**Security Incident Reporting Form**

**Date of Report:** April 30, 2023

*Employee Details*

**Name**: Nahuel Hernan Ramos

*Incident Details*

**Date & Time of Incident (Initial Detection in Log File):** 8/24/20 11:23:47.000 AM

**Type of Incident:**

Malware / Data Breach / Phishing / Physical Security Breach / Other: UNION-based SQL Injection

**List of 5 systems, IP addresses or other components involved:**

**IP client:** 1.23.111.103

**IP host:** 10.0.1.126

**Uri:** Values Count %

/mutillidae/index.php?page=user-info.php&username=a%27+UNION+SELECT+1%2Cschema\_name%2C1%2C1%2C1%2C1%2C1++FROM+information\_schema.schemata+--+&password=&user-info-php-submit-button=View+Account+Details

**Username:** a%27+UNION+SELECT+1%2Cschema\_name%2C1%2C1%2C1%2C1%2C1++FROM+information\_schema.schemata+--+

**Source:** /opt/lampp/logs/access\_log

**Sourcetype:** access\_common

**Parameter in the URL:** username

**Status:** HTTP 200

**Page:** user-info.php

**File:** index.php

**How was the incident detected? (Text only)**

I used Splunk (IDS), and I reviewed the logs using filters to detect SQL injection as for example some logs had a parameter of username in the URL with words as SELECT, UNION and WHERE, it indicated a attack with UNION-based SQL injection.

**What are your immediate recommendations for handling this incident?**

-Once a person responsible for coordinating remediation is identified, the responsible will work with your CSIRT team and the areas of Networking and/or Infrastructure, Information Security Operations and Security Operations Center(SOC) to answer together at the incident.

- Update CSIRT members and relevant personnel. Do not revert any configuration made – these may be used as forensic evidence for later

-Coordinate an investigation of potentially vulnerable web pages and resources amongst developers or other stakeholders.

-A review of web, application, and database logs may reveal the point of vulnerability and source of attacks.

-Develop a plan to remediate any confirmed SQL Injection flaws and prevent future attacks.

**What would you recommend in order to detect this vulnerability/breach faster next time?**

-Implement server-side authentication and an intrusion detection system that detects abnormal client-side behavior.

-Use SQL controls across queries to avoid massive data exposure in the event of SQL injection.

-To minimize vulnerability to attacks that replace data with unexpected command execution, separate commands from data.

**What are your long-term recommendations for mitigating the vulnerability/breach?**

-Prepared statements: It ensure that attackers cannot change the intent of a query, even if SQL commands are inserted by an attacker.

-Stored procedures: It is a SQL statement that is precompiled and stored on the database server. It is convenient to create a stored procedure on a frequently used SQL script and save it on the database server. The next time that SQL script needs to be executed, you can just call that stored procedure.

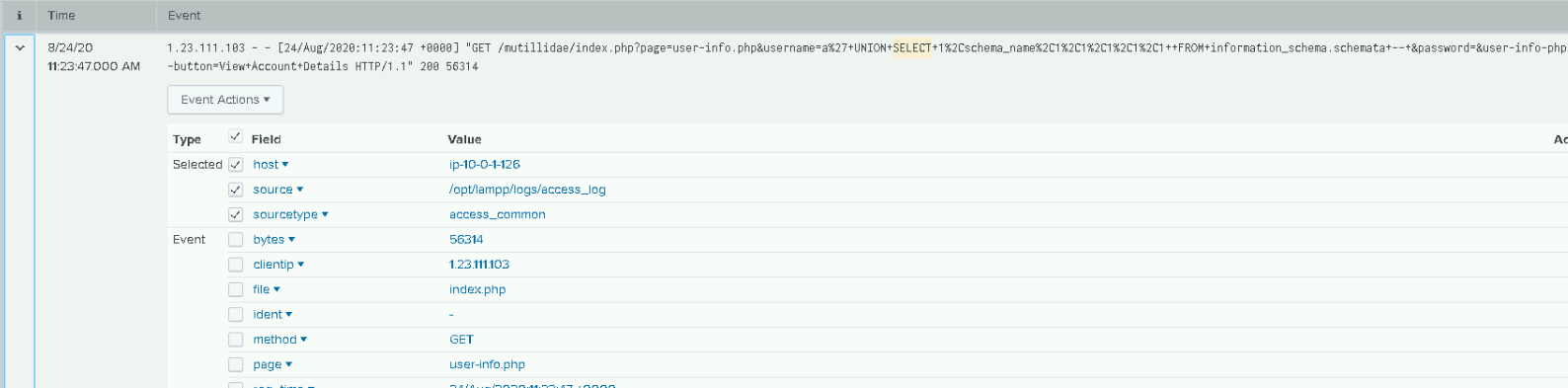
-Input validation and whitelists: Create a strict ruleset that allows inserting only relevant input in a specific parameter. For example, the parameter “ID number” should only allow numbers – no special characters or letters.

**Which corporate functions should be informed of this incident? (At least 3)**

* Networking and/or Infrastructure.
* Information Security Operations.
* Security Operations Center(SOC).
* CSIRT Team

**Add any supporting information here (log files, pictures, screenshots, videos, etc.)**

****

****

[**https://www.youtube.com/watch?v=PCTUKu4Ijig&ab\_channel=ASMO**](https://www.youtube.com/watch?v=PCTUKu4Ijig&ab_channel=ASMO)

[**https://foojay.io/today/8-best-practices-to-prevent-sql-injection-attacks/**](https://foojay.io/today/8-best-practices-to-prevent-sql-injection-attacks/)

