THE PROBLEM

A noisy night, and a decision nobody wrote down.

Scattered evidence, an anxious judgement call, and no record of who decided what.

Ten past two in the morning. The log-in records for one of the company's public servers start filling with failures: hundreds a minute, from dozens of addresses. At the same moment, a computer inside the company starts contacting places it has never contacted before.

The person on call has three questions to answer before morning. Is this a real attack? How sure are we? What should we do? Answering them usually means checking one tool for the server records, another for network activity, and a chat thread where someone eventually types "go ahead".

By the next day, nobody can say what the decision was based on, who made it, or whether it could have been undone. The number the analyst quoted was never written down. The odd network activity was either ignored or treated as proof. The approval was a chat message.

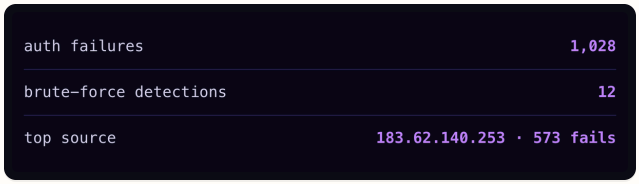
What the analyst needs is a number they can stand behind, a clear note of what is still unknown, and a proposed response in front of a named person. That is what SOAR builds, and the next three pages follow the night through it.

WHAT SOAR SAW

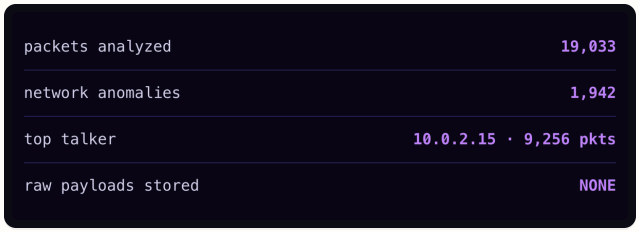
The night, as SOAR shows it.

The failed log-ins, the network activity, and one view that separates what is proven from what is not.

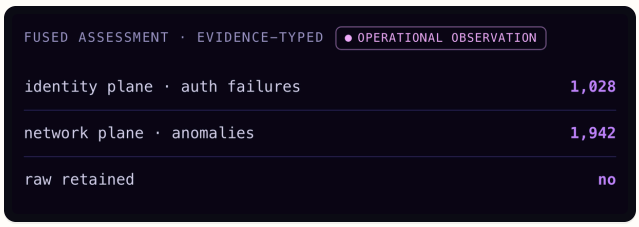
We used two of SOAR's built-in demonstrations, each a real, publicly available recording: a server's log of password-guessing, and a separate recording of network traffic. They are two separate runs, not one night; this is what each showed, and how SOAR puts such findings side by side.



SOAR reads the server's records of log-in attempts as they arrive. In the recording it counted 1,028 failed attempts from 24 addresses, and recognised the pattern of someone guessing passwords. It labels the count **measured**, because it came straight from the records rather than from an estimate.



On the network it looks only at the outside of each message: who sent it, to whom, and how big it was. Of 19,033 messages it flagged 1,942 as unusual, and about half of all the traffic came from one internal computer. It never stores what the messages said.



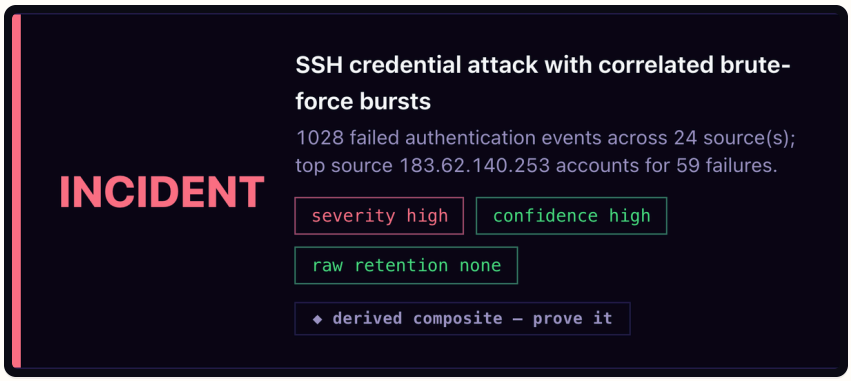
Then it puts the two together and says plainly which part is proven. The password guessing is an attack. The unusual network activity is only a lead: SOAR noted that it needs more checks, such as who owns that computer and which program was talking, before it can be called an incident.

By now the analyst has a number they can stand behind, and a lead that says what is still unknown. What is missing is a decision.

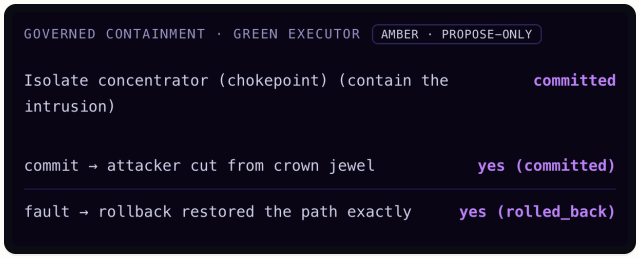
WHAT SOAR PROPOSED

A rehearsed response, waiting on a name.

How a finding becomes a rehearsed response, and who has to approve it.



The case summary. The “high” ratings come from the password-guessing count; the network lead is kept separate.



A practice run on SOAR’s built-in sample network, so the names on screen differ from this attack.

SOAR drafted a response, using AI models that run on your own computers: cut the attacker’s computer off from the rest of the network. Before anyone approves, SOAR rehearses that kind of change on a sample network, a simulation rather than your real one, and the attacker is cut off. Then it deliberately makes something fail so the undo can be seen, and the sample network goes back exactly as it was. That rehearsal is not a test of your live network, and the screen marks the response as a proposal only.

Every suggested action must say what it will change and how to undo it, or SOAR refuses it. Every case has a named owner.

Who signs depends on how risky you decide the action is. In the demonstration, one named person approved the step, and the approval ran out six hours later. At your site, you set the rules. If you decide that cutting off a server is high risk, two independent people must approve, and SOAR refuses two approvers it considers too closely linked. Anything that could affect safety needs the case owner and a process engineer. Steps your policy already allows go ahead without a person and are recorded.

Each approval a person gives records who gave it, when it runs out, and what evidence they saw. If the evidence changes before anything happens, the approval no longer counts. Once approved, your own team makes the change, using its usual process.

RUNNING IT AT YOUR SITE

The same night, with SOAR in place.

What changes for your team, what SOAR needs, and what stays with you.

Picture that night again, with SOAR in place. The analyst opens one screen instead of three. The password-guessing count is there, labelled measured. The odd network activity is there too, labelled as a lead, with a note of what would confirm it. A drafted response waits in front of the person your rules name, and when they approve, the approval records who they are, when it runs out and what they saw. The next morning, anyone can read what was decided and why.

To get there, SOAR needs two things your team already collects: the log-in records from your servers and a copy of your network activity. If you choose, it can also read a limited view from two widely used security products, SentinelOne and Splunk, without changing anything in them. Splunk searches return up to 50 rows, so analysts will still open those tools for a full search.

SOAR keeps counts, findings, cases and the record of who approved what. It does not keep your raw records or the contents of network traffic, and it cannot search old logs, so it works beside the system your team uses to store logs rather than replacing it.

When an analyst asks it a question, for example whether a known software flaw affects the server under attack, it looks the answer up in records kept on your own computer, including public lists of known flaws and what it has already seen on your network, shows which records it used, and labels the answer “Simulated”, meaning a model wrote it rather than measured it. If an attacker hides instructions inside the data, SOAR treats them as text, not commands, and nothing can be switched on or off by chatting with it.

It can run on a computer at your site with no internet connection; Google sign-in and the optional SentinelOne and Splunk links need one, so choose which way you run it. QASO’s three products have been tested together, cut off from the internet, on a rented computer in a data centre; they have not yet been tested on physical hardware. People sign in with a local user name and password, or with Google for addresses you allow; there are no separate roles yet, and it does not connect to your company’s own sign-in system. Your team still makes every change to live systems, and SOAR cannot read the specialist languages industrial equipment uses, so keep industrial equipment out of a first trial.