Unomaly

How Unomaly works

How Unomaly turns raw, vast volumes of software-generated data into visibility and ground truth?

Data is everywhere because software is everywhere

IT systems aren't metal. In fact, the only "must have" part of an IT system is something completely different and non-physical: software.

IT systems and services are composed of software and lines of code. All software generates data as part of what they do. In fact, it is commonly known that machine and software generated data is the single greatest source of data today - by volume. Software and systems simply outrun us humans in terms of activities per day or operations per second. It is literally impossible for us to tweet, post, capture images in the same pace.

This data, software and machine generated data, is also commonly referred to as log or event data. It captures any action and event that software takes based on current circumstances. So as software runs, it speaks. Almost from a first person perspective like

"I'm processing a transaction", "I'm receiving an incoming login request from X", "I couldn't finish doing that", even "I have no idea what just happened". As all those activities and events are captured in the data it offers us an immensely valuable resource for understanding our environments. It is basically all we need, so the question is why aren't we making greater use of it?

" It is basically all we need – so the question is why aren't we making greater use of it? "

The challenge is of course that this data is produced in the systems themselves where it can't typically be accessed, and it amounts to millions and trillions of lines per day, and there is really no natural way of determining which data is relevant. It all comes down to "of all the things software is saying, what should I be aware of?" Since we can't know - we effectively need to analyse everything.

Do you know that a car has over 150k lines of code? And that it processes more than 10k lines of code every second? And that every 100:th line of code issues an event that is logged? That results in 10 events per second.

Applied machine learning that keeps track of data

The majority of all data that running software produces is just the same - day in and day out. When our environments run as "business as usual" everything is on repeat, and so is the data. In fact, we can show that over 99.99999% of all data produced by the software of an average environment are just repetitions.

Unomaly figures this out by continuously analysing all data that the environment and its systems are producing - it literally learns, with applied machine learning algorithms - the repetitions and normal data that each system produces. As data is received, Unomaly looks at it, compares it to history and either increment statistics or sees it as an anomaly. It even draws conclusions and learns dynamic parameters from parts of the data, like timestamps, sequence numbers etc.

This normal data may be interesting - and Unomaly offers the ability to look at the aggregated learning per system. This summary data, or "system profile" which we call it, means that organisations can inventory their environment, look for bad things that are part

of “business as usual” all while just looking at just a dense picture of everything. Not all events line by line day in and day out. Just one line, with metadata.

The same applies to how Unomaly technically treats data. Unomaly doesn’t store all that repeating data for every event, but persists a complete, ever-historic summary. It makes it a lot easier to work with data in this form.

> Do you know that an environment producing 100GB of log data per day where 99.999% is repeating only grows with 1 MB per day? It’s a huge difference managing 1 MB per day vs 100GB.

Anomaly detection that uncovers relevant data

Most of the risks that are facing our technical environments in terms of availability, security and resilience will be an anomaly when they actually happen. The devastating ones will definitely be unknown, and something we can’t possibly prepare for. Even a previously known incident that is likely to happen more frequently, will most likely have ended up there for unknown reasons as well. So every incident has a degree of uncertainty and unknown part of it.

This is what the grand challenge is: we simply don’t know how incidents will look like, so we can’t actively search for them, build rules, or be on the lookout for them. We end up waiting for impact to find them and manually work our way backwards to understand why. Rarely are any real answers found during that process due to natural time constraints.

Since every incident is “different from normal”, they make software behave differently. As an incident progresses, it changes more and more software and code paths, and as such making the software produce new and different data, since it does things out of the ordinary. This new and changing data may offer a complete audit trail of how the incident has evolved and an opportunity to follow it, detect it, investigate it and explain it.

Unomaly enables this in essence by keeping track of all things normal and comparing all data to that, highlighting the anomalies.

An anomaly may be different things, but is essentially the difference or indifference

between the particular data being processed and all previous data. It may be as subtle as an individual parameter that have changed (like usernames are always: Carl, but now all of the sudden is root) or a completely new event on a certain system (like: an error log) or even a completely new event in the entire population of systems (like: a nasty segmentation fault). Since Unomaly understands the type of anomaly, it can apply a score that raises the appropriate amount of awareness of the situation. By the way, a situation is how Unomaly presents anomalies - as a clustered set of time related anomalous events.

Do you know that an average environment consisting of 100 servers generate just 3-7 anomalous situations per day – those are high risk changes, manipulations and problems.

Simple, visual and interactive interaction with data

How good is a system that hides important things by being too complex and advanced to use? Not much, especially since most incident and problems are solved in collaboration with others. A security expert needs help from operations to investigate an anomaly, just as well as an operations engineer needs the help of a developer to investigate an application issue. An incident may span all parts of the environment and the entire organisation - so it has to be simple, universal and easy to understand. Solutions with a steep learning curve or unintuitive user interface, will make engineers that may have the greatest need take a step back and use the traditional means of investigating issues.

Unomaly is real time, intuitive and visual. The algorithm has an average processing time for an event at 200 ms - with transmission time, database synchronisation etc, it typically means that an event and its analysis results is visible in the UI and sent as an alert in 10 seconds or less.

Just send data

Getting started is equally important. How good is it to have a capable solution that never lands in production because it requires too much setup?

With Unomaly, you just configure systems to send their streaming data to it. Different systems and software have different support for sending their data. Every Unomaly

instance has a very flexible set of built in receivers for standard data such as Syslog and SNMP, but also for plain TCP and UDP-based streaming data, custom transports for Hadoop, Syslog servers and many commercial log management solutions that doesn't practice automation. Integrating Unomaly is often completed in less that 2 hours.

Sending data is of course just the start. The key with Unomaly is that it doesn't use any parsing, field extractions or similar, which means it just receives data verbatim - just the way your systems and software have chosen to produce it. All data gets treated and analysed in the exact same way. There are simply no "packs", "apps" or similar, but instead a universal way of analysing any data. This is the major reason for why Unomaly is plug and play not just from an "integration" perspective but also from a value perspective.

All in all - you can get started quickly.

Conclusion

Organisations face a complex situation: the need for increased pace of change and openness of IT, all while increasing availability and resilience from incidents of all natures and types. As preventive controls and protection doesn't help, awareness and responsiveness are the only way forward.

Unomaly offers a simple, plug and play and automated solution for creating visibility into the environment. By algorithmically analysing all the data that the environment produces while it runs, it provides the raw and true picture of what is really normal and highlights whenever something changes. And by doing that, it universally clarifies technical change-related risks - which uncovers anything from crashes, to changes and intruders.