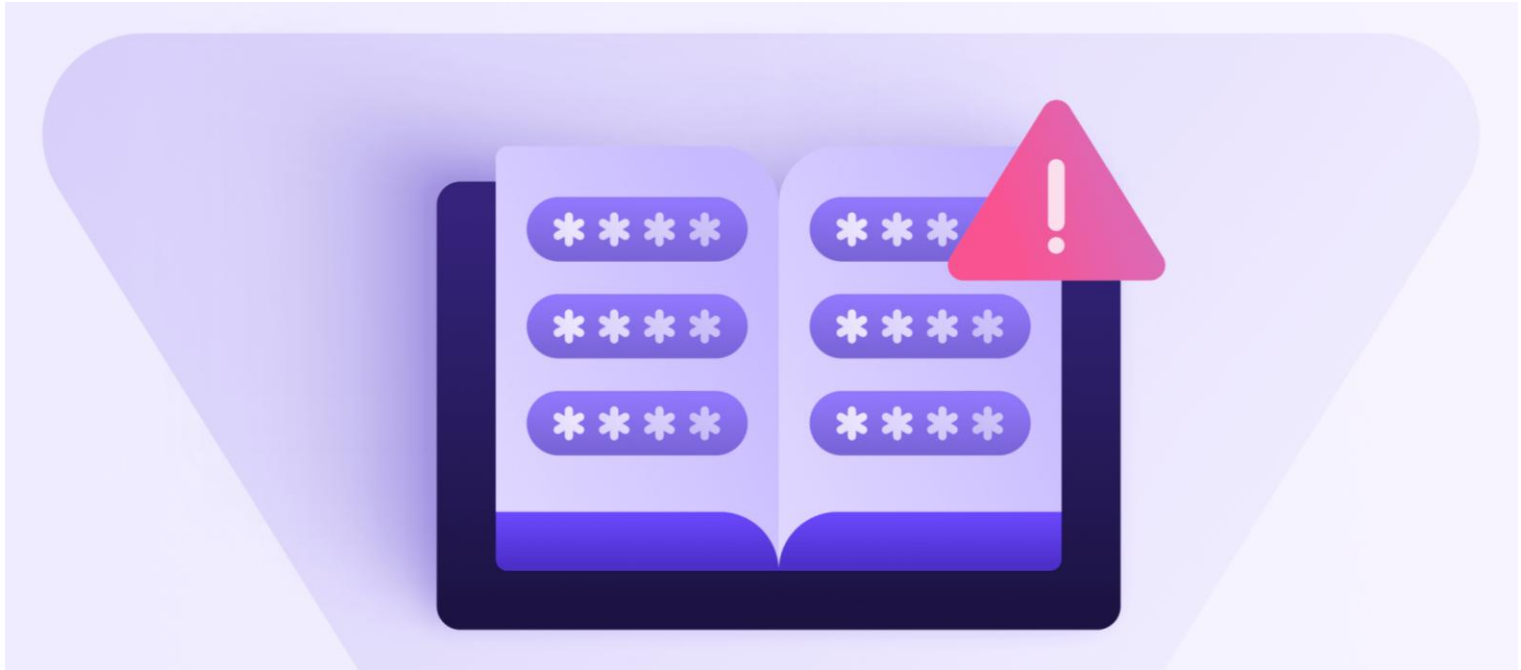
- That does not mean restricting them with whatever...
- Minimal one number, 3 special, 14 characters and if you are up to it an ‘@’....
- Helps with that already minimal set

SECURITY BY DEFAULT
PASSWORD HASHING

- Brute-Force
  - A…
  - B…
  - AB…
  - ABA…
- Attempting every option until one is found.

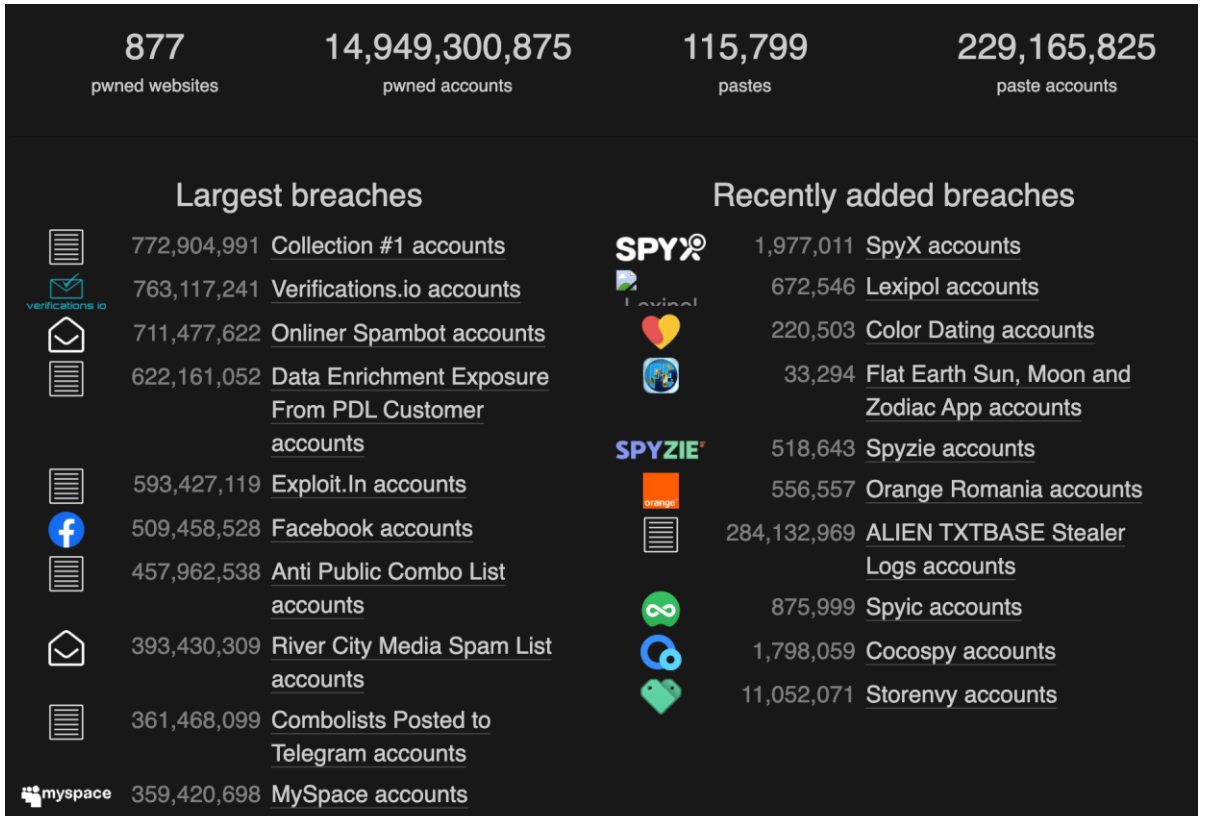


- Dictionary Attack
  - Use a table to try some common passwords
    - Iloveyou
    - 12345
    - god
    - …

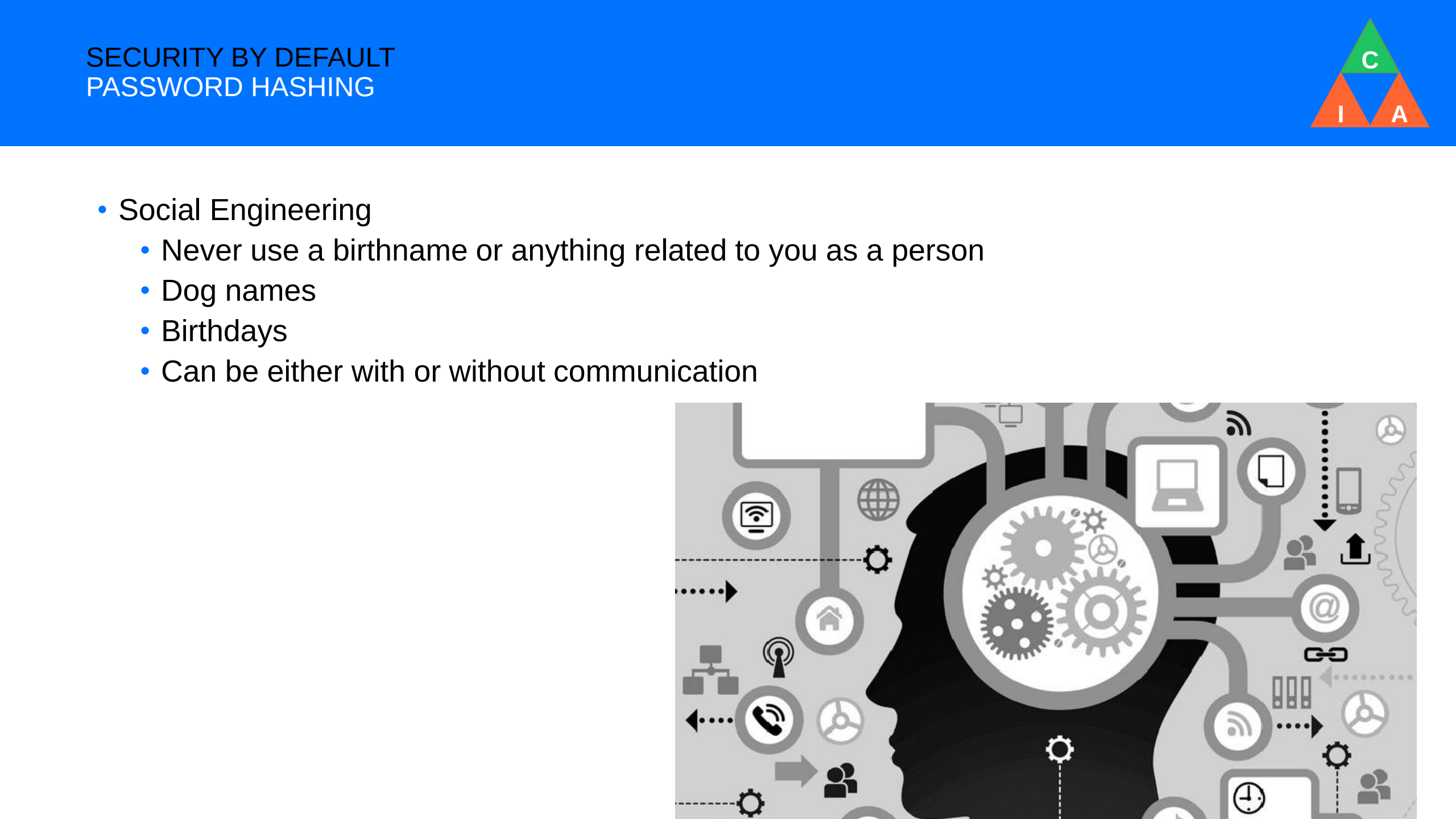


SECURITY BY DEFAULT
PASSWORD HASHING

- Credential Stuffing
  - Is a dictionary attack but using data breaches
  - Databases can be bought and found everywhere
  - That means if you use the same password everywhere?
  - https://haveibeenpwned.com/



SECURITY BY DEFAULT
PASSWORD HASHING



- Social Engineering
  - Never use a birthname or anything related to you as a person
  - Dog names
  - Birthdays
  - Can be either with or without communication

SECURITY BY DEFAULT
PASSWORD HASHING

- So, what can we do to protect the users as best we can?
  - It’s all about time;
  - By forcing a minimum number of characters, which increases brute-force time astonishingly.
  - Using complex and expensive (time) hashing algorithms instead of cheap ones.
  - Using salts… to differentiate between the same passwords.

SECURITY BY DEFAULT
PASSWORD HASHING

- There is a difference; password hashing vs integrity!
  - If you want a checksum of a file, database row, anything to validate integrity use the most failsafe one. You don’t need salting or anything like that.
  - Recommendation is then to use the fastest there is:
    - BLAKE2/3
    - SHA-1

Algorithm	Hashes per second
MD4 (Integrity) (collisions)	137.9 GH/s
MD5 (Integrity) (collisions)	76,526.9 MH/s
SHA1 (Integrity)	25,963.3 MH/s
SHA256 (Integrity)	9,392.1 MH/s
SHA512 (Integrity)	3,235.0 MH/s
SHA-3(Keccak) (Integrity)	2,500.4 MH/s
PBKDF2-HMAC-SHA256 (Password) (Customizable)	3,582.7 kH/s*
PBKDF2-HMAC-SHA512 (Password) (Customizable)	1,303.1 kH/s*
bcrypt, Blowfish (OpenBSD) (Password)	43,551 H/s
scrypt (Password)	1,872.4 kH/s
Argon2 (Password) (Customizable)	2.64 H/s*

SECURITY BY DEFAULT
PASSWORD HASHING

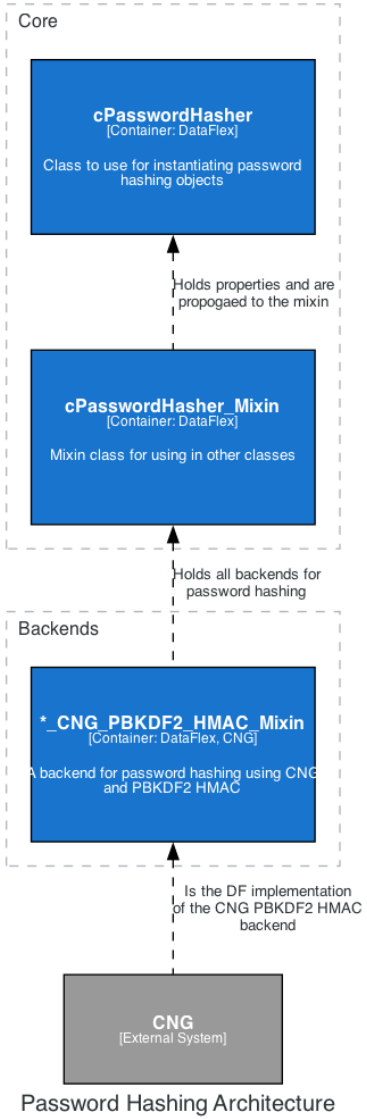
- The facts:
  - PBKDF2 is required for FIPS certification.
  - PBKDF2 is approved by NIST.
  - Argon2 is approved by a lot of people but not all (depends who you ask).

SECURITY BY DEFAULT
PASSWORD HASHING

- As some may have noticed in DataFlex 25.0 we set another step into the direction of secure by default.
- That means that by default DataFlex 25.0 projects will use built-in password hashing.
- Some important choices:
  - We want it to work without dependencies on say a Security Library (nothing changes there!)
  - Design needs to be portable looking at Tech Stack and cross-platform compatibility
  - Auto upgrading passwords
  - Dependency-Injecting style using Mixins
  - We chose for FIPS and NIST support, meaning PBKDF2 over Argon2

SECURITY BY DEFAULT
PASSWORD HASHING

- What does that mean?
  - A new project will automatically use the new hashing techniques using the cWebSessionManagerStandard
  - cWebSessionManagerStandard does version control since it can access the database
  - Backends do the hashing specifics
  - The Mixin does the stringifying and is purely parameterized
  - The cPasswordHasher holds settings and can thus be instantiated



Password Hashing Architecture

SECURITY BY DEFAULT
PASSWORD HASHING

- Things to know:
  - pbPasswordHashing indicates whether to use password hashing by default
  - Abstractions should keep on working, if you already do password hashing, that's completely fine. It might even be better depending on the use case
  - Migration is required if you don't already use password hashing, otherwise you will simply not be able to login
    - The What's New 2025 has an article on the added Password Hashing functionality and provides an example on how to migrate
  - The Password Hashing is reusable through cPasswordHasher.pkg

SHOWCASE

AND ONCE AGAIN TO STRESS, PLEASE KEEP UP TO DATE WITH BOTH DATAFLEX AND ITS FRAMEWORKS IF YOU CAN!

ARE THERE ANY QUESTIONS?