Cybersecurity mechanics

SECRIOUS
PROJECT

UKRI Engineering and Physical Sciences Research Council

Project ID : EP/T017511/1

Cybersecurity mechanics

SECRIOUS
PROJECT

UKRI Engineering and Physical Sciences Research Council

Project ID : EP/T017511/1

1

IDENTITY FORGERY

DESCRIPTION

Use a fake certificate signed by rogue certificate authorities to forge the identity of legitimate services or users

IMPACT

Make user believe the service or website is from legitimate provider

RELATED VULNERABILITIES

2

3

3

2

MAN-IN-THE-MIDDLE ATTACK

DESCRIPTION

Intercept and replace the certificate or public key of one party in a communication to fake their identity in the communication

IMPACT

Observe or alter data transferred between the communicating parties

RELATED VULNERABILITIES

1

3/4

6

3

1

RAGE ATTACK

DESCRIPTION

An insider abuses its legitimate access to the services for redemption of rage or fire

IMPACT

All services accessible by a rage insider or stakeholder

RELATED VULNERABILITIES





1/2

2

SOCIAL ENGINEERING

DESCRIPTION

Use different means to lure legitimate users to give away their access or knowledge

IMPACT

Gain knowledge or access illegally

RELATED VULNERABILITIES

1

1

5

1

SQL INJECTION

DESCRIPTION

Malicious SQL (Structured Query Language) statement inserted into entry fields for illegal execution to attack data-driven services

IMPACT

Pollute the database or leak out data from the database

RELATED VULNERABILITIES

1/2/3

2

COMMAND / DATA INJECTION

DESCRIPTION

Malicious command inserted into entry fields for illegal or unexpected execution on underlying systems or clients

IMPACT

Pollute system or other clients, or gain illegal access and knowledge

RELATED VULNERABILITIES

2/3/7

3

CROSS SITE REQUEST FORGERY

DESCRIPTION

Make a victim's computer submit a request to a legitimate service on the victim's behalf in the background without their knowledge or consent

IMPACT

Web request with victim's identity without their consent

RELATED VULNERABILITIES

7/8

CODE CORRUPTION

1

DESCRIPTION

Modify compiled program codes or variables to change the execution behaviour or to simply crash the program

IMPACT

Change the service behaviour

RELATED VULNERABILITIES

4 2

2

CONTROL FLOW HIJACKING

DESCRIPTION

Modify the execution flow of a compiled program by redirecting memory pointers to other memory locations

IMPACT

Altered program execution or data leakage

RELATED VULNERABILITIES




4

2

3

PRIVILEGE ESCALATION

DESCRIPTION

Modify certain privilege variables in order to gain higher-privilege illegally or trick legitimate higher-privilege services to complete a request on behalf of lower-privilege service

IMPACT

Bypassing access control and privilege system

RELATED VULNERABILITIES



3

4/5

2

4

MEMORY THIEF

DESCRIPTION

Invalid memory access that attempts to make the program leak information stored in memory which is normally illegal to access (example: HeartBleed)

IMPACT

Gain illegal knowledge of data stored in the memory

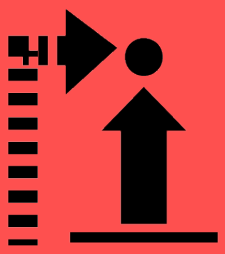
RELATED VULNERABILITIES



3

4/5

2



1

RACE CONDITION

DESCRIPTION

Abusing the gap between time-of-check to time-of-use (TOCTTOU)

IMPACT

Bypass checking to gain access to services and data

RELATED VULNERABILITIES

1/3

1

SIDE CHANNEL

DESCRIPTION

Observe the processing time or energy consumption to guess the correct data or secret

IMPACT

Gain knowledge illegally

RELATED VULNERABILITIES

4

2

2

FAULT INJECTION

DESCRIPTION

Purposely stress the target system with faulty configurations or execution variables to understand how the system behaves

IMPACT

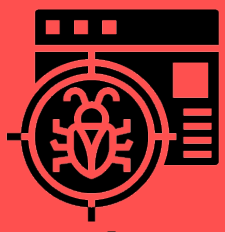
Gain knowledge of execution flow and potential vulnerabilities

RELATED VULNERABILITIES

4

2

1



1

ILLEGAL CODE EXECUTION

DESCRIPTION

Hide malicious code for it to be executed by the system or unaware users

IMPACT

Execute code without user consent or notice

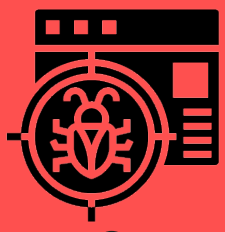
RELATED VULNERABILITIES

2

1

2/3

3



2

DISTRIBUTED DENIAL OF SERVICE (DDOS)

DESCRIPTION

Flood the system with communications to make it fail to response to legitimate user requests, generally the requests are initiated from a large set of infected systems

IMPACT

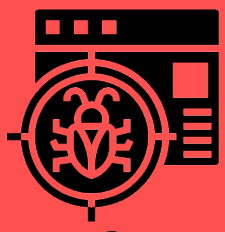
Make a service unavailable to legitimate users

RELATED VULNERABILITIES


2



1



3

REGISTRY OVERWRITING

DESCRIPTION

Hide the existence of illegal code by over-writing data in the computer registry or persistent memory

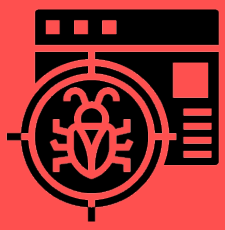
IMPACT

Control data flow and manipulate data

RELATED VULNERABILITIES

1

2



4

BRUTE FORCE

DESCRIPTION

Trying every possible combination of credentials

IMPACT

Gain knowledge or access illegally

RELATED VULNERABILITIES

1

2/4

1

1

PARAMETER MANIPULATION

DESCRIPTION

Replace user input parameter sent in plain-text to change the system behaviour or get knowledge of the user choice or secret information

IMPACT

Pollute the user input or get knowledge of user input illegally

RELATED VULNERABILITIES

1

3/4

2

ILLEGAL DOWNLOAD

DESCRIPTION

Perform automatic downloads on behalf of the user without consent or notice

IMPACT

Malware entering the user's computer

RELATED VULNERABILITIES

1/4

7/8

1

CODE ASSERTIONS AND REVIEWS

DESCRIPTION

Add assertive statement to service code to verify the status of certain variables during execution to ensure it runs as expected

RELATED VULNERABILITIES

1

3/4

1/2/3/4/
5/6/7/8

1

2

OWNERSHIP VERIFICATION

DESCRIPTION

Verify the ownership and related information of an artefacts just before use to avoid race condition and last minutes changes

RELATED VULNERABILITIES

2

1/3

7

3

3

ALIAS TRANSLATION

DESCRIPTION

Resolve aliases of artefacts and review their absolute paths to ensure they are not pointing to unexpected locations or artefacts

RELATED VULNERABILITIES

2

1/3

4

MONITORING

DESCRIPTION

Background monitoring of the execution, memory and network traffic to detect abnormal behaviour

RELATED VULNERABILITIES

1

1

3

3

5

STACK CANARIES

DESCRIPTION

Add static specific values in specific address in the memory. Check if they have been modified by illegal memory overflow before executing or using something from memory

RELATED VULNERABILITIES

2

6

EXECUTION INTEGRITY

DESCRIPTION

Make a copy of the execution flow before execution and use it to verify if the control flow has been illegally altered

RELATED VULNERABILITIES

2

7

TAINT ANALYSIS

DESCRIPTION

Scan the service and program to detect any injection vulnerability pattern in source code that accept malformed user input

RELATED VULNERABILITIES

1

1

1/2/3/4/ 5/7/8

3

8

PENETRATION TESTING

DESCRIPTION

Simulate cyber attack against your computer systems or services to discover exploitable vulnerabilities and put in place necessary security measures

RELATED VULNERABILITIES

1/2

1/2/4

1/2/3

1/3

9

MALWARE SCANNING

DESCRIPTION

Match processes with known malicious behaviour to detect possible hidden malware in the system or memory

RELATED VULNERABILITIES

1

1

3

3

CODE GUIDELINES

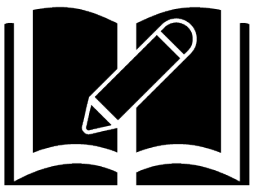
1

DESCRIPTION

Provide secure coding guidelines and requirements for software development

RELATED VULNERABILITIES

1/2/3/4/7



2

SECURITY AWARENESS TRAINING

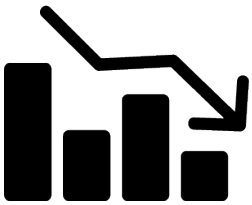
DESCRIPTION

Provide general security awareness training for users to educate them on possible threats, security pitfall and measures

RELATED VULNERABILITIES

1/2

1/3/4



1

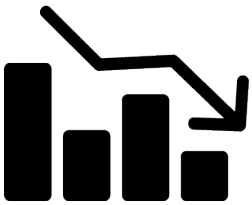
SAFE LANGUAGE AND DEFENSIVE PROGRAMMING

DESCRIPTION

Use prrogramming languages and practices with safer settings and avoid or verify the use of dangerous functions or unsafe memory operations

RELATED VULNERABILITIES

4/5/8



2

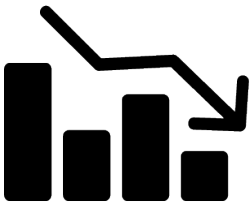
INITIALISATION OF RESOURCE ACQUISITION

DESCRIPTION

Initialise resources and variables to protect against attacks directed to un-initialised or default initialisation of variable and resources

RELATED VULNERABILITIES

5/6



3

USE APPROVED TOOLS

DESCRIPTION

Use approved tools which have been re-
pletely analysed and verified

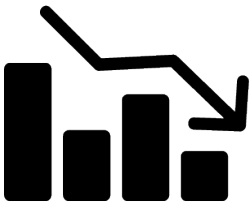
RELATED VULNERABILITIES

3/4

1/2/3

4/6/8

1/2/3



4

SANDBOXING

DESCRIPTION

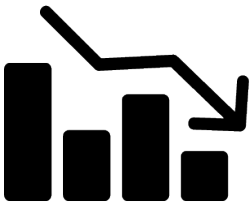
Logically separating memory, resources and processes of different services to avoid the spreading of malicious activity or privilege escalation

RELATED VULNERABILITIES

1

1/2

3



5

PRINCIPLE OF LEAST PRIVILEGE

DESCRIPTION

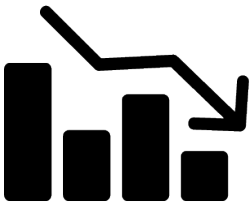
Only provide the least privilege to service for completing their necessary execution to limit the possible damage when hijacked

RELATED VULNERABILITIES

1/2

1/2

3



6

CONTRACTS

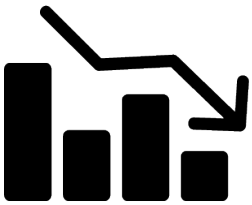
DESCRIPTION

Sign contracts with contractor to state their responsibility in putting in place security features in their services or codes and to state their liability in case of incident

RELATED VULNERABILITIES

1/2

1/3/4



7

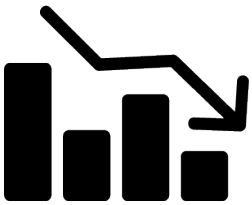
SAME-ORIGIN POLICY

DESCRIPTION

Only allow a web service to execute scripts from the same domain

RELATED VULNERABILITIES

3 2/3/6/7



8

CONTENT SECURITY POLICY

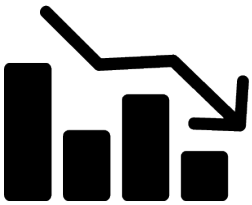
DESCRIPTION

Refuse executing script or code from user-generated section in a web service

RELATED VULNERABILITIES

1/4

2/3



9

SECURE SOFTWARE DEVELOPMENT LIFE CYCLE

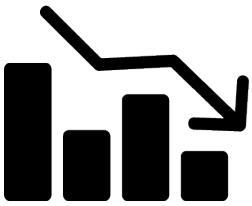
DESCRIPTION

Include security measures in each process of the software development life cycle

RELATED VULNERABILITIES

1/3/4

3



10

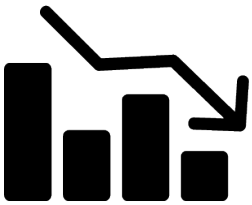
ADDRESS SPACE LAYOUT RANDOMIZATION

DESCRIPTION

Randomize the address space to avoid easy guessing of the arrangements and location of critical data and execution commands

RELATED VULNERABILITIES

2



11

LOAD BALANCERS

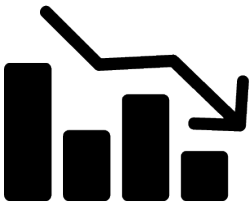
DESCRIPTION

Install a proxy in front of main service machine to equally distribute (or reject) request load to multiple machine to avoid flooding of request

RELATED VULNERABILITIES

2

1



12

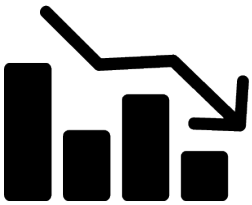
PERMISSION DIALOG BASED ACCESS CONTROL

DESCRIPTION

Popup and warn the user before executing some dangerous process and obtain the user consensus before executing them

RELATED VULNERABILITIES

1/4



13

SECURITY HYGIENE

DESCRIPTION

Maintain code quality and security practices on both development and normal working environment to decrease risk from system developed and physical working environment

RELATED VULNERABILITIES

1

1/4

1

CORRECT USE OF FUNCTIONS

DESCRIPTION

Follow guidelines of library or API function calls to ensure security standard and requirement are met

RELATED VULNERABILITIES

4/8

3

2

INPUT SANITISATION AND FILTERING

DESCRIPTION

Check, clean and filter special characters from user input to avoid polluted input to get accidently processed, executed or stored

RELATED VULNERABILITIES

1/2/3/6/7

3

ONE-TIME PASSWORDS

DESCRIPTION

Introduce the use of one-time password to deny replay attack from attacker which attempt to use previously known password

RELATED VULNERABILITIES

2

4

NON-EXECUTABLE MEMORY AND IMMUTABLE STATE

DESCRIPTION

Deny executing commands or changing of states if it is stored in protected memory section or immutable variables

RELATED VULNERABILITIES

8

2

5

TOKENISATION OF SENSITIVE DATA

DESCRIPTION

Generate undecipherable tokens for sensitive data when transferred through insecure channels

RELATED VULNERABILITIES

1 1/4 6/7/8

6

SECURE UPGRADE PROCESS

DESCRIPTION

Implement a process to preserve the integrity of packages when deployed into existing services

RELATED VULNERABILITIES

1/3/4

Cybersecurity mechanics

SECRIOUS
PROJECT

UKRI Engineering and Physical Sciences Research Council

Project ID : EP/T017511/1

INCORRECT CONSTRUCTION OF SQL STATEMENT

1

DESCRIPTION

Incorrect construction of Structured Query Language (SQL) statement allows for potentially malicious user input to be executed

RELATED ATTACKS

1

RELATED DEFENCES

1/7

1

2

INCORRECT SANITIZATION OF STORED DATA

2

DESCRIPTION

Fail to check and sanitize special characters or commands that originate from user input which have ended up in storage and can lead to malicious code execution

RELATED ATTACKS

1/2 1

RELATED DEFENCES

1/7 1 7/8 2

IMPROPER INPUT VALIDATION

3

DESCRIPTION

Input fields are not validated leaving execution of input data vulnerable to malicious input

RELATED ATTACKS

1/2

1

1

RELATED DEFENCES

1/7

1

7/8

2

MISUSE OF DANGEROUS FUNCTIONS

4

DESCRIPTION

Ignoring implementation requirements for dangerous functions which may lead to memory and execution vulnerabilities

RELATED ATTACKS

1/2/3/4

1

RELATED DEFENCES

1/7

1

1/3

1

MISSING DEFAULT INITIALISATION FOR INPUT

5

DESCRIPTION

Abuse of non-typed systems and sending in much larger data to an uninitialised variable causing integer or variable overflows

RELATED ATTACKS

2

3/4

RELATED DEFENCES

1/7

1/2

SENDING UNPROTECTED PARAMETERS

6

DESCRIPTION

Acceptance of plaintext (unprotected) parameters, exposing potentially sensitive data to attackers

RELATED ATTACKS

2

RELATED DEFENCES

1

2/3/7

2/5

7

INADEQUATE DATA AUTHENTICITY AND ORIGIN VERIFICATION

DESCRIPTION

No verification of whether data was sent by a legitimate sender in some legitimate domain

RELATED ATTACKS

2/3

2

RELATED DEFENCES

1/2/7

1

7

2/5

8

INADEQUATE SESSION EXPIRATION

DESCRIPTION

Old or expired session information (e.g. request credentials) is not identified and verified, leaving requests for information vulnerable to attackers who may capture the session information

RELATED ATTACKS

3

2

RELATED DEFENCES

1/7

1/3

1/4/5

1

RACE CONDITION

DESCRIPTION

No verification of time checks and resource usage in programs, allowing attackers to modify the program's behaviour or its resources

RELATED ATTACKS

1

RELATED DEFENCES

2/3/7/8

3

2

SIDE CHANNEL

DESCRIPTION

Monitoring normal system behaviour through physical or digital means, such as analysing network traffic, could leak sensitive information

RELATED ATTACKS

1/2

RELATED DEFENCES

8

11/3

3

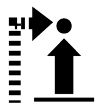
WEAK SERVICES OR PROCESSES ISOLATION

DESCRIPTION

Lack of authentication and verification off the origin and destination of requests can result in code being executed on the same system

RELATED ATTACKS

3/4



1

RELATED DEFENCES

2/3/4/8/9

3/4/5/7/9

BAD CREDENTIALS HANDLING

1

DESCRIPTION

Using default or easy-to-guess passwords, or writing them down on paper, can leave password-protected accounts vulnerable to attackers

RELATED ATTACKS

2 2 3/4

RELATED DEFENCES

1/4/7/8/9 2 13/4/5/6 5

2

INAPPROPRIATE MANAGEMENT DECISION

DESCRIPTION

Non-expert management may decide to skip parts of risk analysis and security implementations due to time or budget pressures

RELATED ATTACKS

1

1

RELATED DEFENCES

8

2

5/6

LOAD BALANCING FAILURE

1

DESCRIPTION

Failure to handle large volumes of traffic and/or balance large volumes of traffic properly can result in failures to service legitimate users

RELATED ATTACKS

2 2/4

RELATED DEFENCES

1/8 11/3

2 NO EXECUTION CONTROL IN MEMORY

DESCRIPTION

Failures to deny code execution which involves critical memory can result in malicious code execution or leakage of sensitive information

RELATED ATTACKS

1/2/3/4

3

RELATED DEFENCES

5/6

3/10

3/4

3

BAD CERTIFICATE MANAGEMENT

DESCRIPTION

Blindly trusting the authenticity of certificates and the authority who signed them could allow them to spy on communications

RELATED ATTACKS

1/2

1

RELATED DEFENCES

2/4/7/8/9

3

1

CARELESS INSIDER

1

DESCRIPTION

Legitimate users can fail to properly handle (e.g. lose) credentials, equipment or data due to a lack of care or attention to detail

RELATED ATTACKS


1/2


1


1/2

RELATED DEFENCES


4/8/9


2


12/13/4/5/6/8/9


5/6

2

INSIDER ACTIONS

DESCRIPTION

Legitimate users may be provoked to act maliciously, such as using their privilege to leak sensitive information to competitors or tamper with internal services

RELATED ATTACKS

1

2/4

RELATED DEFENCES

2/3/8

4/5

DEVELOPER BIAS

3

DESCRIPTION

Developers may wrongly judge the necessity of certain security features and miss out on verify security protection or risks involved

RELATED ATTACKS

1/2

RELATED DEFENCES

1

2

3/6/9

6

4

SECURITY FATIGUE

DESCRIPTION

Reluctance to deal with cybersecurity can lead to skipping key security decisions or updates, increasing the likelihood of attacks to happen

RELATED ATTACKS

2

1/2

4

2

RELATED DEFENCES

1/8

2

12/13/3/6/8/9

5/6