Quantitative Analysis of Systems (QAS)

Academic Year 2020-2021

Anomaly-Based Error / Intrusion Detection

Scenario: Healthcare-related System

You are the owner of a server that hosts an Healthcare-related system. This system stores sensitive data of citizens and patients, and exposes some web-services to access those data remotely.

More in detail, this server consists of a single machine that usually executes Disk-intensive jobs, with high usages of RAM (to load pages of memory from hard drives) and network (to communicate results), despite having a relatively low usage of CPU (data does not really need to be elaborated, just embedded into webpages or JSON/XML files that web-services provide as outputs).

We assume it is possible to monitor some performance indicators of such server, installing specific software monitoring i) network usage, ii) disk usage, and iii) OS-specific metrics such as size of system buffers, system calls, page faults / cache misses.

Potential Faults and Threats to the System

As an administrator of this system, you are worried about possible failures of hard drives, availability of your platform or integrity losses of data, and also you want to preserve confidentiality as those healthcare-related data are private and should be disclosed only with the owner or authorized medical personnel.

To check the resilience of this system to potential integrity and confidentiality issues above, you want to setup a campaign where you simulate at least the following three faults/attacks:

- INTEGRITY: Erasing/Corrupting HD Memory. Integrity of data may be threatened by progressive or sudden corruption of some files stored in the hard drive, erasing, hiding (e.g., encrypting) or overwriting their content. Therefore, you want to write a software that simulates those activities, opens files, modifies, deletes or damages them in some way.
- AVAILABILITY: Denial of Service (DoS). DoS is a cyber-attack in which the attacker seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.

- CONFIDENTIALITY: Malware. A Malware is any software intentionally designed to cause damage to a computer, server or computer network. Malware does the damage after it is implanted or introduced in some way

into a target's computer and can take the form of executable code, scripts, active content, and other software. The code is described as computer viruses, worms, Trojan horses, ransomware, spyware, adware, and scareware, among other terms. In particular, the students have to simulate a malware that – once installed - reads sensitive data of the user and send it through the network.

Data gathered through monitoring activities both when the system is operating normally and when those three issues happen individually will be used to train ML-based binary classifiers that can detect those issues by processing monitored data. Both supervised and unsupervised classifiers should be used and discussed.

Project: Main Steps

To setup and perform the experiments above you will need to:

- Create a copy of the real system, simulating the server on your own laptop.
- Derive a monitoring strategy with probes that gather data related to the three issues above e.g., amount of r/w accesses to the disk, network-related data. This is critical as observing system indicators that are not directly related to those issues will negatively impact data analysis
- Conduct experiments with and without the three issues above, which have to be injected separately. The injection of attacks may be performed either using custom scripts or using existing tools such as *nmap* (https://en.wikipedia.org/wiki/Nmap).
- Analyse results to derive the best anomaly-based detection strategy for this particular system and fault/attack model. Data analysis should be carried out through ML strategies, both supervised (using WEKA or other tools) and unsupervised (using RELOAD or other tools). Metric scores achieved by those algorithms have to be discussed and compared.

Possible Project Extensions

(not mandatory, but recommended for groups)

1. **Simulating more issues** than the three above constitutes an additional merit.

2. It would be interesting to specifically explore **the impact of False Negatives** on this system, looking out for specific metrics (F2-Score, Recall …) that focus on FNs rather than FPs. False Negatives represent items where an issue is happening in the system and the detector fails to identify it, and therefore are of great interest when dealing with critical systems.

Project Submission

The project can be developed **individually** or by a group of students (**3 students max per group**).
The result of the project consists of a written report and the implemented models.

When submitting the project, you have to produce:

- the main document, a PDF file in which you discuss how did you approach to each of the steps above, reporting at least:
    - Name, surname and serial number of the students.
    - Text of the project.
    - The **main problems** addressed.
    - The **methodology** applied for the investigation and analysis of the problem, and any information on the results of the various phases of this methodology
    - The description of any developed software architecture, and **any details about the software**.
    - The detailed **description and discussion of the results**, with any information on the quality of the measurements and intrusiveness,
    - along with a final discussion on metric scores of algorithms, also expanding on the detection of specific issues
    - Appendix with possible instructions for installation and re-execution of the project.
- Additional code developed for the project (e.g., scripts for injecting attacks), or details about the usage of third-party tools for monitoring and attack injection.
- Configuration files of WEKA/RELOAD or details about the usage of other tools for data analysis.

A single tar.gz/zip format file named “QAS-assignment-xyz.tar.gz” or “…zip”, where x, y, and z are the initials of name and surname of each of the members of the group, should be sent to lollini@unifi.it, tommaso.zoppi@unifi.it containing only TWO files:

- a PDF file for the report;
- an archive containing the developed software, files, configurations.