Secure Software Development (Computer Science) November 2021

Home / / My courses/ / SSDCS_PCOM7E November 2021 / / Unit 1 /

/ Collaborative Discussion 1: UML flowchart /

Collaborative Discussion 1: UML flowchart

This discussion will last for 3 weeks covering units 1, 2 and 3. Ensure you include appropriate citations and references in all your posts (this is not included in the word count for each week's post/submission).

Discussion Topic

Open source tools are available to create UML diagrams, some are listed below. This list is not exhaustive. The benefit of using such tools is that they ensure that the recognised UML components are used to represent the parts of the model correctly.

- **Visual Paradigm**
- **Sequence Diagram**
- **Umbrello**

Choose an open-source UML tool from the list above. Select one of the coding weaknesses which have been identified by OWASP and create a flowchart of the steps which may have led to the weakness occurring. Which UML models might you use to present the design of your proposed software, and why are they the most appropriate choice(s)?

Remember to save this to your e-portfolio.

Learning Outcomes

- Identify and manage security risks as part of a software development project.
- Critically analyse development problems and determine appropriate methodologies, tools and techniques (including program design and development) to solve them.

Assignment Guidance

- Your initial posting should respond to the question and be at least 200 words long. This should be labelled as '**Initial Post**'
- You will then respond to **at least 2 of your peers' posts** in unit 2 (each labelled as '**Peer Response**'). To guide your responses, look at the guidelines for the peer review process on the **Department's homepage**. Focus on the possible measures that could have been put in place in order to prevent the incidents highlighted by your peers. Please try to limit your response posts to **200-300 words maximum**, so that others may be encouraged to reflect on, and respond to your ideas.
- In Unit 3, you should provide a summary post based on your initial post, the feedback from your peers and the content of the three units. Please label this as '**Summary Post**'. It should be **300** words.
- Referencing: When you have referred to other authors thoughts, ideas and opinions in your posts, you must reference the author as you would in an academic assignment using the UoEO Harvard reference style.
- You will be not be assessed on your contribution to this forum throughout the module, but the tutor will post group feedback via a module announcement.
- This activity forms a component of your e-portfolio which you will submit in unit 12. All e-portfolio activities are intended to demonstrate your ability and strengths through evidence and reflection.

Secure Software Development (Computer Science) November 2021

Home / / My courses/ / SSDCS_PCOM7E November 2021 / / Unit 1 /

/ Collaborative Discussion 1: UML flowchart / / Initial Post /

« Collaborative Discussion 1: UML flowchart

Michael Botha

Initial Post

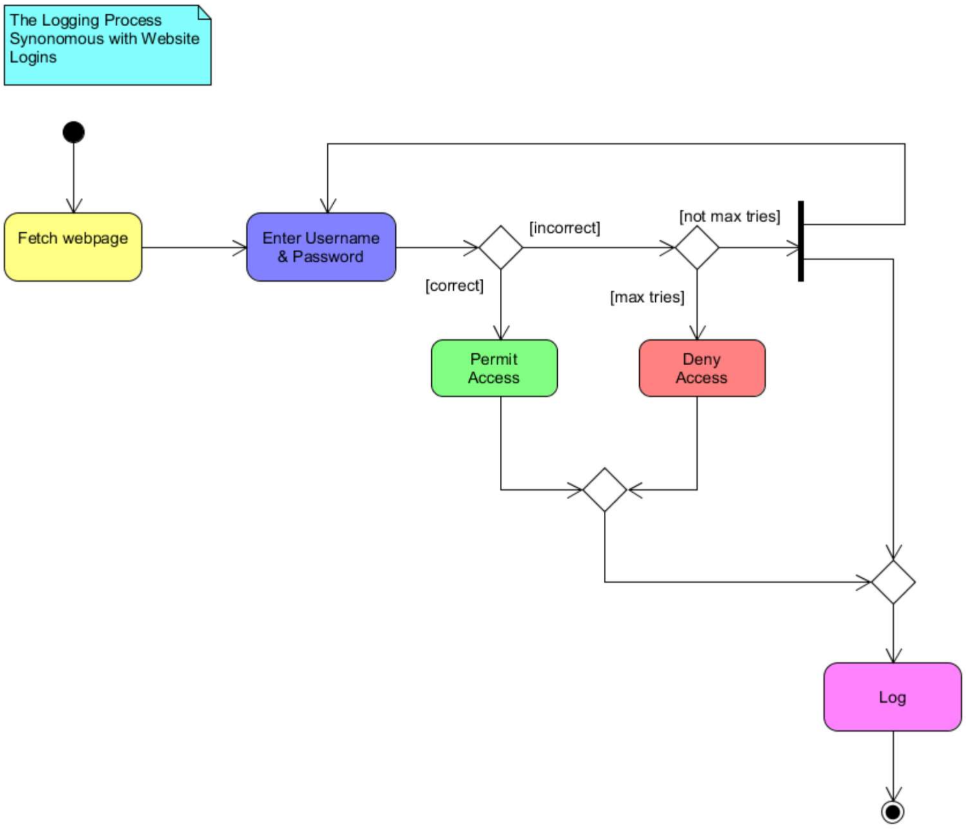
39 days ago

5 replies

Last 24 secs ago

Monitoring and logging is a critical aspect of many Information Technology (IT) systems, and is analogous to that of attaining and evaluating the pulse-rate of an ultra-marathon runner. For instance, in my current employment environment, Telecommunications equipment and networks are monitored 24-7 to ensure that critical services are maintained, and Service Level Agreements (SLAs) met. Within the domain of IT security, consider the importance of the monitoring, logging, and in some cases, response provided by Network Intrusion Detection and Network Intrusion Prevention Systems (NIPSs) (Fortinet, N.D). At the application level, it may be argued that this is where it is most important, as all the layers of the Transmission Control Protocol/ Internet Protocol (TCP/IP) model exist to facilitate the interaction between applications providing a service (Ciscopress, 2011).

According to the Open Web Application Security Project (OWASP) security logging and monitoring failures is one of the top ten web application security risks (OWASP, 2021). Please see the figure below, where the process of logging into a system is represented by a Unified Modelling Language (UML) activity diagram. Should a malicious user try various username and password combinations, perhaps attained through dumpster-diving, and the application not log these occurrences, or log them incorrectly, a future breach may be possible (OWASP, 2021). Additionally, if an application were to often run exception logic, this needs to be logged clearly, as a portion of the program may be incorrect (OWASP, 2021). There are many other use cases for security monitoring and logging for applications (OWASP, 2021).



If one were to present the design of proposed software via UML, various diagrams would be used, each emphasising particular aspects (University of Essex Online, 2021). The set of diagrams most apt for design representation is the Class, Object, Activity, and Sequence diagrams (University of Essex Online, 2021). The class and object diagrams will model the envisaged code structure and relationships, whereas the Sequence and Activity diagrams the domain specific processes, and code behaviour (University of Essex Online, 2021).

References:

Ciscopress. (2011) The TCP/IP and OSI Networking Model. Available from: **https://www.ciscopress.com/articles/article.asp?p=1757634&seqNum=2** [Accessed 16/11/2021].

Fortinet. (N.D) Intrusion Detection System. Available from: **https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system** [Accessed 16 November 2021].

OWASP. (2021) OWASP Top Ten. Available from: **https://owasp.org/www-project-top-ten/** [Accessed 16 November 2021].

University of Essex Online. (2021) *Introduction to Secure Software Development* [Lecturecast]. SSDCS_PCOM7E November 2021 Secure Software Development November 2021.

Reply

Maximum rating: -

5 replies

1 Post by **Cathryn Peoples**

38 days ago

Re: Initial Post

Thank you very much for posting this, Michael.

Taking the ideas further, what is important to log here? The username and password attempt details. Perhaps the location of where the login attempt is originating from. Possibly also the frequency at which login attempts are being made and login attempts from a new device. These details might be retained within a 'threat log', in recognition of the fact that a variety of logs may be maintained for different objectives (e.g. information, warning, error).

Best wishes,

Cathryn

Reply

2 Reply to **Cathryn Peoples** from **Michael Botha** ↑

4 hours ago

Re: Initial Post

Hi Cathryn,

Thank you for your comments. According to OWASP it is important that each event log fundamentally records the following (OWASP, N.D):

1. Related to "when":
   a.) The date and time of the event in international format.
   b.) The date and time the event was logged.

2. Related to "where":

   a.) The software that generated the event.

   b.) A form of address related to the device hosting the application. For instance, the port number and IPv4 address.

   c.) A Geolocation.

3. Related to "who":

   a. Addressing related to the source invoking the event. Such as: an IP address, cellular number, or cell tower ID.

b. The user's identity as per his/her stored authentication details.

4. Related to "what"
   a. The type of event.
   b. The level of severity.
   c. A form of description.

Regarding which events are logged is specific to each application, with one needing to bear in mind the consequences of over-logging or under-logging (OWASP, N.D). However, there are some essential events which OWASP advises to log (OWASP, N.D). Considering the suggestions you made, I would log the authentication failure, but perhaps leave the frequency at which login attempts are being made to some form of logging analyser. Thereby, decreasing the amount of processor time required of the underlying hardware, and decreasing logging "fog" (OWASP, N.D). Furthermore, with regards to a separate 'Threat log', it may be better to aggregate all alarms in a single log file and then filter it off premises or on another device so as to save system resources on the device running the application hardware.

References:

OWASP. (N.D) Logging Cheat Sheet. Available from: **https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html** [Accessed 25 December 2021]

Reply

Maximum rating: -

3

Post by **Lukman Mohamed**

25 days ago

Peer Response

Hi Micheal,

I enjoyed reading your initial post and I found your UML diagram and thought process very interesting.

As you have mentioned in your post, logging and monitoring go hand in hand and there is no point in having sufficient logs if they are not adequately monitored.

The dilemma of insufficient logging and monitoring covers the entire IT infrastructure. One of the biggest challenges facing organizations is that there are so many logs because almost all contemporary systems generate their logs. Therefore, log management becomes a major problem.

By the time that an organization collects the logs, the utter volume of the data set becomes too big to adequately monitor manually.

Some systems will make their logs and will alert the organization if it suspects something unusual – but again it needs the security team responsible to understand the alerts and prioritize the organization's response. (Wijaya)

Here's a problem in a nutshell, -which may seem counterintuitive- systems have to generate sufficient logs, and the organization's security team has to adequately monitor and interpret the information coming from those logs.

This problem is getting worse with the emergence of very complex, that are specially made to be stealthy and to not activate any alerts from installed monitoring and logging systems.

For example, fileless attacks -which are a type of malicious activity that uses native and legitimate tools built into a system to execute a cyber attack- will not leave any malicious files onto storage drives which means that there is nothing to be detected by always-on monitoring or anti-virus software. They utilize legitimate operating system software, like PowerShell, which will not trigger monitoring software watching for unusual behavior.

In your opinion, how can we overcome this problem?

References:

- OWASP. (2021) OWASP Top Ten. Available from: https://owasp.org/www-project-top-ten/ [Accessed 27th November 2021].
- “A09 Security Logging and Monitoring Failures - OWASP Top 10:2021.” *Owasp.org*, 2021, owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/. Accessed 27 Nov. 2021.
- Lyba, M.-V., and L. Uhryn. “MODERN TOOLS for SECURITY TESTING from OWASP.” *Bulletin of Lviv State University of Life Safety*, vol. 22, 28 Dec. 2020, pp. 18–22, 10.32447/20784643.22.2020.03. Accessed 21 Mar. 2021.
- Wijaya, Yansyah Saputra. “Web-Based Dashboard for Monitoring Penetration Testing Activities Based on OWASP Standards.” *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, vol. 6, no. 1, 26 July 2020, p. 36, 10.26555/jiteki.v16i1.17019. Accessed 27 Nov. 2020.

Reply

4 Reply to **Lukman Mohamed** from **Michael Botha** ↑

2 hours ago

Re: Peer Response

Hi Lukman,

Thank you for your response. This is a huge topic, which requires many hours of study to give you a holistic answer. However, I can comment and give a few suggestions below:

- Too many log messages may mean that the implementation of a system was not done correctly (Kent & Souppaya, 2006). Most equipment has an inordinate amount of events available to log, and requires specific selections of the most appropriate ones to record (Kent & Souppaya, 2006). Additionally, the aggregation of events is an essential part of logging (Kent & Souppaya, 2006).

- Various technologies allow the assignment of a severity to an event (OWASP, N.D). Should the incorrect severity be allocated to an event the relevance of such is obfuscated (OWASP, N.D).

- It is essential that monitoring technologies be deployed to perform automated analysis and flagging (Janakiram, 2017). For instance, AI and ML is an incredibly disruptive technology being deployed in industry and can be leveraged to facilitate intelligent monitoring and flagging (Janakiram, 2017).

- The best way to prevent a cyber-attack is not to rely solely on logging or monitoring, especially at the application level alone (McAfee, N.D). A defence in-depth approach is requried (McAfee, N.D). For instance, to be able to have sufficient permissions to run a PowerShell script, requires passing a few layers of security first. Therefore, although fileless attacks may be difficult to pick up once in memory, there should be a certain amount of defence before that level of penetration has been acquired (McAfee, N.D).

Reference:

Janakiram, M. (2017) Meet the Startups that Bring Artificial Intelligence to log Management and Analysis. Available from: **https://www.forbes.com/sites/janakirammsv/2017/07/04/meet-the-startups-that-bring-artificial-intelligence-to-log-management-and-analysis/?sh=17642d447d3f** [Accessed 25 December 2021].

Kent, K., Souppaya, M. (2006) Guide to Computer Security Log Management. Revision 1. United States of America: National Institute of Standards and Technology. Available from: **https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pd** [Accessed 25 December 2021].

McAfee. (N.D) What is Fileless Malware? Available from: **https://www.mcafee.com/enterprise/en-in/security-awareness/ransomware/what-is-fileless-malware.html** [Accessed 26 December 2021].

OWASP. (N.D) Logging Cheat Sheet. Available from: **https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html** [Accessed 25 December 2021]

Reply

Maximum rating: -

5

Post by **Michael Botha**

24 secs ago

Summary Post

In summary, the accurate logging of specific events related to an application, and how it behaves with regards to its interactions with other applications, systems, users, and itself is a critical aspect of software development (Kent & Souppaya, 2006). Furthermore, the monitoring of the generated logs is essential to maintaining a secure software system (Kent & Souppaya, 2006). Additionally, OWASP has flagged logging and monitoring as one aspect which is often incorrectly applied, and thereby produces vulnerabilities of varying sorts. For instance, below is a list of common logging mistakes (OWASP, 2021):

- Important events not being logged, preventing incident investigation.

- The generation of unclear log messages making analysis difficult.

- An application cannot detect and log in real-time, therefore hindering a fast response time to an incident.

To assist monitoring and analysis it is important to correctly select the events that should trigger logs (OWASP, N.D). This is often done during the configuration of certain technologies when implementing them. It is essential to have the right severity level appended to a particular log, and to have a standard and uniform logging format (OWASP, N.D). Generally, OWASP advises that the “when”, “where”, “who”, and “what” of any event be present in a log (OWASP, N.D).

Over-logging creates the problem of not easily being able to analyse situations, due to the large number of irrelevant logs, mudding the waters of the flow of events with regards to an incident (OWASP, N.D). Additionally, too many logs may stress underlying hardware infrastructure supporting an application, decreasing its performance (OWASP, N.D).

Due to the specialised nature of monitoring and log-analysis, certain technologies like AI may need to be leveraged to obtain real-time notifications and information (Janakiram, 2017).

Lastly, it is critical to not put all one’s security eggs into a single logging and monitoring basket (McAfee, N.D). Therefore, a security in-depth approach is essential to the overall hardening of any software system (McAfee, N.D).

Reference:

Janakiram, M. (2017) Meet the Startups that Bring Artificial Intelligence to log Management and Analysis. Available from: **https://www.forbes.com/sites/janakirammsv/2017/07/04/meet-the-startups-that-bring-artificial-intelligence-to-log-management-and-analysis/?sh=17642d447d3f** [Accessed 25 December 2021].

Kent, K., Souppaya, M. (2006) Guide to Computer Security Log Management. Revision 1. United States of America: National Institute of Standards and Technology. Available from: **https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pd** [Accessed 25 December 2021].

McAfee. (N.D) What is Fileless Malware? Available from: **https://www.mcafee.com/enterprise/en-in/security-awareness/ransomware/what-is-fileless-malware.html** [Accessed 26 December 2021].

OWASP. (N.D) Logging Cheat Sheet. Available from: **https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html** [Accessed 25 December 2021]

OWASP. (2021) OWASP Top Ten. Available from: **https://owasp.org/www-project-top-ten/** [Accessed 16 November 2021].

Reply Edit Delete

Maximum rating: -

Add your reply