**REPORT**

**SANS Top 20 Security Vulnerabilities In Software Applications**

[**https://www.softwaretestinghelp.com/sans-top-20-security-vulnerabilities/**](https://www.softwaretestinghelp.com/sans-top-20-security-vulnerabilities/)

1. **Vulnerability Name:** Memory Buffer Error

**CWE:** **CWE-119**

**SANS Category:** A01:2021- Memory Buffer Error

**Description:**

The product performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary of the buffer.

**Business Impact:**

Execute Unauthorized Code or Commands; Modify Memory

If the memory accessible by the attacker can be effectively controlled, it may be possible to execute arbitrary code, as with a standard buffer overflow. If the attacker can overwrite a pointer's worth of memory (usually 32 or 64 bits), they can redirect a function pointer to their own malicious code. Even when the attacker can only modify a single byte arbitrary code execution can be possible. Sometimes this is because the same problem can be exploited repeatedly to the same effect. Other times it is because the attacker can overwrite security-critical application-specific data -- such as a flag indicating whether the user is an administrator.

Read Memory; DoS: Crash, Exit, or Restart; DoS: Resource Consumption (CPU); DoS: Resource Consumption (Memory)

Out of bounds memory access will very likely result in the corruption of relevant memory, and perhaps instructions, possibly leading to a crash. Other attacks leading to lack of availability are possible, including putting the program into an infinite loop.

Read Memory

In the case of an out-of-bounds read, the attacker may have access to sensitive information. If the sensitive information contains system details, such as the current buffers position in memory, this knowledge can be used to craft further attacks, possibly with more severe consequences.

1. **Vulnerability Name:** Cross-site Scripting

**CWE:** CWE-79

**SANS Category:** A02:2021- Cross-site Scripting

**Description:**

The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

**Business Impact:**

Security-Database help your corporation foresee and avoid any security risks that may impact your IT infrastructure and business applications.

1. **Vulnerability Name:** Unvalidated Input Error

**CWE:** CWE-20

**SANS Category:** A03:2021- Unvalidated Input Error

**Description:**

The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

**Business Impact:**

When software does not validate input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution.

1. **Vulnerability Name:** Sensitive Information Exposure Error

**CWE:** CWE-200

**SANS Category:** A04:2021- Sensitive Information Exposure Error

**Description:**

The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

**Business Impact:**

Potential impact can vary depending on application, environment and other circumstances. There are many different problems that involve information leaks and their severity can be widely ranged based on information that is disclosed.

1. **Vulnerability Name:** Out-of-bounds Read Error

**CWE:** CWE-125

**SANS Category:** A05:2021- Out-of-bounds Read Error

**Description:**

The product reads data past the end, or before the beginning, of the intended buffer.

**Business Impact:**

Typically, this can allow attackers to read sensitive information from other memory locations or cause a crash. A crash can occur when the code reads a variable amount of data and assumes that a sentinel exists to stop the read operation, such as a NUL in a string. The expected sentinel might not be located in the out-of-bounds memory, causing excessive data to be read, leading to a segmentation fault or a buffer overflow. The product may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent read operation then produces undefined or unexpected results.

1. **Vulnerability Name:** SQL Injection

**CWE:** CWE-89

**SANS Category:** A06:2021- SQL Injection

**Description:**

The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component.

**Business Impact:**

Due to a major flaw in their systems, businesses can seriously be threatened by injection threats. These attacks make use of flaws in operating systems, databases, or online applications. This is done to allow hostile actors to insert and run unauthorized code or commands. A successful injection attack can have disastrous results and wide-ranging repercussions for the targeted company and its clients.

1. **Vulnerability Name:** Free Memory Error

**CWE:** CWE-416

**SANS Category:** A07:2021- Free Memory Error

**Description:**

Referencing memory after it has been freed can cause a program to crash, use unexpected values, or execute code.

**Business Impact:**

1. **Vulnerability Name:** Integer Overflow Error

**CWE:** CWE-190

**SANS Category:** A08:2021- Integer Overflow Error

**Description:**

The product performs a calculation that can produce an integer overflow or wraparound, when the logic assumes that the resulting value will always be larger than the original value. This can introduce other weaknesses when the calculation is used for resource management or execution control.

**Business Impact:**

this integer overflow bug caused a hit to the wider cryptocurrency market, causing Ethereum to dip from $664 to $612 in the course of a [single day](https://cryptoslate.com/batchoverflow-exploit-creates-trillions-of-ethereum-tokens/).

1. **Vulnerability Name:** Cross-Site Request Forgery

**CWE:** CWE-352

**SANS Category:** A09:2021- Cross-Site Request Forgery

**Description:**

The web application does not, or can not, sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request.

**Business Impact:**

 Depending on the application functionalities, an attacker might be able to gain access to otherwise restricted areas and perform simple tasks such as publishing comments under user's name, creating administrative accounts or use this weakness as a surface for further attacks.

1. **Vulnerability Name:** Directory Traversal

**CWE:** CWE-22

**SANS Category:** A10:2021- Directory Traversal

**Description:**

The product uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory, but the product does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory.

**Business Impact:**

An attacker can gain access to sensitive and system information on the system, delete or modify files. The maximum impact depends on the functionality of the application.

1. **Vulnerability Name:** OS Command Injection

**CWE:** CWE-78

**SANS Category:** A11:2021- OS Command Injection

**Description:**

The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component.

**Business Impact:**

The consequences of command injection attacks include data loss and integrity and unauthorized remote access to the machine that hosts the susceptible application.

1. **Vulnerability Name:** Out-of-bounds Write Error

**CWE:** CWE-787

**SANS Category:** A12:2021- Out-of-bounds Write Error

**Description:**

The product writes data past the end, or before the beginning, of the intended buffer.

**Business Impact:**

This can result in corruption of data, a crash, or code execution.

1. **Vulnerability Name:** Improper Authentication Error

**CWE:** CWE-287

**SANS Category:** A13:2021- Improper Authentication Error

**Description:**

When an actor claims to have a given identity, the product does not prove or insufficiently proves that the claim is correct.

**Business Impact:**

As an impact it is known to affect confidentiality, integrity, and availability.

1. **Vulnerability Name:** Dereferencing NULL Pointer

**CWE:** CWE-476

**SANS Category:** A14:2021- Dereferencing NULL Pointer

**Description:**

A NULL pointer dereference occurs when the application dereferences a pointer that it expects to be valid, but is NULL, typically causing a crash or exit.

**Business Impact:**

Depending on privileges of the application, this weakness can result in a denial of service attack against the entire system or can be used to gain complete control over it.

1. **Vulnerability Name:** Incorrect Permission Assignment

**CWE:** CWE-732

**SANS Category:** A15:2021- Incorrect Permission Assignment

**Description:**

The product specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

**Business Impact:**

Read Application Data; Read Files or Directories

An attacker may be able to read sensitive information from the associated resource, such as credentials or configuration information stored in a file.

Gain Privileges or Assume Identity

An attacker may be able to modify critical properties of the associated resource to gain privileges, such as replacing a world-writable executable with a Trojan horse.

Modify Application Data; Other

An attacker may be able to destroy or corrupt critical data in the associated resource, such as deletion of records from a database.

1. **Vulnerability Name:** Unrestricted File Upload

**CWE:** CWE-434

**SANS Category:** A16:2021- Unrestricted File Upload

**Description:**

The product allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment.

**Business Impact:**

Execute Unauthorized Code or Commands

Arbitrary code execution is possible if an uploaded file is interpreted and executed as code by the recipient. This is especially true for .asp and .php extensions uploaded to web servers because these file types are often treated as automatically executable, even when file system permissions do not specify execution. For example, in Unix environments, programs typically cannot run unless the execute bit is set, but PHP programs may be executed by the web server without directly invoking them on the operating system.

1. **Vulnerability Name:** Information Exposure through XML Entities

**CWE:** CWE-611

**SANS Category:** A17:2021- Information Exposure through XML Entities

**Description:**

The product processes an XML document that can contain XML entities with URIs that resolve to documents outside of the intended sphere of control, causing the product to embed incorrect documents into its output.

**Business Impact:**

Read Application Data; Read Files or Directories

If the attacker is able to include a crafted DTD and a default entity resolver is enabled, the attacker may be able to access arbitrary files on the system.

Bypass Protection Mechanism

The DTD may include arbitrary HTTP requests that the server may execute. This could lead to other attacks leveraging the server's trust relationship with other entities.

DoS: Resource Consumption (CPU); DoS: Resource Consumption (Memory)

The product could consume excessive CPU cycles or memory using a URI that points to a large file, or a device that always returns data such as /dev/random. Alternately, the URI could reference a file that contains many nested or recursive entity references to further slow down parsing.

1. **Vulnerability Name:** Code Injection

**CWE:** CWE-94

**SANS Category:** A18:2021- Code Injection

**Description:**

The product constructs all or part of a code segment using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the syntax or behavior of the intended code segment.

**Business Impact:**

Bypass Protection Mechanism

In some cases, injectable code controls authentication; this may lead to a remote vulnerability.

Gain Privileges or Assume Identity

Injected code can access resources that the attacker is directly prevented from accessing.

Execute Unauthorized Code or Commands

Code injection attacks can lead to loss of data integrity in nearly all cases as the control-plane data injected is always incidental to data recall or writing. Additionally, code injection can often result in the execution of arbitrary code.

Hide Activities

Often the actions performed by injected control code are unlogged.

1. **Vulnerability Name:** Hard-coded Access Key

**CWE:** CWE-798

**SANS Category:** A19:2021- Hard-coded Access Key

**Description:**

The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

**Business Impact:**

Bypass Protection Mechanism

If hard-coded passwords are used, it is almost certain that malicious users will gain access to the account in question.

Read Application Data; Gain Privileges or Assume Identity; Execute Unauthorized Code or Commands; Other

This weakness can lead to the exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or even execute arbitrary code.

1. **Vulnerability Name:** Uncontrolled Resource Consumption

**CWE:** CWE-400

**SANS Category:** A20:2021- Uncontrolled Resource Consumption

**Description:**

The product does not properly control the allocation and maintenance of a limited resource, thereby enabling an actor to influence the amount of resources consumed, eventually leading to the exhaustion of available resources.

**Business Impact:**

DoS: Crash, Exit, or Restart; DoS: Resource Consumption (CPU); DoS: Resource Consumption (Memory); DoS: Resource Consumption (Other)

The most common result of resource exhaustion is denial of service. The product may slow down, crash due to unhandled errors, or lock out legitimate users.

Bypass Protection Mechanism; Other

In some cases it may be possible to force the product to "fail open" in the event of resource exhaustion. The state of the product -- and possibly the security functionality - may then be compromised.