VERIS Coding Style Guide

Version 1.2

The VERIS Team

Verizon

Table of Contents

VERIS Resources

VERIS Overview

The Vocabulary for Event Recording and Incident Sharing (VERIS) was created to allow security practitioners to record the details of security incidents and be able to share that data with others in a standardized, repeatable, anonymous fashion. This framework underpins the Verizon Data Breach Investigations Report, and is also used to code cases for the VERIS Community Database Project (VCDB), where we release publicly disclosed data breach cases coded in VERIS format for public use.

If you're just getting started with VERIS, you've come to the right place. Here you will find the schema, with the detailed enumerated lists and the structure of the 4A's that are the center of VERIS: Actors, Actions, Assets and Attributes. Ideally, we would know enough about the incident that we can identify the values for each of the 4A's in every case. However, we know that in incident response, there are always unknowns. VERIS can handle them without causing a fuss. We can code as much or as little detail as we have for the case before us.

Getting started with VERIS can seem rather daunting—it is a complex framework that has evolved over time as we encounter edge cases that challenge the schema. However, it doesn't have to be a complicated ordeal to code up your first case in VERIS format. For the purposes of all of our training materials, we are using publicly disclosed cases taken from the VCDB data. For information on where you can see that data and the cases waiting to be coded, they are stored in our GitHub repo: https://github.com/vz-risk/VCDB. As we encounter new breach reports, we create an issue for them.

We also have a web application to facilitate coding your cases. The webapp is

available at http://veriscommunity.net/veris_webapp_min.html?#/. It stores all data locally, and the output of the case, once coded, is in JSON format. If you open the webapp, the first fields you are presented with deal with the victim demographics, and then it dives into the 4A's. We will discuss the demographics more in their own section.

What are these 4A's, and how do we use them? They are the Actors, Actions, Assets and Attributes involved in an incident or breach.

Actors

Actors are the people who are causing the breach by the actions they take. VERIS is an actor-centric framework, so identifying who the actor is (external to the victim organization, internal to the victim org, or a partner of the org) is the starting point. Some people find identifying the actor a bit confusing. We see cases where both internal and external actors are recorded, because the logic is that "the attack would not have succeeded if the employee hadn't clicked on the link in a phishing lure." That is not how VERIS is intended to work. It is only intended to record direct actions taken to cause the breach—so the external actor is sending the phishing, but the internal actor is the victim, not an active participant in the breach. For there to be multiple actors (and we do see this) they have to both be working together and attempting to cause the breach. There has to be malice involved for collusion. If every failure to implement a control that would stop a breach were counted as an error, it would obscure the actionable causes of breaches and render the statistics unusable. While it can be argued that there is always an underlying error in breach causes, it doesn't drive decisions on where to put controls if the answer is always "put in every possible control to cover every possible place an error can be made." The organization would be funneling all their resources to security and fail at their core purpose.

Actions

Once the actor has been determined, you turn your attention to the **Actions** that were taken during the incident. Was it a malicious action, like hacking or social engineering? Was it an accidental action like the loss of a laptop? Intent and motive play a key role in how these are categorized. For errors, the intent is not malicious. People do not leave their laptop behind on an airplane with malicious intent, typically. People do hack into database servers with malicious intent, however.

Assets

What **Assets** were involved in the incident? (And notice we say incident not breach—because you can record those valuable near misses using VERIS as well.) Was it paper documents? A web server? If it was a social attack, it involved what we call a "person asset", so you can record who is the target in these attacks if you know that information. While we use VERIS for our reporting and research, there are many organizations out there who use it as part of their security metrics infrastructure. Knowing who is being targeted in your organization will give you direction in how to tailor your awareness training.

Attributes

Finally, what the effect on the security **Attributes** as a result of the actions taken? We're talking here about the CIA Triad of Confidentiality, Integrity and Availability. Many cases involve multiples of the CIA triad. In VERIS, compromised security attributes are based on the expanded CIA Traid, which includes Confidentiality / Possession, Integrity / Authenticity and Availability / Utility. Multiple attributes can be affected for any one asset, and each attribute contains different metrics.

Think, for example, of an attack that started out as a phishing lure sent to one of your employees. They fall for it and provide their credentials in a form. The attacker uses those credentials against a web application infrastructure, and gains access. They then download malware, take a copy of any interesting data, and trigger the ransomware to encrypt all the files. So you have a breach of the confidentiality attribute with the data theft, but even before that, with the credential theft from the phishing attack. You also have integrity violations with the software installation of the malware, but also because the social attack caused the employee to alter their behavior and provide their credentials to the attacker's form. Finally, you have an availability violation when the data becomes obscured by the malware. You can see how one case can have multiple impacts to the CIA triad. We will get more in depth with a case that is similar to this down the road, including how the 4A's would come into play here. When learning a complex framework like VERIS, we find that the best approach is to use case examples to show how each kind of commonly encountered case would be handled. To that end, let's move into the Case Coding Examples section and dive into the data breaches.

About the VERIS Coding Style Guide

This guide is designed to both document the standard ways that cases are to be coded for the VERIS Community Database Project (VCDB), as well as other uses when coding cases in VERIS format. It is a reference for those just getting started with VERIS, and also for those who have been coding these cases for a long time, but may have questions how a specific kind of case should be handled. It includes the most relevant fields for each case type, but not all of the fields in VERIS are discussed.

This document will be updated annually after the VERIS wars have concluded, and the update to VERIS has been completed. (The VERIS wars are the DBIR team's annual version of Thunderdome—but we don't have the same

props, sadly. Actually, it is when we meet as a team to talk about the proposed changes to VERIS we've gathered over the past season, and decide what stays and what does not. VERIS is an evolving framework, and as the threat landscape changes, so must the way we are able to record it.) The guide versioning will be changed as it is updated, and the version of VERIS it covers will also be noted in the title.

The main documentation site for VERIS is the veriscommunity.net website. There is also a GitHub repository for VERIS, as well as one for VCDB. Here are the relevant links:

- VERIS Community - http://veriscommunity.net/
- VERIS GitHub Repo - https://github.com/vz-risk/veris
- VERIS WebApp - http://veriscommunity.net/veris_webapp_min.html#/submit/vcdb/1.3.5
- VCDB GitHub Repo - https://github.com/vz-risk/VCDB

Much of what is found on this guide will be duplicated on the VERIS Community site. We do encourage you to dig into the repos and get to know both VERIS as a framework, and VCDB as a dataset that can complement the data you collect.

Videos from Rapid7 about VERIS

If you prefer to learn by watching videos (as compared to reading this wonderful Guide), Rapid7 made some very nice training videos about the VERIS 4As. Here are the links to the videos, and if they do help you, don't forget to smash those Like and Subscribe buttons.

https://www.youtube.com/watch?v=k9OFoSdyLsg VERIS Actions
https://www.youtube.com/watch?v=STJ87WszpLY&t=3s%20Threat%20Actors VERIS Actors

There used to be videos on Assets and Attributes, but they seem to have been

removed.

Rapid7's video reference is another take on the topics of getting to know VERIS, and can be quite useful as an introduction.

Training videos for Coding Cases using the VERIS WebApp

We have a number of training videos developed over time that illustrate how to use the VERIS Webapp, and how to code some common case types. These are also good resources, although the webapp may look slightly different, as it is updated over time.

Here are two videos specific to using the WebApp to code VCDB cases:

From Suzanne: https://www.youtube.com/watch?v=YdwnfDfT1FU&t=208s Using the VERIS WebApp
From Gabe: https://www.youtube.com/watch?v=APNNCILB4aI VERIS WebApp Tutorial

Here are the case coding videos that cover some of the most common incidents we see:

Laptop Theft: https://www.youtube.com/watch?v=xphDFtTLzWk
POS Skimmer: https://www.youtube.com/watch?v=Aj-prZOheA0
Error Misdelivery: https://www.youtube.com/watch?v=g0FN2Zqe9FE
Misuse: https://www.youtube.com/watch?v=SI8eRitqRJE
Hacking: https://www.youtube.com/watch?v=qrt6a-uVD9I
Malware: https://www.youtube.com/watch?v=j8E7ir6rGQc

Here are the live stream Twitch case coding sessions with Gabriel Bassett:

https://www.youtube.com/channel/UCTGViRfVZwozbJtLxR05SKQ/videos

There are lots of different kinds of cases in the list, and you also get input from people watching the Twitch session.

Coding Decision Trees

Overview of the Trees

VERIS, at first glance, is a complex framework. It can seem daunting to people trying to implement it for the first time to figure out what the right answer is for "who is the Actor?" or "What Action(s) were involved in the incident?". To (hopefully) help with that, we have developed a series of decision trees for people to use to help decide the right choice for each case. Do you have to use them every time you code a case? Of course not. But when you are just getting started with VERIS and case coding, it can help to follow the directed decisions path.

In VERIS, the Actors, Actions, Assets and Attributes are not exclusive—that is cases can have multiple selections from each of those categories, rather than just one. In real cases, there is nothing that stops Actors from colluding with each other, for example, and we want the cases that are coded to reflect reality. There is also nothing stopping the determined attacker from using multiple actions to achieve their goals. They may start with phishing, and once they have successfully obtained credentials from their chosen victim, pivot to hacking into a web application using those stolen credentials. They may then drop malware to maintain persistence in the environment and disguise their actions from detection. The point here is that since there is nothing stopping the adversaries from using multiple actions, attacking multiple assets, and violating the confidentiality, integrity and availability all in one attack, the coding framework should not have those constraints either.

When using our coding decision trees, you may need to go through the paths multiple times to account for all of the details of the case. In the above scenario, you would traverse the action decision tree several times to account for the social

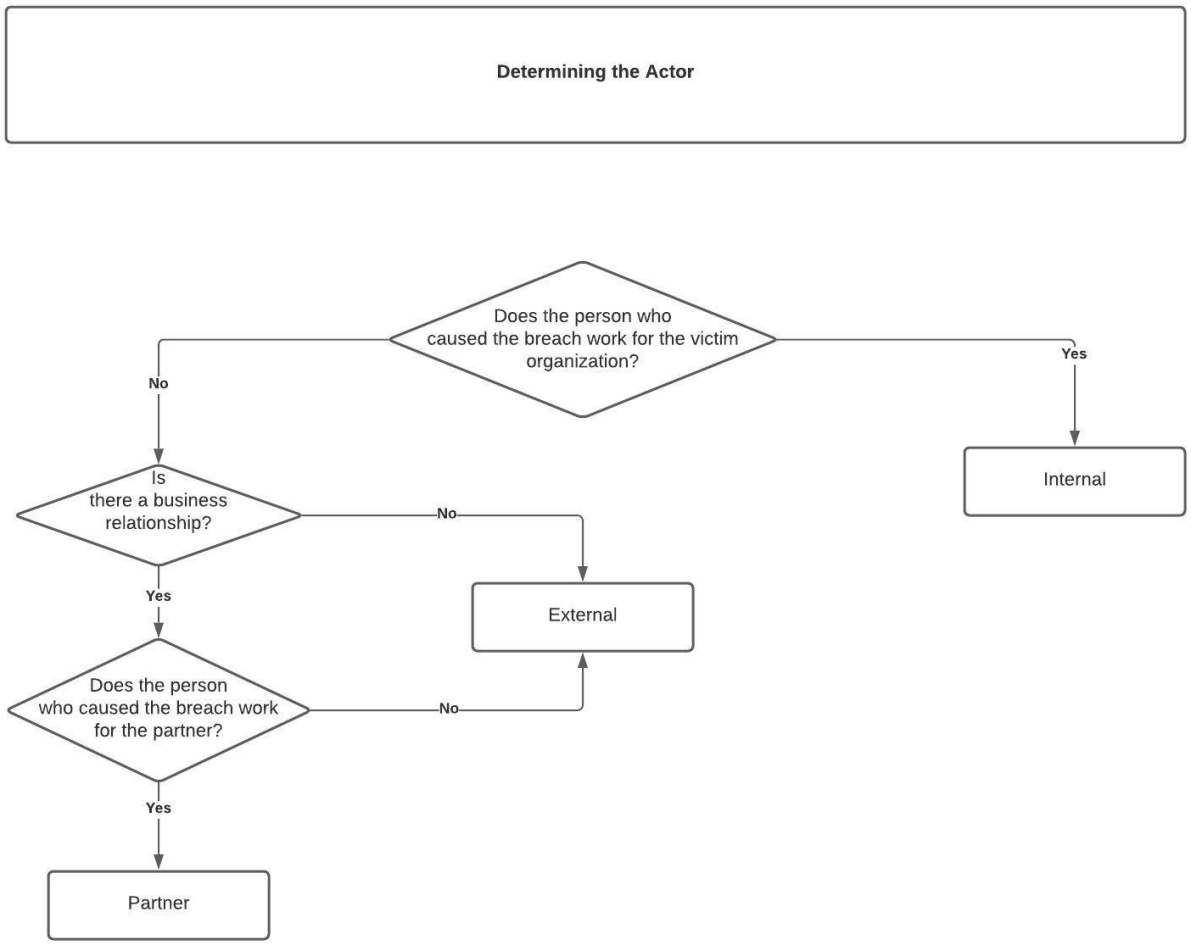
attack, the hacking action and the malware action. You'd also need to go through the asset tree several times to account for the Person who was socially engineered (yes, we count people as assets in VERIS), the User Device they were using (think laptop) when they were compromised, the web application the actor hacked into and installed malware on, and then move onto the Attribute tree to account for those.

Actor Coding Tree

Determining who the actor was in an incident is not always as straightforward it might appear. This is particularly true if you have someone causing a breach who works for a company that has a business relationship with the victim organization.

To help with that, we have created a decision tree to guide you in determining the correct actor. Our first decision is whether the actor who caused the breach works for the victim organization that experienced the breach. If they do, then you're done—it is an internal actor.

If the answer is they do not, then you need to determine if there is a business relationship involved between the perpetrator of the breach and the victim organization (at this point, we know they are not the same entity). These are sometimes referred to as partner or supply chain breaches, because these organizations tend to be engaged as part of the normal processes of doing business. Companies tend to outsource functions that are not their core competencies, and thus the business to business commerce flourishes.



So the question of "Is there a business relationship" needs to be answered. If there is such a relationship, then you must determine if the person who caused the breach works for that company that is providing some kind of product or service to the victim organization. If the actor works for the partner, then you have a partner actor breach. If the perpetrator does not work for that partner, then they are an external actor to the company that they attacked. You then have an external actor breach, not a partner breach.

VERIS is an actor-centric framework, rather than a data location framework. By this, we mean that it is the person performing the actions and what their relation is to the victim org(s) that drives what actor type (and breach type to a certain extent) you are looking at.

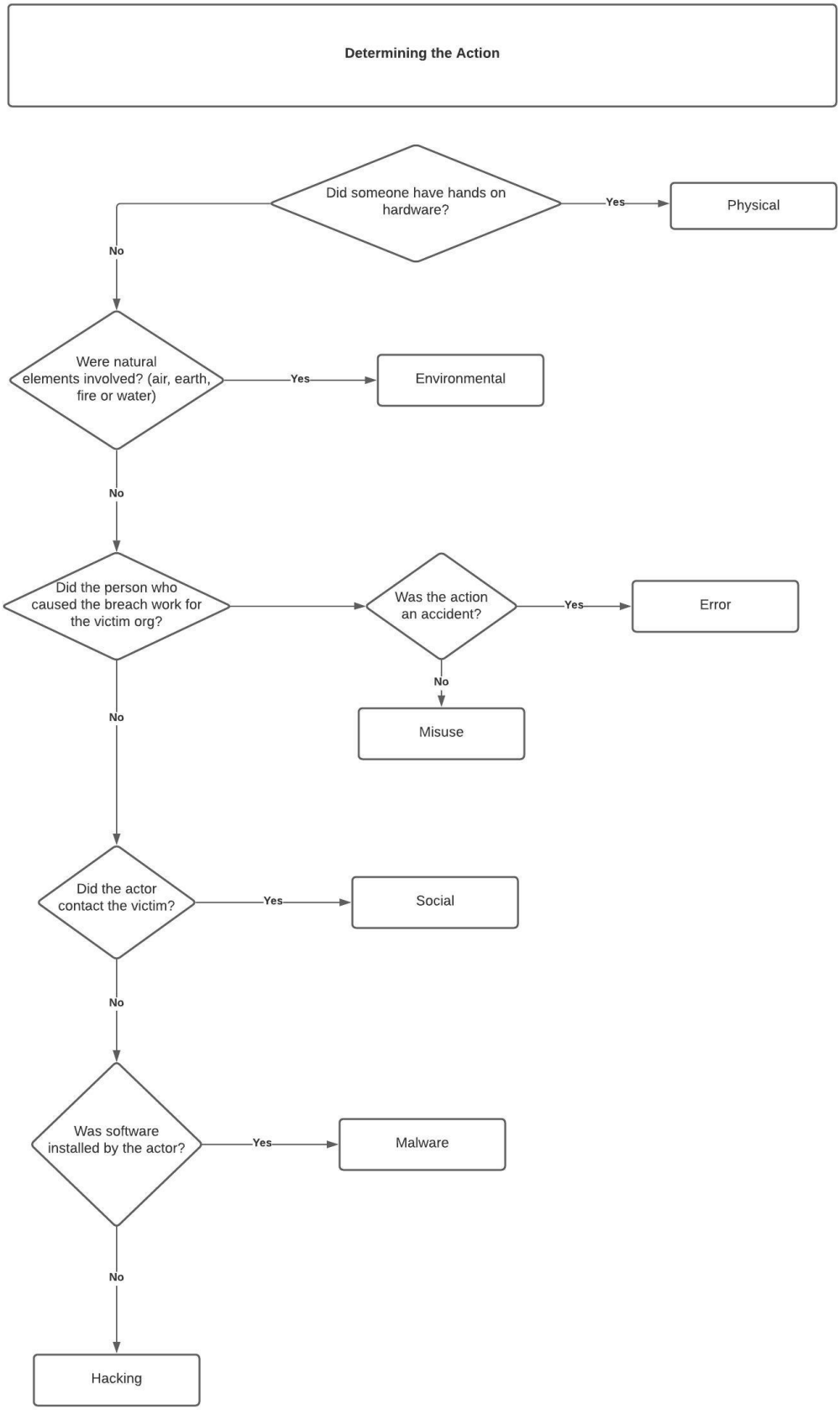
In other taxonomies, you may see that there are lots of partner breaches because they are determining that based on who had custody of the data that was compromised. We account for that information in our asset list, rather than in the actor determination.

One question that comes up in determining the actor at time is if the person who caused the breach is listed as a "contractor". How this is handled depends on the relationship bretween the victim organization and the person causing the breach. If they are an individual contractor, like someone hired to do a job directly, typically someone who is on premises at the company and is treated as one of the team, we would code them as an internal actor. If the term "contractor" is used to describe a business where there is an agreement between the two companies to provide a service, and it is negotiated at the business to business level, we would consider it to be a partner.

Action Coding Tree

There are many actions in VERIS, so making sure you understand and assign the correct action(s) to a case is important. Yes, these are non-exclusive and one incident may have multiple actions (and actors, etc.). It then becomes a matter of which action came first, and placing subsequent actions in proper order. This helps later when you are creating the event chain as well.

So first we will show an overall decision tree for determining the action. You may have to go through these decision trees multiple times if there are multiple actions, but you should be able to figure out which action you should be recording based on some key decisions about what happened in the case.

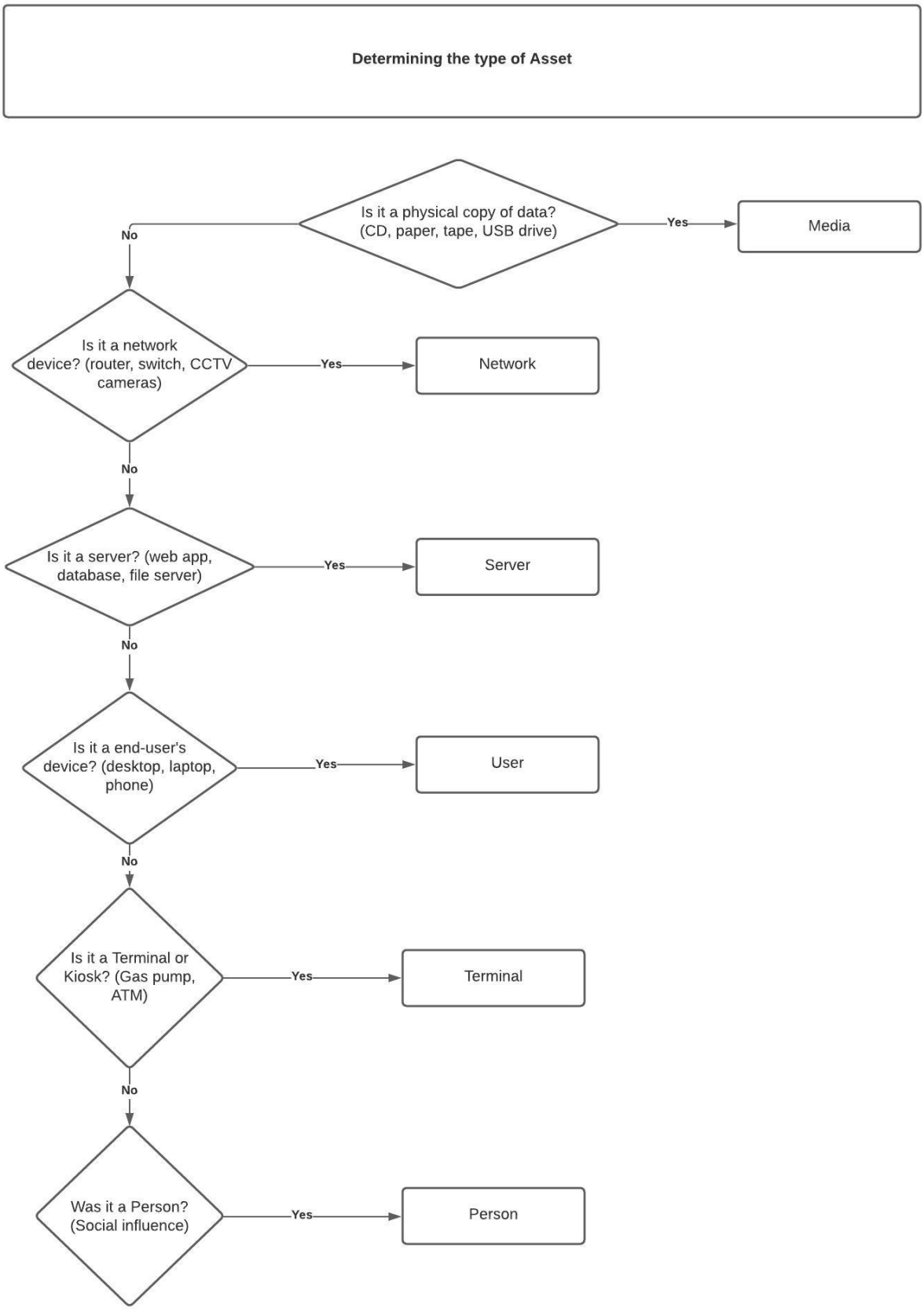

Determining the Action
Did someone have hands on hardware?
Yes
Physical
No
Were natural elements involved? (air, earth, fire or water)
Yes
Environmental
No
Did the person who caused the breach work for the victim org?
Was the action an accident?
Yes
Error
No
Misuse
No
Did the actor contact the victim?
Yes
Social
No
Was software installed by the actor?
Yes
Malware
No
Hacking

Each of the decisions end in one of the top-level Actions in VERIS. Once you have determined which top-level action you are dealing with (and there may be multiples), then you need to go on to determine the variety within that action as well.

Asset Coding Tree

Determining the Asset(s) affected in an incident seems pretty straightforward, but it can be trickier than it looks. Again, with the others of the 4As, we can have multiple selections. Certainly, it is not uncommon to find more than one asset involved in a case—particularly when you realize how common certain attacks are. Social attacks account for a high percentage of successful data breaches, and as such, you are at the very least looking at a Person asset.

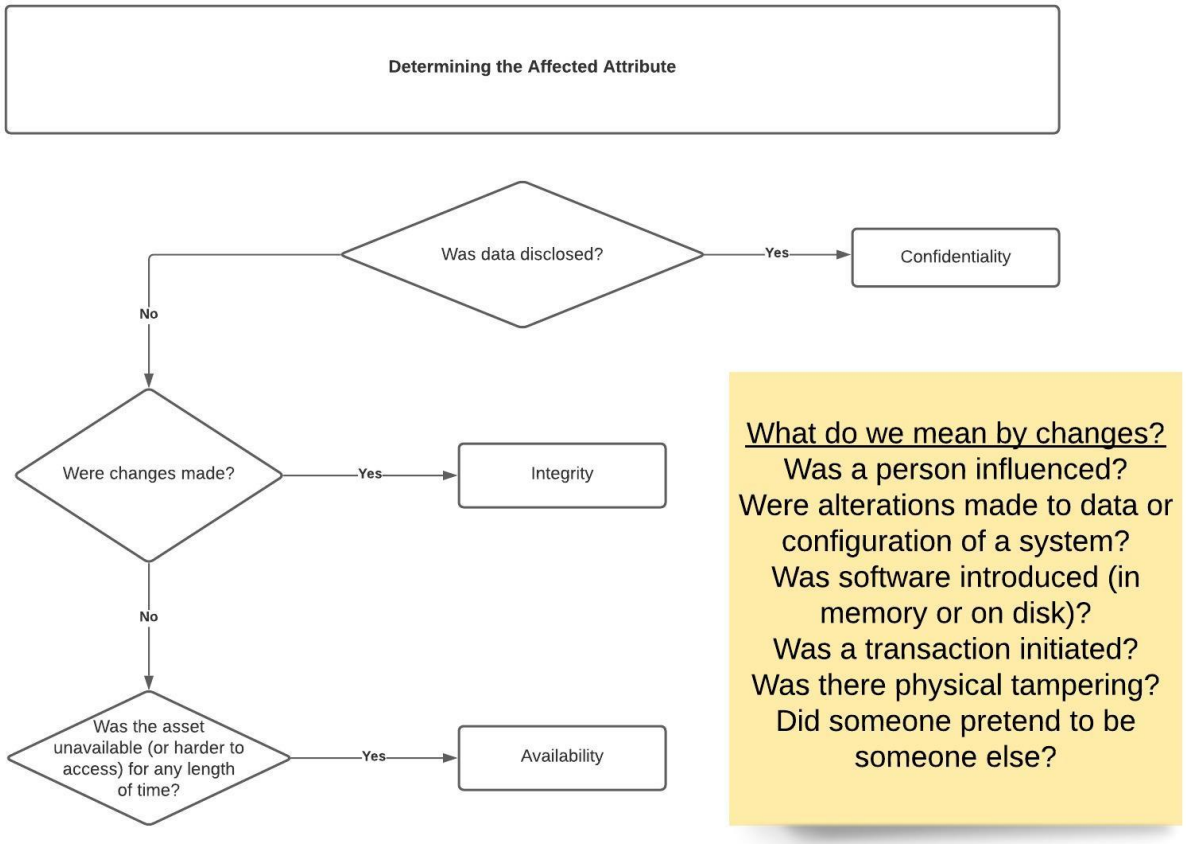
You also have to account for the device the person was using when they were persuaded by the threat actor to give up their credentials (or whatever else the attacker was trying to get them to do). This is most commonly a User Device (like their laptop or desktop) or perhaps even their cell phone. Then, once you have identified the initial compromised assets, you need to account for whatever the attacker did with the results of a successful social attack. Did they gain access to the persona's credentials? How did they use those?


Determining the type of Asset
Is it a physical copy of data? (CD, paper, tape, USB drive)
Yes
Media
No
Is it a network device? (router, switch, CCTV cameras)
Yes
Network
No
Is it a server? (web app, database, file server)
Yes
Server
No
Is it a end-user's device? (desktop, laptop, phone)
Yes
User
No
Is it a Terminal or Kiosk? (Gas pump, ATM)
Yes
Terminal
No
Was it a Person? (Social influence)
Yes
Person

You may need to traverse this decision tree multiple times to account for all the assets compromised in a case (and remember, it may be a confidentiality compromise, but it might also be an integrity violation you need to account for).

Attribute Coding Tree

The Attribute is where the CIA triad comes into play in VERIS. Confidentiality, Integrity and Availability are the cornerstones of information security, and when any of the three are violated, you'll need to account for it.



The tree is short—just the three pillars of the CIA triad. Confidentiality compromises are the true definition (at least to the DBIR and VERIS) of what a data breach is. Someone who was not authorized was able to view (which includes taking a copy of) data at some point in the course of the case. We also have an option to mark Confidentiality as "At-risk" for cases where it is suspected, but cannot be confirmed. The stolen laptop is an example where making that confirmation is quite a challenge. But it is definitely at-risk, since the owner has lost custody of the device.

For Integrity, things get more complex. The Integrity of an asset indicates that it reflects reality—that no changes are made to the system that are unauthorized. So even when the asset is a person and they are influenced to do something they should not be doing, it is an Integrity violation (alter behavior). The same goes for a computer where someone installs malicious software (whether it is on disk or just exists in memory), Software installation is the violation in this case. There are a large number of ways to violate the integrity of a system—even changing the data or configuration files would be a violation.

Initiating fraudulent transactions (think money transfers) is also an Integrity violation, as is impersonation. If an attacker compromises an executive's email account and sends an email to the finance worker to have them pay a fraudulent invoice, you have multiple integrity violations. Some integrity violations are physical in nature—putting a skimming device on an ATM is a good example. That would be considered tampering.

Finally, we have availability violations. If an asset is stolen, we lose access to that asset. Likewise, if the data on a server is encrypted by ransomware, we lost access as well. Denial of Service attacks that bring down infrastructure ensuring the legitimate traffic cannot reach it is another example of an Availability violation. When you look at the enumerations for each of these attributes, you

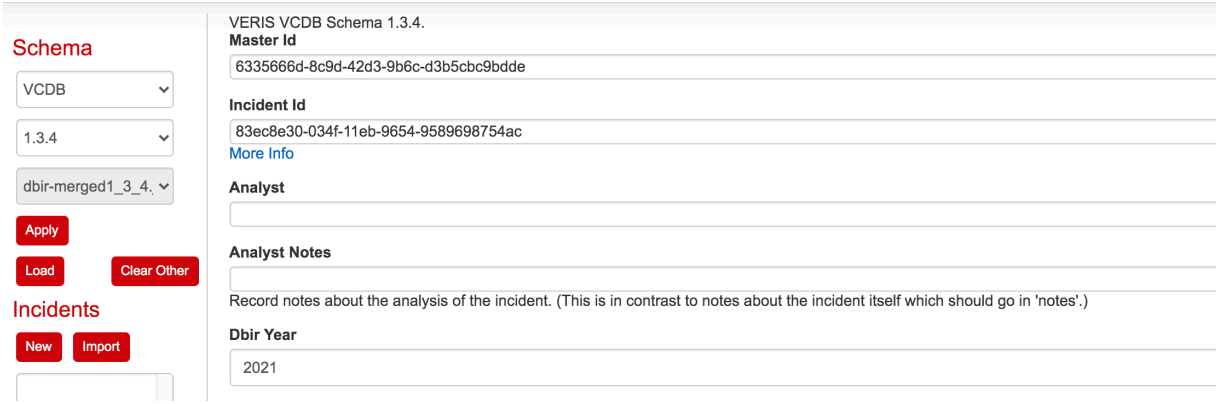
can see why the list is so comprehensive. Most of them have come about from actual cases that were coded by the DBIR team.

Incident Basics

Incident Overview

When we are looking at a case report and need to code it in the Webapp, we are first presented with a number of fields (assuming this is VCDB) that, while necessary, don't directly get into the meat of what happened. However, they are recorded, primarily as housekeeping information that we need as we manage a dataset this large.

The following screenshots are all from the VERIS webapp, setup to use the VCDB schema, and VERIS version 1.3.4. The dropdowns under Schema (below) can be changed based on the source of the data being coded—such as internal cases worked by our own forensic practitioners (VzIR), partner cases contributed by our data contributors for the report, or even VerisC cases being coded by what we call the VERIS community (which is anyone who is coding their own cases, whether they plan to share data with us or not).

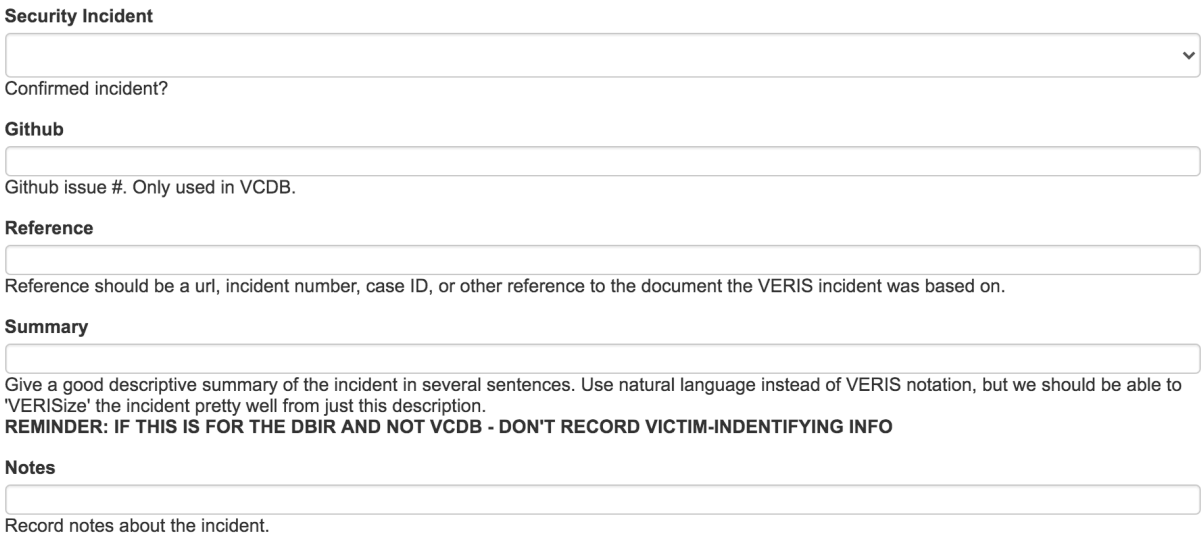


The Master Id and Incident Id are assigned automatically, and can be ignored. Analyst will be the person who is coding up the case. In the VCDB coded cases,

we use our GitHub Id in this field, but if you are using this in a corporate setting, it could be your username on your company's systems.

The Analyst Notes are used for anything the analyst thinks needs to be recorded. We usually use it if there is some follow-up that should be done before the json is considered validated, or if there are other things that need to be recorded that are pertinent to coding the case up.

The DBIR Year is used by the DBIR team to indicate which report year the json file should be included in. Just because the incident happened in 2015 doesn't mean it was discovered until 2019, and so it should be included in the dataset for the 2019 (or 2020, depending on the time frame when it was discovered) report. When selecting the data you are interested in (and using VCDB data), it can be an important distinction to know if you are interested in the year an incident occurred, the year it was discovered, or the report year it showed up in the dataset (which is probably mostly of interest to the DBIR team).



We record whether the case is a security incident (which indicates one of the CIA triad was violated) or a confirmed data breach, where the Confidentiality aspect of the CIA triad has been violated. While all breaches are security incidents, not all security incidents are breaches.

The GitHub field records the value of the GitHub issue number record, so that the json that is generated can be traced back to the issue that it was coded from. This is a VCDB schema field only, ad won't be displayed on other schemas.

The Reference field is where we record the URL(s) for the issue. It is important to record all of the URLs listed in the GitHub issue, as over time many links can go stale, so having additional references will help ensure we still have some working reference down the line.

The Summary field is where you record (in general terms) what happened. It should be long enough to stand on its own and let someone who is reading it know why you made the choices for Action, Actor, Asset and Attribute in the json. If all the referenced URLs become stale, this summary should be enough to justify why a case was coded the way it was.

The Notes field is for any case notes that the analyst thinks should be included, but there are no fields for recording them.

Finally, we get to the Source Id field, which is where we record where that case came from. In VCDB, it will always have the value "vcdb". If you are using this to code your own data, it might be something specific to your organization.

The confidence field lets us indicate how much confidence we have in the data for this case. There are cases where we are more and less confident in the reliability of the data presented. It is an unfortunate fact that not everything you read on the internet can be trusted. This field lets us record whether we think the details should be taken with a grain of salt.

Summaries

We wanted to take a moment to discuss the Summary field in our cases. Often, the summary is all the information that we are provided by our contributors with which to code a case. Therefore, it is extremely important that we obtain as much detail as possible in the summary field. We need to be able to use this summary to make the same decisions as are in the coding trees, and if details are left out, or are ambiguous, we have to go with "unknown" as the value for some of the important fields.

The summary field is also very important when coding VCDB cases, as the links can become stale due to removal of old content to make room for new material. Furthermore, we rely on the summary to make the decisions that you will see in the coding trees. If details are omitted or are ambiguous, we have no alternative other than to choose "unknown" as the value for important fields. Consequently, having a very detailed summary is important for people who may be looking at the record long after the case has been closed.

A good Summary will have unambiguous detail about the actor. Sometimes we see things like "a former employee stole data and took it to their new employer." While this sounds like it might be enough information, was the person actually still an employee when they took the data? Or did they get back into the system (were their credentials not disabled as part of their off boarding when they left the company) after they were no longer an employee? This is the difference between an external and an internal actor breach, therefore granular detail such as this is crucial to getting the case coded correctly.

Even better, the basics about the victim organization should be included. We

really like to have both the industry the organization operates in, and the (rough) number of employees to classify the organization's size. As much detail as possible regarding the victim organization should be included (excluding the name unless it is a publicly disclosed breach). The industry vertical of the organization along with its basic geographical location, and employee count enables us to do much more in-depth analysis and comparison than we could otherwise do.

Speaking of industry vertical, determining which industry an organization belongs to can be a bit tricky at times. We often see cases in which the contributor refers to the victim organization in generic terms such a 'technology company' or a 'nonprofit'. These labels are vague and lead to guesswork on the part of the person coding or reviewing the case. Rather than reinvent the wheel for labeling organizational types, we chose to use the NAICS (North American Industry Classification Standard).

You can ask yourself the following questions to help determine if you have what is needed to make a useful summary:

1. Who was the victim organization? How much do you know about the company that suffered the breach/incident?
2. Who caused the event? Do they work for the victim organization? Is there a business relationship involved?
3. How did the event unfold? Include what actions were taken and how they were accomplished to the extent it is known.
4. What has been impacted because of those actions? Was data breached? Was there fraud? Was someone fooled into sharing their login credentials? Was money transferred or data encrypted?
5. How was all this discovered? Did they notice it themselves, or did someone from the outside tell them?
6. When did all of this happen? When was the first action? When was it discovered?

A good summary would read something like this:

> "A large (over 1000 employees) healthcare (Naics 62) company located in California experienced a phishing attack that led to the theft of 4,891 medical records from their EHR database. The attack started with a phishing email targeting an administrative employee, received on 2/4/2020. The attacker was able to gain the victim's credentials and used them to log into the EHR system. The attacker also used the compromised email account to send email masquerading as the victim to further compromise several other employee's account credentials. Ultimately, 10 sets of credentials were compromised, along with their desktop computers. Malware was installed on those computers. The database was accessed several times, and the medical records of 4,891 patients were copied and sent offsite on 2/15/2020. The attacker then triggered the malware (ransomware) to encrypt the medical records database and demanded a ransom of $1 million USD in bitcoin on 2/16/2020. The victim did not pay the ransom, but restored from their backups over the period of two weeks."

This has sufficient information to code the case and have known values for the majority of the datapoints that we deem most important. The Actor is external to the organization, with a motive of financial gain. The actions include Social (phishing), Hacking (use of stolen creds), and Malware (ransomware). The assets include Person (end-user), User device (desktop or laptop), and Server (database and email). The Attributes include Confidentiality (Yes), with both medical data and credentials as the stolen data types; Integrity violations of Alter behavior and Software installation; and the Availability violation of Obscuration to account for the data being encrypted.

You see we have good timeline data in the summary as well. We know the

Compromise was phishing on 2/4/2020. The Exfiltration was on 2/15/2020, and the Discovery was on 2/16/2020 when the actor triggered the encryption and the ransom message was displayed. This also lets us know the Discovery method was Actor disclosure.

The organization demographics allow us to classify the organization's size and NAICS code, and the geographic location of the company. All of this in a one paragraph narrative summary, so that if this is all we have, the case's coding can be verified as correct. That is our ultimate goal for the Summary field.

Timelines

The first section in the webapp is where we record the timeline information (when we have it). There are several different metrics to track within the timeline. They include the Incident date/time (when the first action occurred), the Compromise date/time (when the actor was able to successfully get access), the time to exfiltration of data, how long it took the victim to discover they'd been breached, how long it took to contain the incident, and finally, when the organization notified the data victims that their information had been compromised. We may not get all of these (and some are quite rare to get information on in the public dataset), but if you are recording data of your own breaches, you should have this information. If you don't, start looking at why you don't.

This is the timeline graphic from the 2019 DBIR, and it shows why we record this information.

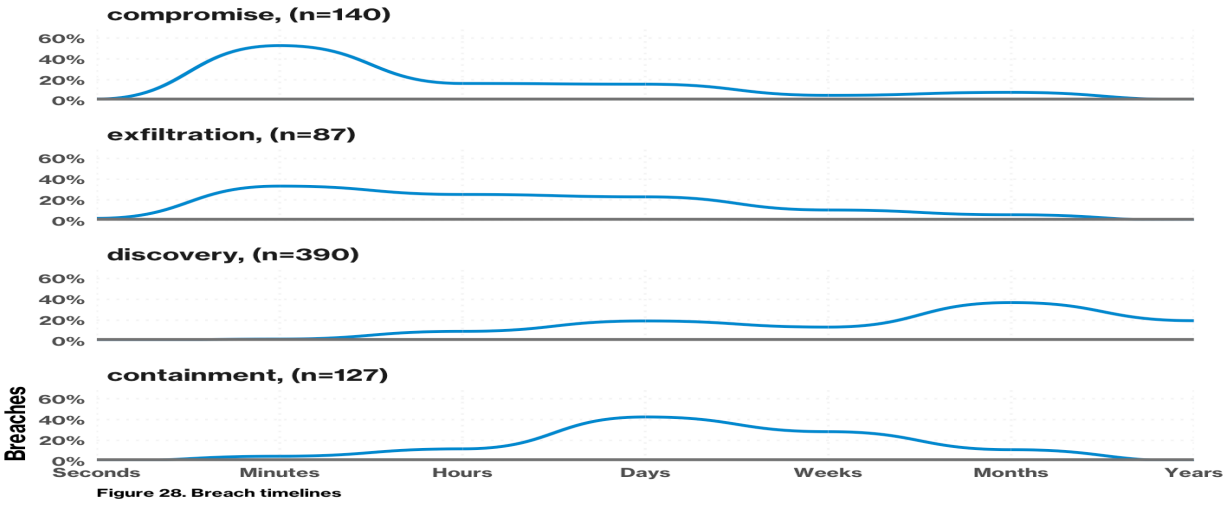


Figure 28. Breach timelines

We are able to show that system compromise happens very quickly in most cases—within seconds to hours. While exfiltration takes a little longer, discovery of the breach takes quite a bit longer still. The most common in that graphic was months to years to discover the organization had been compromised. Containment, in contrast, happens within days typically.

By recording this information for your own organization, you can establish these metrics, and show how they are changing over time based on the controls you are putting into play. Other metrics you might also want to look at include the difference between how long it typically takes your organization to catch a breach, and how far back your logs go to allow you to accurately scope an intrusion. These two need to be in sync so that you do not find out you've been breached, and also find out you cannot tell how the intruder got in, and what activities they have been up to.

Victim Demographics

As mentioned in the VERIS Overview section, we record victim demographics (when we can get them) for each incident. They include fields such as the name of the victim (for VCDB cases), the organization's size (in terms of how many employees) and their industry. We really like to have the last two whenever

possible, since these are two of the ways we commonly split out the data in the DBIR.

We also record what country the victim organization resides in (and if it is multinational, then either choose the location where the breach occurred if known, or the headquarters of the organization), and the region code so we can look at geographic differences.

If there are multiple organizations affected by a breach, they can be listed (along with their NAICS code) in the Secondary Victims section of the webapp. If the case was a supply chain breach where a partner was breached and many of their customers were affected (and were thus having to report individual breaches), we code this from the perspective of the partner who was breached rather than create a record for each of their customers (see Coding from the Partner Perspective) .

A Note about NAICS

We on the VERIS team use the North American Industry Classification System or NAICS codes to designate an organization's industry membership. Some of the NAICS codes are quite straightforward, like Healthcare (62). It is easy for most companies to know if that is where they would belong, and to make sure they're looking at the right industry when they read the DBIR statistics. However, there are a number of the NAICS codes where the grouping is very widely distributed, and it may be difficult for people to know for certain where they should be looking. To that end, this is one of the most straightforward references for what kinds of organizations are in each of the NAICS industries.

http://www.farsmarterbids.com/reference/naics-list.php

One example we ran across was the Arts, Entertainment, and Recreation (71) NAICS code. It is made up largely of live performances, rather than those that

are recorded (like film and TV). However, when presenting to a film industry group, we discovered that they had been looking at this industry as their own.

Here is an excerpt from that NAICS code:

Sector 71: Arts, Entertainment, and Recreation

- 711: Performing Arts, Spectator Sports, and Related Industries
- 7111: Performing Arts Companies
- 71111: Theater Companies and Dinner Theaters
- 711110: Theater Companies and Dinner Theaters
- 71112: Dance Companies
- 711120: Dance Companies
- 71113: Musical Groups and Artists
- 711130: Musical Groups and Artists
- 71119: Other Performing Arts Companies
- 711190: Other Performing Arts Companies
- 7112: Spectator Sports
- 71121: Spectator Sports
- 711211: Sports Teams and Clubs
- 711212: Racetracks
- 711219: Other Spectator Sports

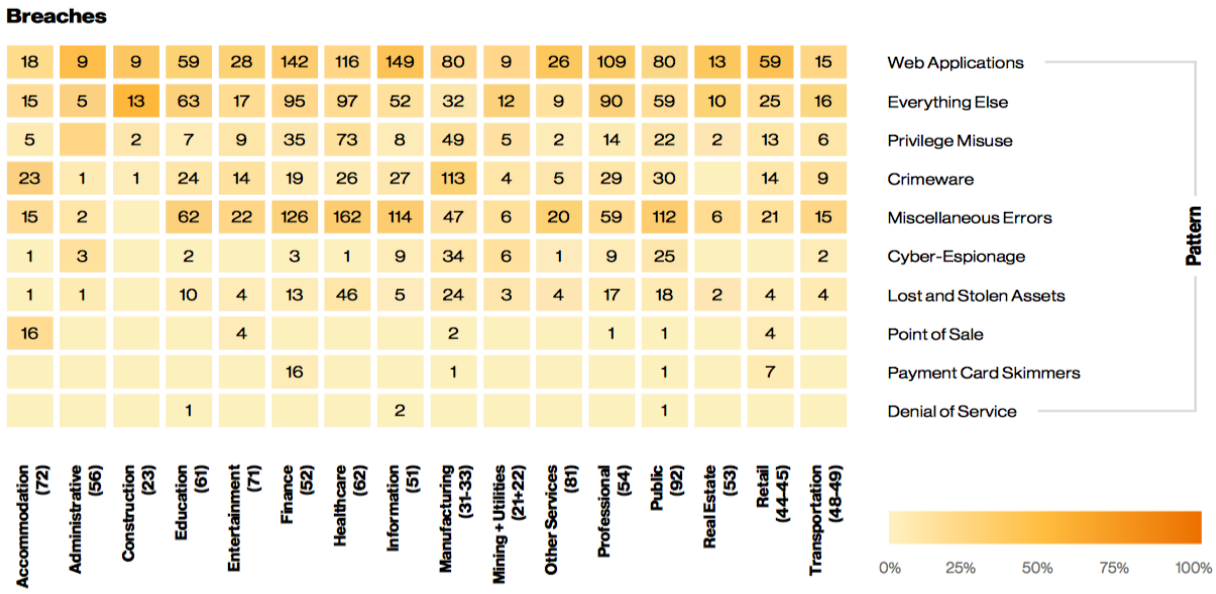
While it goes on to a much longer list of organization types, it does not include television or film companies. In fact, that kind of organization belongs in the Information (51) NAICS code.

Sector 51: Information

- 512: Motion Picture and Sound Recording Industries
- 5121: Motion Picture and Video Industries
- 51211: Motion Picture and Video Production
- 512110: Motion Picture and Video Production
- 51212: Motion Picture and Video Distribution
- 512120: Motion Picture and Video Distribution
- 51213: Motion Picture and Video Exhibition
- 512131: Motion Picture Theaters (except Drive-Ins)

- 512132: Drive-In Motion Picture Theaters
- 51219: Postproduction Services and Other Motion Picture and Video Industries
- 512191: Teleproduction and Other Postproduction Services
- 512199: Other Motion Picture and Video Industries
- 5122: Sound Recording Industries
- 51221: Record Production
- 512210: Record Production
- 51222: Integrated Record Production/Distribution
- 512220: Integrated Record Production/Distribution
- 51223: Music Publishers
- 512230: Music Publishers
- 51224: Sound Recording Studios
- 512240: Sound Recording Studios
- 51229: Other Sound Recording Industries
- 512290: Other Sound Recording Industries
- 515: Broadcasting (except Internet)
- 5151: Radio and Television Broadcasting
- 51511: Radio Broadcasting
- 515111: Radio Networks
- 515112: Radio Stations
- 51512: Television Broadcasting
- 515120: Television Broadcasting
- 5152: Cable and Other Subscription Programming
- 51521: Cable and Other Subscription Programming
- 515210: Cable and Other Subscription Programming

This is an incomplete listing of this NAICS code, but you get the idea. When we break out our data in the DBIR, one of the most popular graphics is the one that shows the attack patterns by industry. This is the example from the 2020 DBIR:



So you can see here why it is so important to know which is the correct NAICS code for your organization. Hopefully the link above will help you determine that in short order, if it was ever in question.

Discovery Method

The Discovery method tells us how the organization found out about the security incident. This is frequently correlated with how long it takes to discover the breach—because if the method is internal, it stands to reason that it is faster than if you have to wait for notification from outside your organization.

Internal discovery methods include:

Antivirus
Financial audit

Fraud detection
HIDS
Incident Response
IT audit
Log review
NIDS
Reported by user
Security alarm
Other
Unknown

External discovery methods include:

Actor disclosure
Audit
Customer
Fraud detection
Law enforcement
Monitoring service
Security researcher
Other
Unrelated party
Unknown

One of the areas people get tripped up in determining the Discovery method is when the data is offered for sale on the internet, and found by a security researcher who monitors the underground forums. Is this Actor disclosure? Is this Security researcher? We debated this within our team and decided that when the data is put up for sale, it counts as Actor disclosure even when it is found by a Security researcher. Our reasoning is that the Actor had to make the call that they would put the data up for sale, so even though it may have been

subsequently found by the researcher, it was first disclosed by the person who stole the data.

Case Coding Examples

Case Coding Style Examples

The easiest way to discuss how cases should be coded is to show examples from the VERIS Community Database (VCDB). The JSON filename of the specific case is included, so the full record can be accessed from the GitHub repository to see all the details. For brevity, all of the VERIS variables are not included in the example diagrams. The diagrams are broken into several major sections. The Victim Demographics lists the most important pieces of information that we prefer to see in all cases wherever possible. Since this is VCDB, the organization's name is typically included, but in some cases where it is not available, it can be left blank. (If you are using this for your own organization's cases, you can use that field as you need—some use it to record the division of their company, for example.) That is how any variable we do not know the answer to is treated—it is more important to have accurate data than it is to have all fields filled out. So we try hard not to infer information unless we have a solid basis to make an assumption.

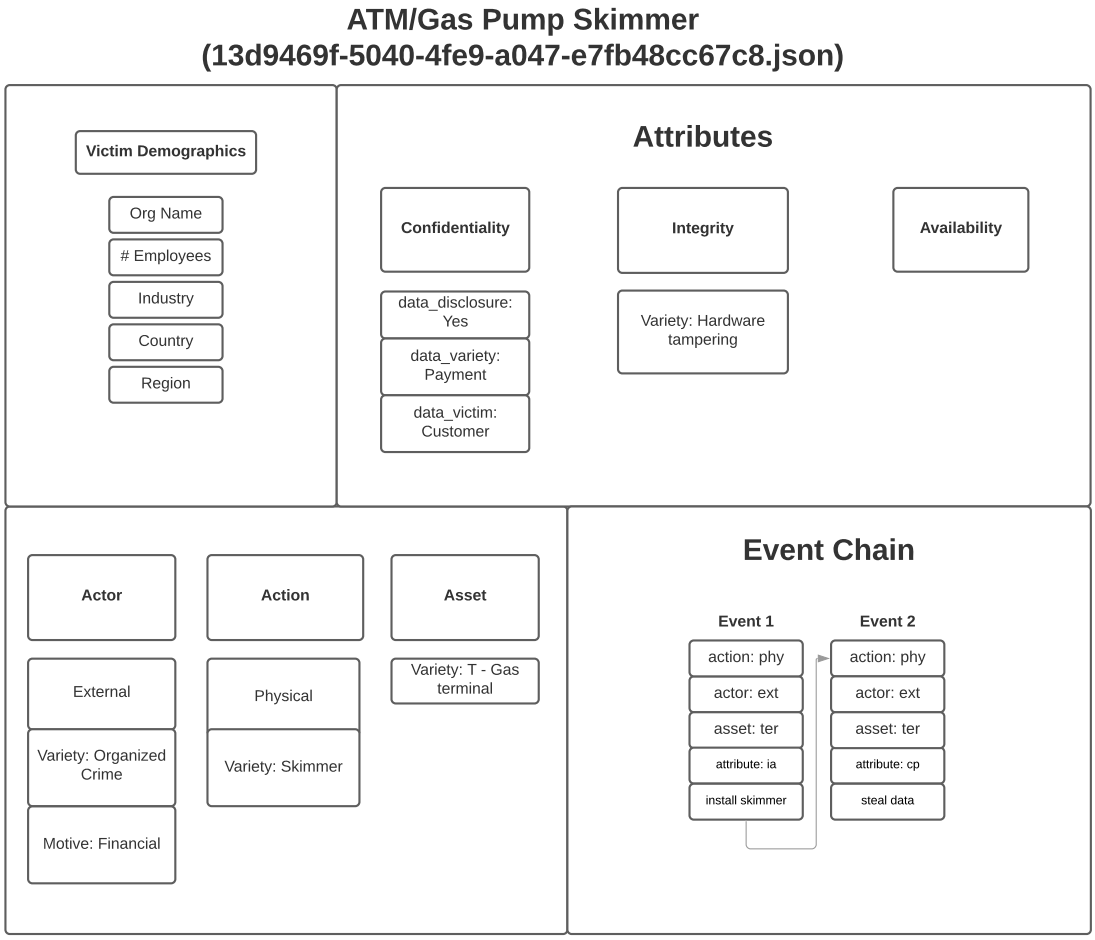
The examples in this guide are organized by the Action type. VERIS has seven action types:

Error
Environmental
Hacking
Malware
Misuse
Physical
Social

While an incident can have more than one action, and frequently do, the coding

examples are designed to show how to code the most common cases we encounter. As edge cases are identified, this guide will be updated with new examples and diagrams to provide a reference as to how they should be handled.

We have tried to make learning VERIS as easy as possible. To that end, we developed case diagrams that show you the most important fields within VERIS for each case. They are also used to highlight aspects you might not always remember to include, depending on the case. Here is an example from the ATM Skimming example.



The diagrams are all in a similar format. The type of case is listed at the top, along with the JSON filename that this represents. That way, you can go into the VCDB repository and see the actual json that was generated by the WebApp for the case you are looking at. You will see much more detail in the json file than we depict in the diagram, since the idea with the diagram is to make it simple

and easy to understand. Json files can be a bit intimidating at first glance, so this will hopefully help with that.

Each diagram has four or five main sections. The Victim Demographics section has the basic information we really want about the victim organization. If you are implementing VERIS in your own organization, you can decide what to record here—cost center is a potential value people track for granularity. We divide our data for the DBIR into several different views, including looking at regional differences, industries, and organization size, so we really like to get that information if at all possible. Your areas of interest may vary based on the kinds of questions you plan to ask your data once it is collected.

The Attributes section deals with the CIA triad and how they are affected by the incident. Data_disclosure has options of "Yes", "Potentially", "No" and "Unknown". If the compromise of the confidentiality of the data is confirmed, choose "Yes". If it is at-risk but cannot be confirmed, choose "Potentially". The other choices are self explanatory, it is the first two that people get confused with.

The Actor will be External, Internal, Partner, Unknown, or some combination of those. You are allowed to have multiple actors, actions and assets at the same time. We wanted VERIS to reflect reality as closely as possible, and certainly there are no restrictions on the actors not to involve other actor types, or to restrict themselves to just committing one action in pursuit of getting to their goal. Therefore, VERIS does not have these constraints either, and this makes it quite powerful for recording details of very complex cases.

Actors have varieties and motives. Actions have varieties and vectors. If an action is accidental, the motive is going to be NA, because it is an unintentional action. You can see the enumerated lists of what the varieties and vectors are in the VERIS schema.

Finally, we have the event chain section. We record the events that led to the breaches as step by step chains, allowing us to analyze how breaches occur, and evaluate where controls might be best placed to cause the attacker to need to work harder to reach their goal. The idea being the more expensive in terms of time and effort you make it to breach your systems, the higher the likelihood the

attacker will go somewhere the pickings are easier.

The first several cases we go over as examples have just one event in their chains. That is because we start off with the error action. Once we get into more complex cases (or where we have multiple CIA triad attributes affected) we need more than one event in the chain to represent the case. You can see in this example, we have two events in our chain. The first event is the physical attachment of the skimmer to the ATM. This is an integrity violation of tampering. Then to show the data compromise, we have a second event where they steal the data the skimmer has captured. We will go over this in more depth when we look at this case. But this gives you information on how to read each of the diagrams.

Error Examples

Error Overview

The Error action, by definition, is non malicious in nature. We only record actions that directly resulted in the event—not indirect or passive actions. Otherwise any time there was a breach, the choice not to implement every control on the planet could be pointed to as an error. The Error variety enumerations are as follows:

Classification error
Data entry error
Disposal error
Gaffe
Loss
Maintenance error
Misconfiguration
Misdelivery
Misinformation
Omission
Physical accidents
Capacity shortage
Programming error
Publishing error
Malfunction
Unknown
Other

Are all of these encountered regularly? Not so much. Most common are the Loss, Misconfiguration, Misdelivery, Publishing and Disposal errors. The others come up now and again, but some are quite rare.

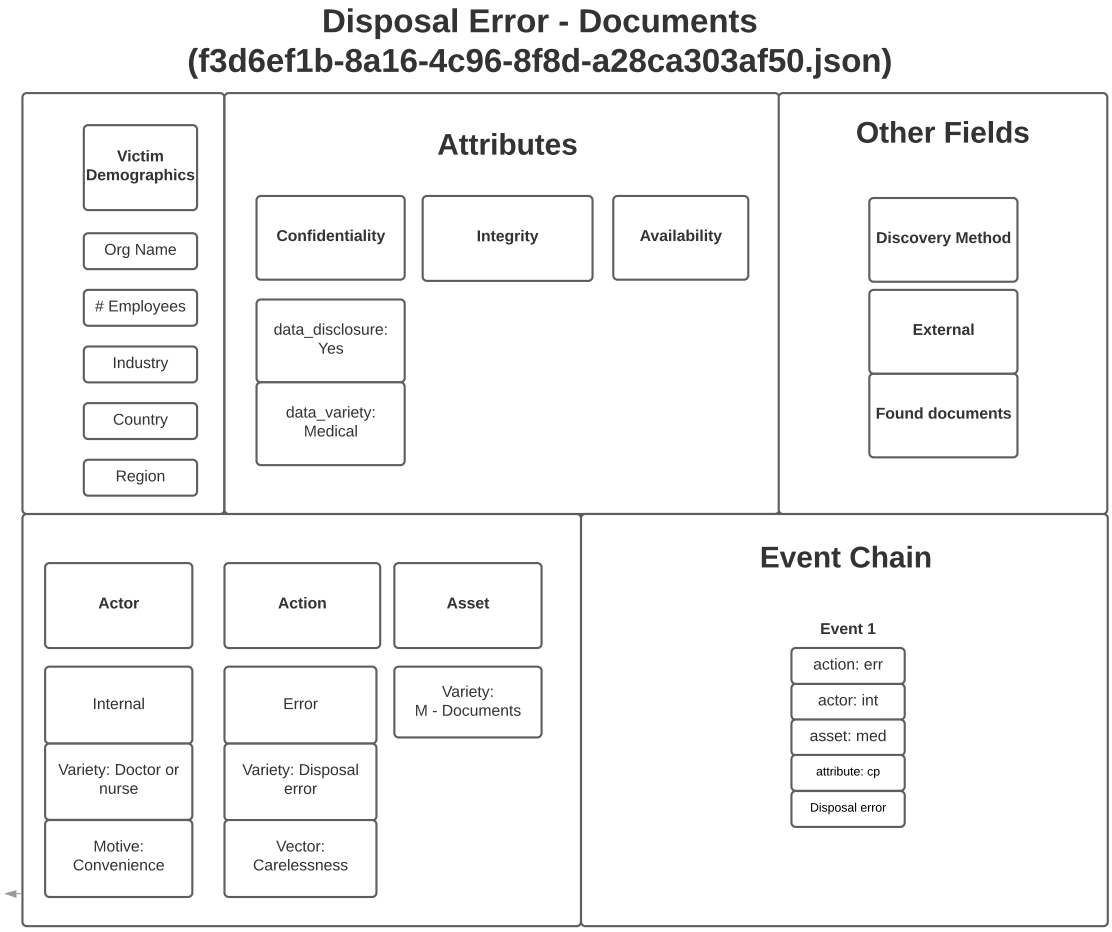
We have case examples from the common Error types where we go into detail for each case. Let's move into those.

Disposal

Disposal errors occur when either electronic or paper assets are disposed of without taking into consideration the need to protect the data on them. For paper documents, the typical solution is to shred the documents prior to disposal, or in some cases, this is performed by a partner vendor for an organization under contract. Paper documents are the special case in that we consider any time they are lost (or stolen) it is considered a confirmed compromise of the confidentiality of the data printed there. The person who finds them typically has to read them to even realize they contain sensitive information, and there is simply no control that will combat the vulnerability.

Our first case example is the disposal of paper documents. You can see the Victim Demographic information we most want to have filled out. We tend to split our data along organizational size, industry and regional lines, so we like to have that information. If you are working on implementing VERIS in your own organization, and there are multiple locations, it is nice to see where the hot spots are as well.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/8c8798f2999fd19330a30ca685fc31994351956b/data/json/validated/f3d6ef1b-8a16-4c96-8f8d-a28ca303af50.json



This is a case where a psychiatrist stored patient files in the basement of a home, and allowed a tenant to have a key and access to the basement (particularly when workmen were present to do maintenance). When the tenant was told they had to move, they alerted law enforcement about the improperly stored medical files. The doctor eventually reported the incident to the Health and Human Services as a breach.

As the tenant had a key for an unspecified amount of time, they essentially could have read through any of the files. They had to have at least looked at them to determine what they were and there is mention of the data types they contained in the article. Under the Attributes section of the diagram, you can see that we have data_disclosure: Yes, and the data_variety is medical. There is also space to indicate number of records total and in the case of multiple types of

records, you can designation that information as well. As mentioned, the goal of these diagrams is to give you a visual aid of the most important aspects of this type of case. The WebApp will allow you to record much more detail.

The Other Fields section is to highlight some of the fields that are unique to this kind of case, that people sometimes may forget to record, even when they know it. Found documents as the discovery method is an enumeration we created when this was the way so many of these cases were discovered. It is an external method—meaning it is most frequently someone discovering the documents who do not work for the victim organization.

The Actor in this case is an internal employee with a variety of Doctor or nurse. The motive is convenience, since it was just easy for them to leave those old files in their basement rather than properly get rid of them.

The Action is Error, with a variety of Disposal. The vector is carelessness, as it frequently is in these cases. It isn't that there wasn't a process to handle the disposal of these documents in place—it was that the person didn't follow it, and took the easy way out. We see this quite a lot—I'm sure it is no surprise to incident responders that people take unapproved shortcuts.

Finally, the Asset is Media - Documents. Now while you can certainly make disposal errors with electronic equipment as well as paper documents, it is far more frequently to see the asset being paper with this type of error. However, we do show the difference in how to handle the two types of assets when we get to the Misdelivery error action—the other error type that is frequently either documents or electronic data.

The final section is the event chain. We record each action that occurred in an incident and do research on those attack paths based on this information. You can see this is a one-event chain. Most error actions tend to be one-event chains, as a matter of fact, which is good. You certainly don't want the employee to

continue making error upon error to cause even more mayhem.

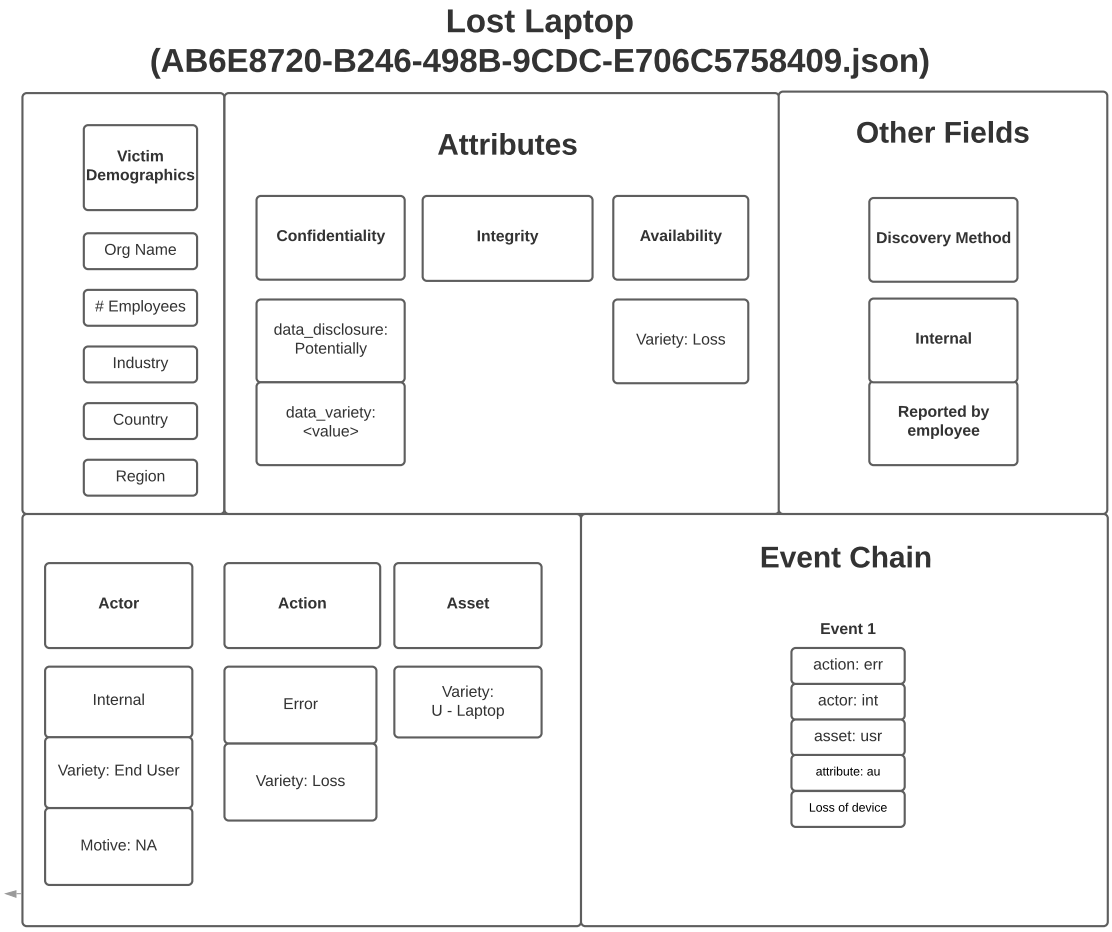
Lost Laptop

People lose things all the time. It is when the lost item contains unprotected sensitive data that the breach reporting comes into play. While paper documents are lost, those cases are very similar to the disposal error in how the asset is handled. So this time we are looking at an electronic asset rather than a dead tree.

"While at a conference in Buenos Aires, a Boston Children's Hospital employee lost a laptop containing a file with information about 2,159 patients, including names, birth dates, diagnoses, and treatment information. The laptop was password protected but not encrypted, according to a hospital press release. "

The important thing here is that the laptop was not encrypted. If it was encrypted, it would not be considered a confirmed compromise of the confidentiality of the data (unless the passphrase was also taken in human readable form). It is password protected (which is a very weak control) and thus it is not a confirmed compromise in the Confidentiality section of the attributes. In this case, we mark data_disclosure as Potentially, and it is not counted in the breach total of our report. It will show up in the incident total, and how you handle your data is entirely up to you.

Here is the link to find the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/8c8798f2999fd19330a30ca685fc31994351956b/data/json/validated/AB6E8720-B246-498B-9CDC-E706C5758409.json



There is no integrity violation, but there is an availability violation, given that the access to the asset has been lost. Discovery method in this kind of case is Internal, reported by the employee having to tell their boss they lost the laptop—never a fun conversation. When we get to the physical examples, take a look at the discovery method for stolen laptop, and how it changes.

The actor involved is the employee. End User is a catchall for someone we don't really have details about their role, but we know they are an employee. The motive is NA, since it is an unintentional action.

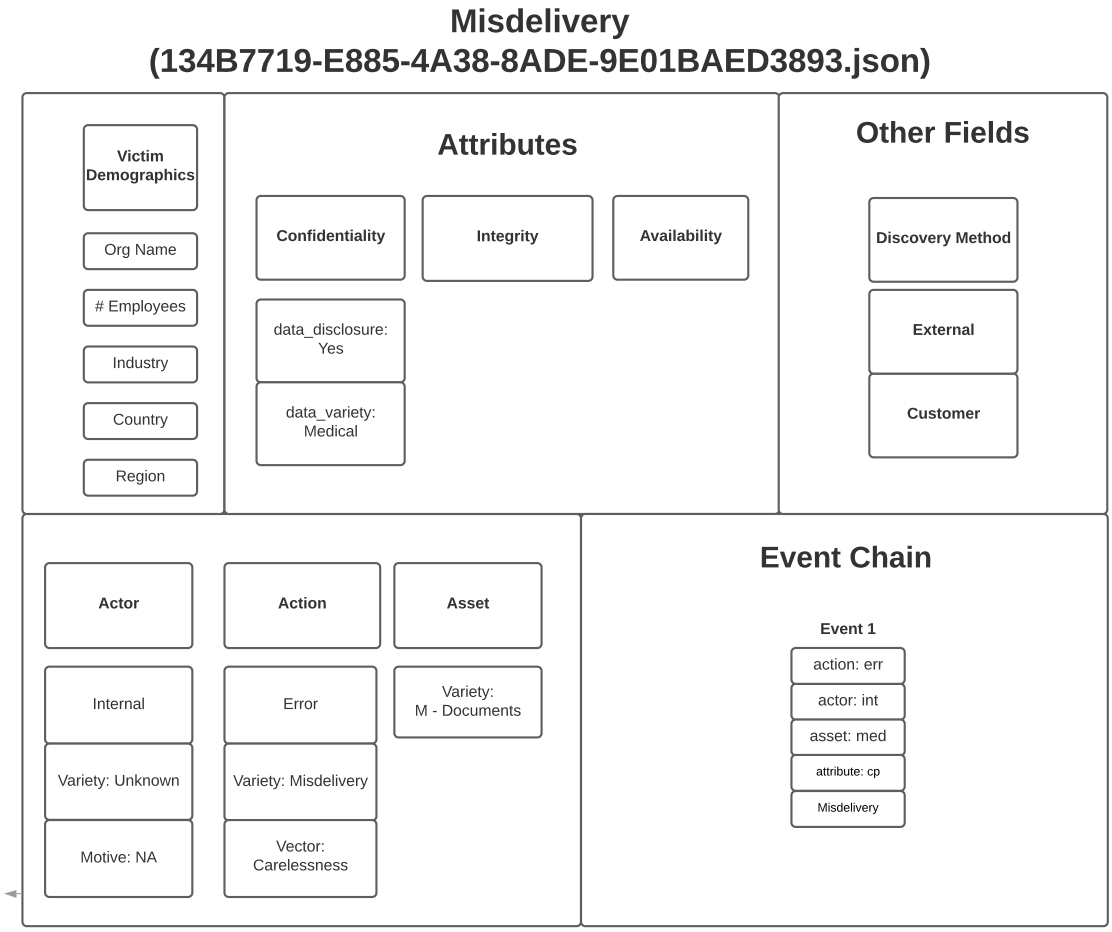
The action is Error with a variety of loss, and the asset is the User Device of Laptop. Again, the event chain is a chain of one step.

Misdelivery (Documents)

Misdelivery is when data (whether on paper or in electronic format) goes to the wrong recipient. Common cases are documents in a mass mailing when the envelopes and their contents become out of sync, and suddenly your customers (patients, constituents) and receiving someone else's information. Another common case is when someone either selects the wrong distribution list in email, or attaches the wrong file to the email. We will cover both paper and electronic misdelivery in this section so you can see the differences.

First we look at the document case. Since there are no controls on printed human readable material (paper), we mark the data_disclosure as confirmed. To be found, and identified as sensitive data, at least some of it must be read.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/8c8798f2999fd19330a30ca685fc31994351956b/data/json/validated/134B7719-E885-4A38-8ADE-9E01BAED3893.json



This was a mailing gone awry, and the customer is the one who realized they got the wrong person's data, and contacted the victim organization. The actor was an internal person, variety was unknown, and motive does not apply to an error action.

> "Veteran A's means test was returned to Veteran B with a request to complete the form in pen. The form contained Veteran A's name, address, social security number, date of birth, employment information, and financial information."

The error variety was misdelivery, and we marked carelessness on the vector. This is frequently the case with error actions—because someone just wasn't

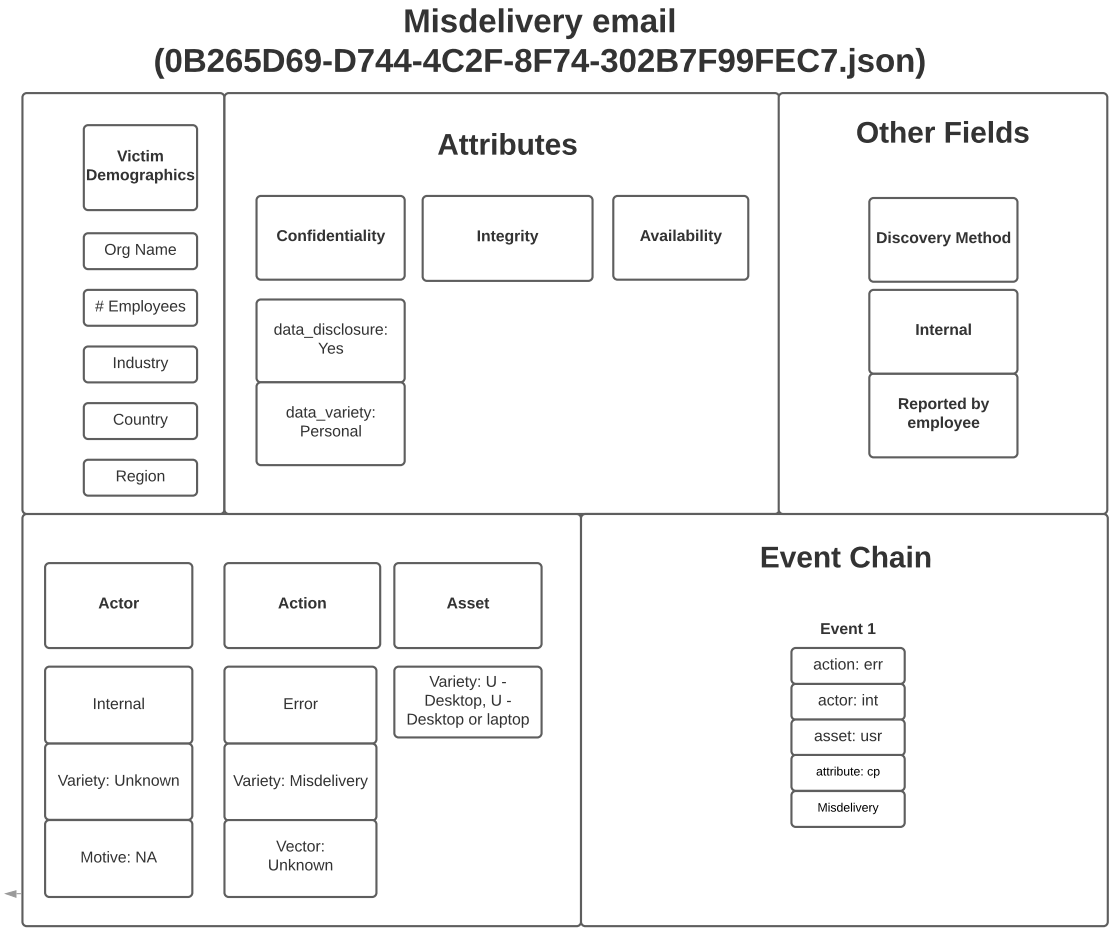
paying enough attention. The asset was printed documents as mentioned above. The event chain is just one step to cause a breach in most errors.

Misdelivery (Electronic)

While paper misdelivery is quite common, it doesn't mean the same error can't happen with electronic data. In fact, it is quite common. In our case below, someone sent email with data they should not have to a business partner. Here is an excerpt from the notification.

> "The Jonathan M. Wainwright Memorial VA Medical Center in Walla Walla is offering free credit monitoring for up to 1,519 veterans whose personal information, including social security numbers, was inadvertently emailed to an external education partner on Nov. 1."

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/8c8798f2999fd19330a30ca685fc31994351956b/data/json/validated/0B265D69-D744-4C2F-8F74-302B7F99FEC7.json



The diagram doesn't change all that much from paper misdelivery. The main differences are in the discovery method. Usually this is a case where the person who sent the email realizes what they did fairly quickly. Other times, it will be the recipient who alerts the sender that they got data they didn't expect. In this case, it was the former.

The other difference here is the asset involved. The variety of User device - Desktop or laptop is the parent of the User device - laptop, so both are selected when the child is selected.

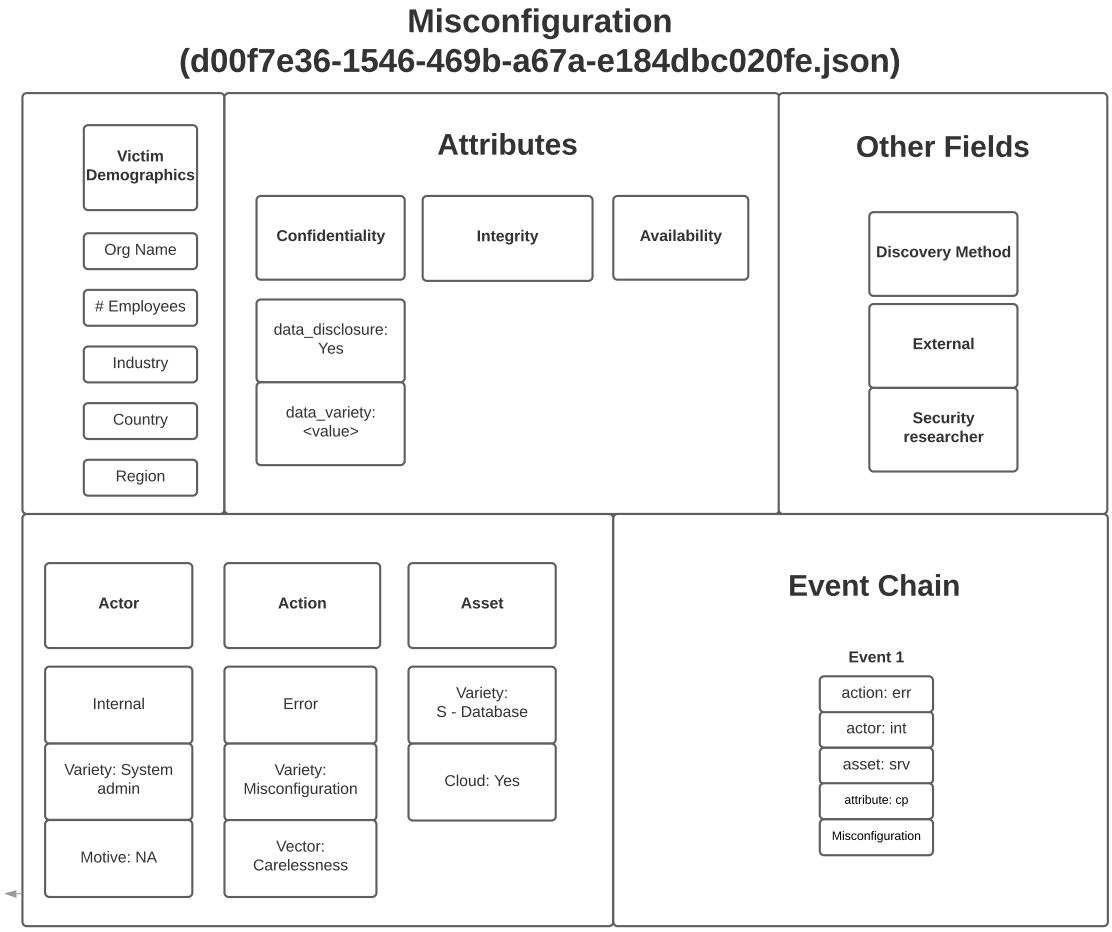
So the main difference we see in these electronic cases are in the assets and discovery methods. The rest are very similar—it is an error committed by an internal actor, with no malice.

Misconfiguration

Historically, this case was when someone would misconfigure their site when they installed it, and eventually somehow the private data would come to light. These were fairly rare, and mainly happened when a developer did an oops. Nowadays, it is primarily the place where we put all those instances where someone stood up a database in cloud infrastructure, but failed to put any controls on it. Then a security researcher finds it, and hilarity ensues. Ok, not really, but you get the idea.

Since you are more likely to encounter the latter case, we are modeling it as an example. The main difference between the two would be the discovery method. External/customer versus External/security researcher are the two options for these two scenarios.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98f0f09240f4a92948a3c2198e00e3bf47e2edc/data/json/validated/d00f7e36-1546-469b-a67a-e184dbc020fe.json



Data_disclosure tends to be confirmed in these cases, given that they are frequently found by people who then contact the organization. The act of finding the data and looking through it to try and identify who it belongs to is a breach in itself, since the security researcher is not authorized to see that data. However, consider that this is someone who is trying to make the notification—how many others viewed the data (and potentially took a copy) without saying a word?

> "On July 26th (I) discovered a non password protected elastic data set that contained 5.2 million documents in total. Immediately, I knew this information should not be publicly accessible and began trying to identify the ownership. Upon further investigation the data appeared to belong to

an organization called Leadership for Educational Equity (LEE)."

The Actor is internal, system admin—someone who has the ability to access large amounts of data and either change the configuration or stand up a new cloud datastore with it. The motive is NA, as it is an Error action. The variety in either case is Misconfiguration, and the vector is carelessness. The asset is a database, and these instances tend to be cloud based when the security researcher is the one to find it—they use specialized search engines such as Shodan to find these unprotected datasets, and they are usually on some cloud infrastructure.

The event chain is just one step, as it only takes one step for most errors to cause breaches.

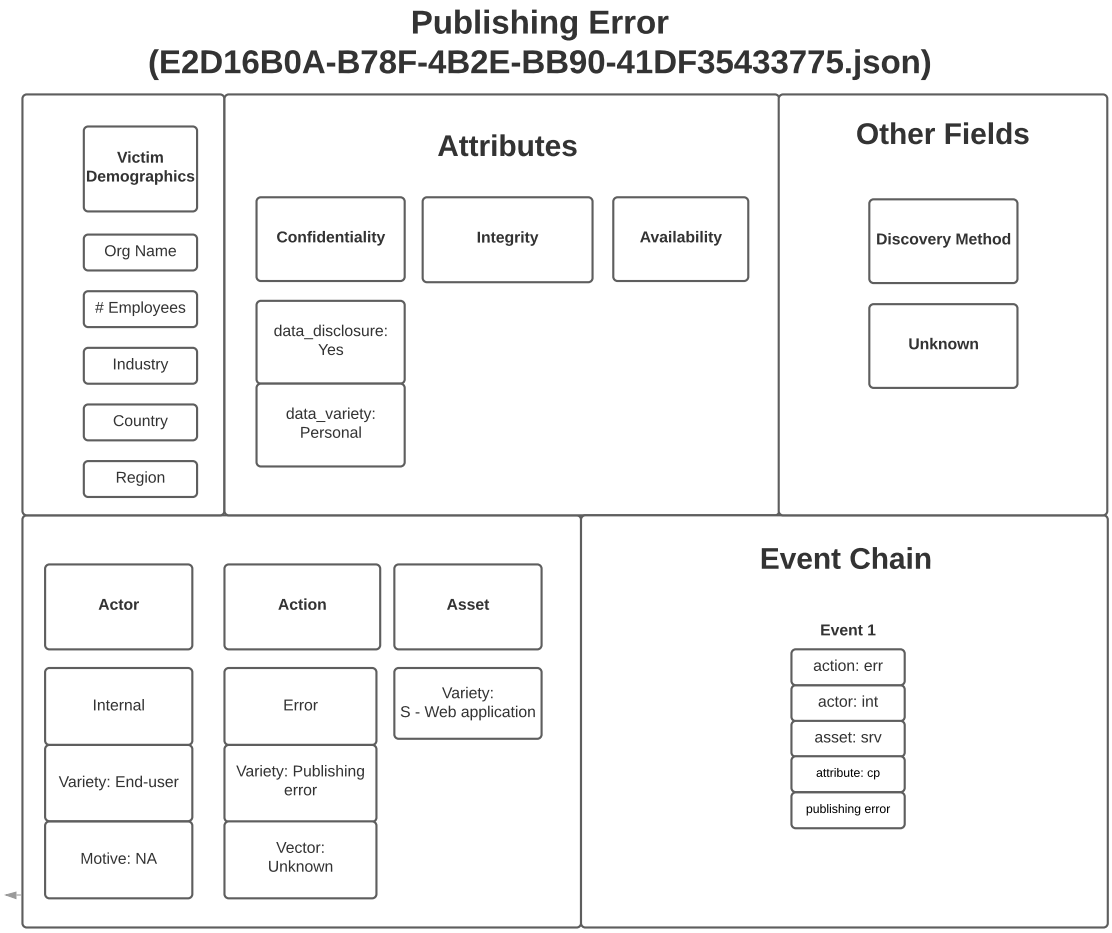
Publishing

A publishing error occurs when private data is put on a public facing website and is somehow accessible to the world. The website is then indexed by the helpful search engines and frequently comes to light when a customer googles themselves and finds their data flapping in the wind. We typically find this as a result of some change control process failure, where the site's controls were suspended to make some kind of change, and then after it was completed, the controls were not put back in place.

It also occurs when private data that should not have been uploaded gets put on a site that is open to all.

"A police activity log for the period of January 7 through January 13 was published on the Littleton department's website. Someone forgot to remove personal details from the log and the sensitive information was available online for 10 days. Names, Social Security numbers, dates of birth, and addresses, were available between January 14 and January 24."

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/8c8798f2999fd19330a30ca685fc31994351956b/data/json/validated/E2D16B0A-B78F-4B2E-BB90-41DF35433775.json



For this case, we have data_disclosure as confirmed. The data variety exposed was Personal. There was no integrity or availability violation. The discovery method here is not known but this is a case type where the customer frequently alerts the organization.

The actor responsible is Internal with a variety of End User. That likely means we didn't have detail into who did it, and End User is a safe bet. Motive of NA because it is not an intentional action.

The asset is the web application server, no surprises there. And finally, the event chain of just the error action is nice and short.

Environmental Examples

Environmental Overview

Actions where the environment is the cause are quite rare, but they do happen. In this section, we will explore some actual cases where data breaches were caused by Mother Nature, sometimes with some help from humans, sometimes not.

The enumerations for Environmental actions are as follows:

Deterioration
Earthquake
Electromagnetic interference (EMI)
Electrostatic discharge (ESD)
Temperature
Fire
Flood
Hazmat (Hazardous material)
Humidity
Hurricane
Ice (and snow)
Landslide
Lightning
Meteorite
Particulates (e.g., dust, smoke)
Pathogen
Power failure
Tornado
Tsunami

Vermin
Volcano
Water Leak
Wind
Unknown
Other

Have we seen examples of all of these? Well, no. When the framework was conceived, there was a brainstorming session for what this might look like, and this was the result. We have seen a few of them, but they are so rare that when we actually run across a valid environmental breach case, we on the DBIR team get all excited. I know, we're geeks.

Hurricane

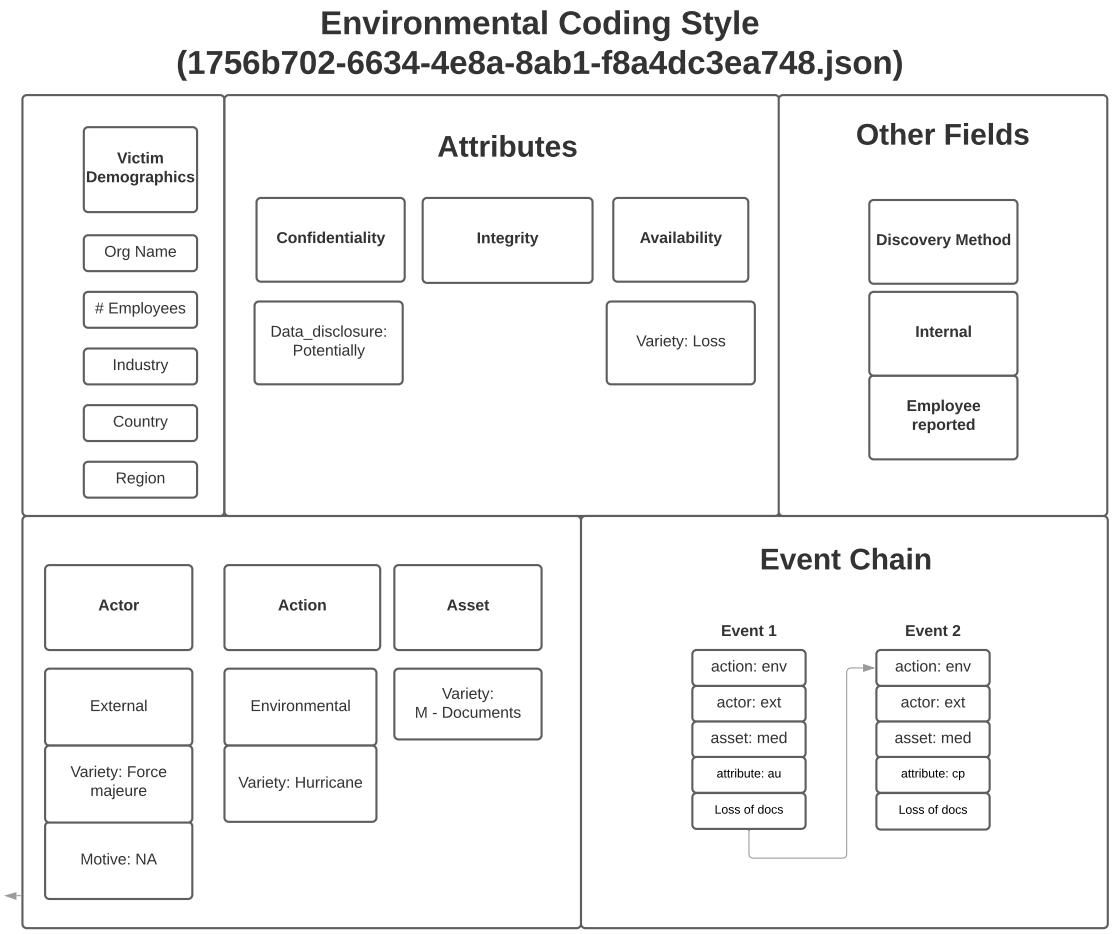
Yes, Mother Nature is occasionally the cause of data breaches. In this case, we have a hurricane sweeping away the paper records of this company.

"We've lost a number of the records as a result of the flooding and while we've made all efforts to try to salvage some of them, some of it was just not possible," Weech told reporters, during a press conference announcing restoration efforts for the storm-ravaged facility….Weech noted that the PHA is currently unable to quantify how many medical records were lost as a result of the Category 5 storm.

So, given we don't have an indication whether people found the records among the storm debris, we have marked this as data_disclosure: Potentially. Usually we mark breaches involving documents as confirmed, but there is a good chance here that the papers were simply destroyed by the storm and the likelihood is low that people will find them in readable format.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/

vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/1756b702-6634-4e8a-8ab1-f8a4dc3ea748.json.



The availability loss is also recorded. The duration would be Never (for the likelihood that they will ever get the records back, or could read them if they did). That is the usual value for a loss event (as well as physical theft) although we do sometimes see articles where the device or records were recovered after a time.

The actor here is considered External, and Force majeure is another term for Mother Nature here. The motive is not applicable, as we don't attribute malicious intent to hurricanes. The Action is Environmental, and we have hurricane as our variety. The asset is documents, as the excerpt from the article

above mentions.

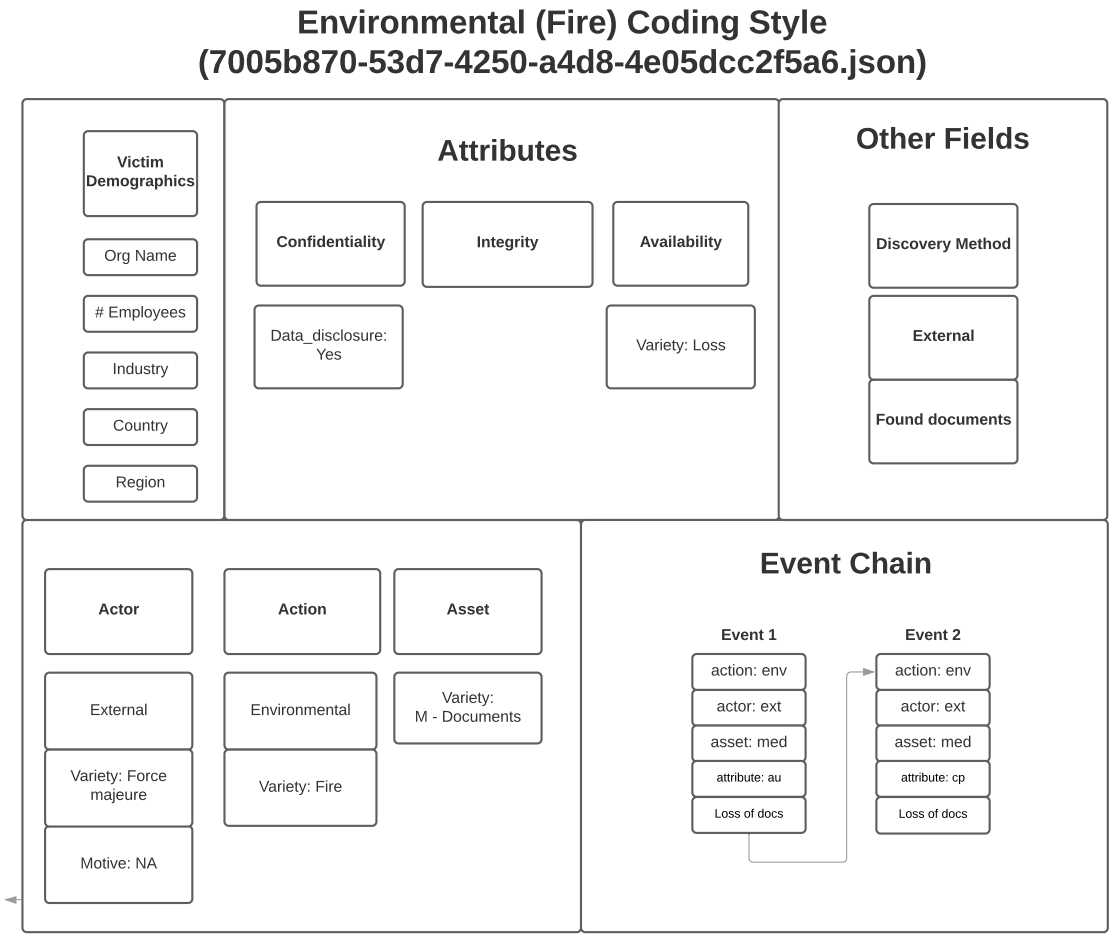
One thing to note here is the event chain has two events. When you have a case like this where there is a confidentiality and an availability impact, it is coded this way. The events look like two things, but it is really to account for both of the attributes. It is a weakness in the schema and how we handle event chains, but it is the best way we could come up with to account for this.

Fire

You probably don't think of fire as a lead to a data breach, but it does happen, however rarely. Our case was a large fire at a records storage building, and the paper records were strewn by the fire and the efforts to douse it.

> "No lives were lost in the huge fire that gutted a storage building on the Brooklyn waterfront over the weekend. But the flames put plenty of lives on display as the crumpling warehouse belched up its contents: decades' worth of charred medical records, court transcripts, lawyers' letters, sonograms, bank checks and more.
>
> "They're like treasure maps, but with people's personal information all over them," Spencer Bergen, 24, said of the half-charred scraps that he said he had seen strewn around the Williamsburg neighborhood as far inland as Berry Street, several blocks from the warehouse."

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/7005b870-53d7-4250-a4d8-4e05dcc2f5a6.json



There are many similarities in this case when compared with the Hurricane case. The data disclosure is confirmed—the article goes on to talk about people on the beach sifting through the papers. The loss of the documents violates the availability attribute.

The discovery method is one we created just for these kinds of cases—where documents are lost and someone stumbles upon them.

The actor again is Force majeure, not Mother Nature this time, but still an outside force in terms of the fire. The action is environmental with a variety of fire. The asset is the M - documents, where the "M" stands for Media.

Again, we have a two-step event chain for a single environmental action. This

is to account for both the confidentiality loss and the physical availability loss.

Hacking Examples

Hacking Overview

The hacking action is a broad action category with a wide variety of options to choose from. As new techniques are encountered, we update VERIS when we find something is popping up in enough cases to be worth tracking. In our current version, here are the hacking varieties:

Abuse of functionality
Brute force
Buffer overflow
Cache poisoning
Session prediction
CSRF: Cross-site request forgery
XSS: Cross-site scripting
Cryptanalysis
DoS: Denial of service
Footprinting
Forced browsing
Format string attack
Fuzz testing
HTTP request smuggling
HTTP request splitting
Integer overflows
LDAP injection
Mail command injection
MitM: Man-in-the-middle attack
Null byte injection
Offline cracking
OS commanding

Path traversal
RFI: Remote file inclusion
Reverse engineering
Routing detour
Session fixation
Session replay
Soap array abuse
Special element injection
SQLi
SSI injection
URL redirector abuse
Use of backdoor or C2
Use of stolen creds
XML attribute blowup
XML entity expansion
XML external entities
XML injection
XPath injection
XQuery injection
Virtual machine escape
Unknown
Other

Quite a list, isn't it? We primarily get this level of detail from cases worked by our forensic analysts and our partners who work cases. We don't usually get this information from the VCDB dataset, but sometimes we do strike gold. As with other aspects of VERIS, you can multiselect from this list when the detail is rich and you know exactly what the attacker did. It should be noted that if you select varieties where the actor exploited a vulnerability, you should also choose "Exploit vuln" in addition to more specific choices.

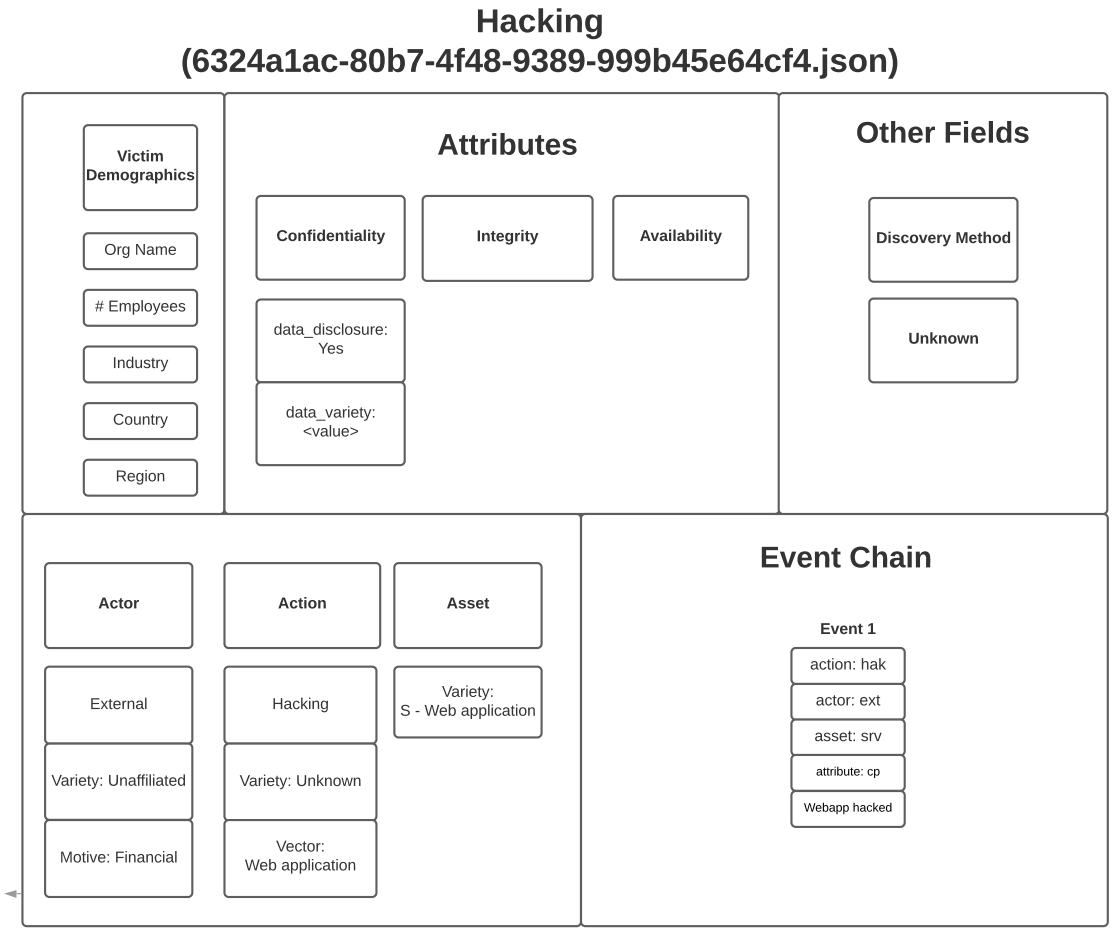
General Hacking

We have a lot of cases where we know the action was hacking, but we don't know all that much in the way of detail. This is particularly true of cases in VCDB, because we are at the mercy of what the organization was willing to disclose, and how interesting the reporter found the details. That combination frequently means we don't know much.

> "An Arkansas unemployment system centered around the COVID-19 pandemic has shut down after a data breach of secure information. The system data breach was caused by applicant who illegally accessed the system, according to Hutchinson. A probe will determine whether any personal data from applicants was obtained. If any individuals had their data compromised, they will be notified and steps will be taken to address the situation, including possible credit monitoring. Nearly 30,000 people have applied to the program."

For this case, we know they were able to hack into the web application, and that based on the data they took, they were financially motivated. We don't know how they got in, but we know that the vector was the web app.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/6324a1ac-80b7-4f48-9389-999b45e64cf4.json



We also didn't know how the break in was discovered, so that is marked as Unknown. We believe it is better to indicate that we don't know than it is to speculate and go down that slippery slope of introducing possible scenarios rather than sticking to the facts that we do know.

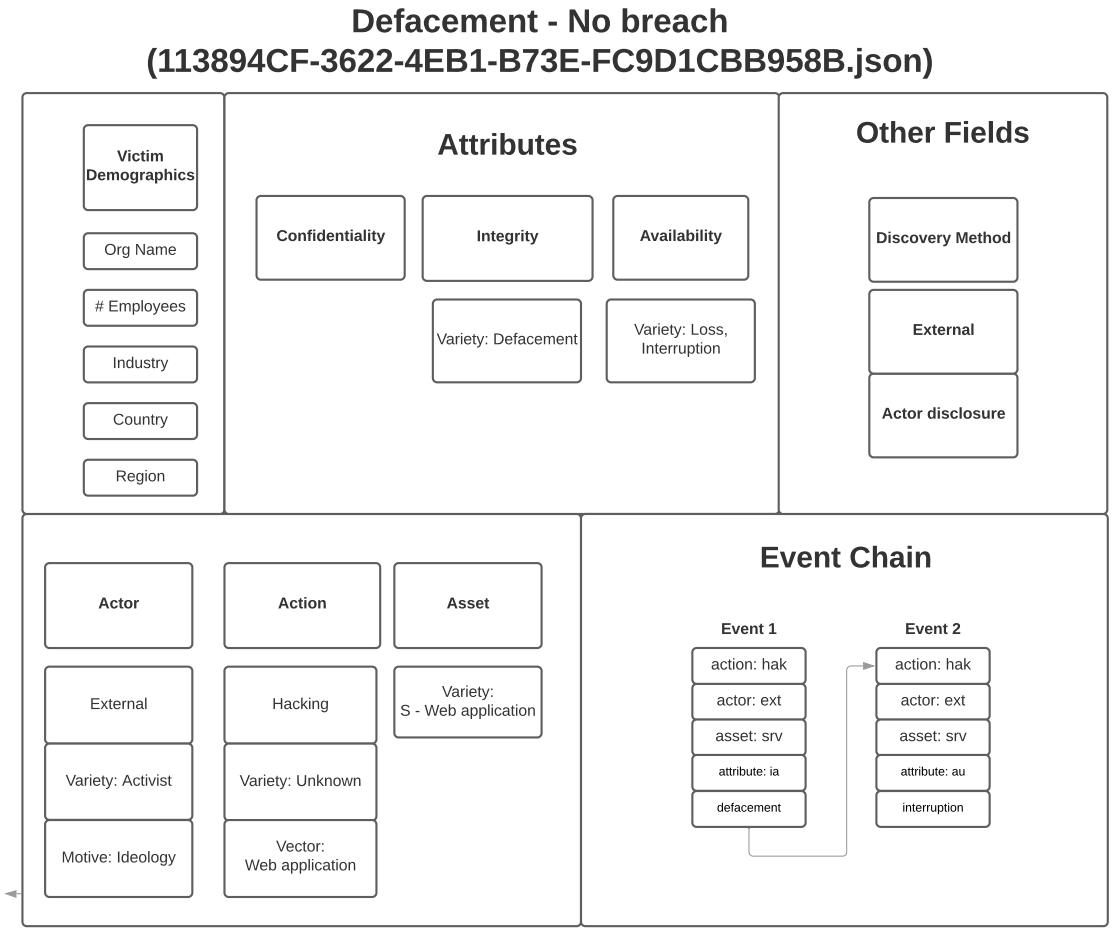
Defacement

This case type used to be more popular than it has been recently. There was a time when people with the motive of ideology would hack into websites and use

them for electronic graffiti. They would post their own views and generally wreak havoc on the website of the victim organization. It still happens, but it doesn't get the attention it used to in the news media, which is a large source of our data breach reports.

Defacement cases typically do not involve an actual data compromise. The actors have been more concerned with writing on the walls about their favorite subjects than seeing if there is any data to steal. So when this case was coded up, you can see there is not even a Potentially violation of the data_disclosure field under Confidentiality. Integrity, however, is trampled. The variety of modify data is how we represent the actions of the attacker in changing the website. In this case, there is also disruption of the availability of the website for its intended purpose.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/8c8798f2999fd19330a30ca685fc31994351956b/data/json/validated/113894CF-3622-4EB1-B73E-FC9D1CBB958B.json



This case shows us the Other Fields discovery method of External; Actor disclosure. We see this whenever the action of the attacker draw attention to the incident. We also see this in ransomware cases, where there is typically a popup of some kind letting the victim know how to pay the person who is causing their systems to be encrypted.

The actor in these is typically external of type Activist. Their motive is ideology—they're doing this for their cause. The asset is typically a Server - web application. Their victims may have been targeted because of the ideology of the actor—it may be that their business is in some way viewed as related to the causes the actor is championing or vilifying. It may also be that they simply were vulnerable to something the actor knew how to exploit. However, we also have seen a lot of cases where they just do it for the lulz, in which case, it would

be coded as “fun.”

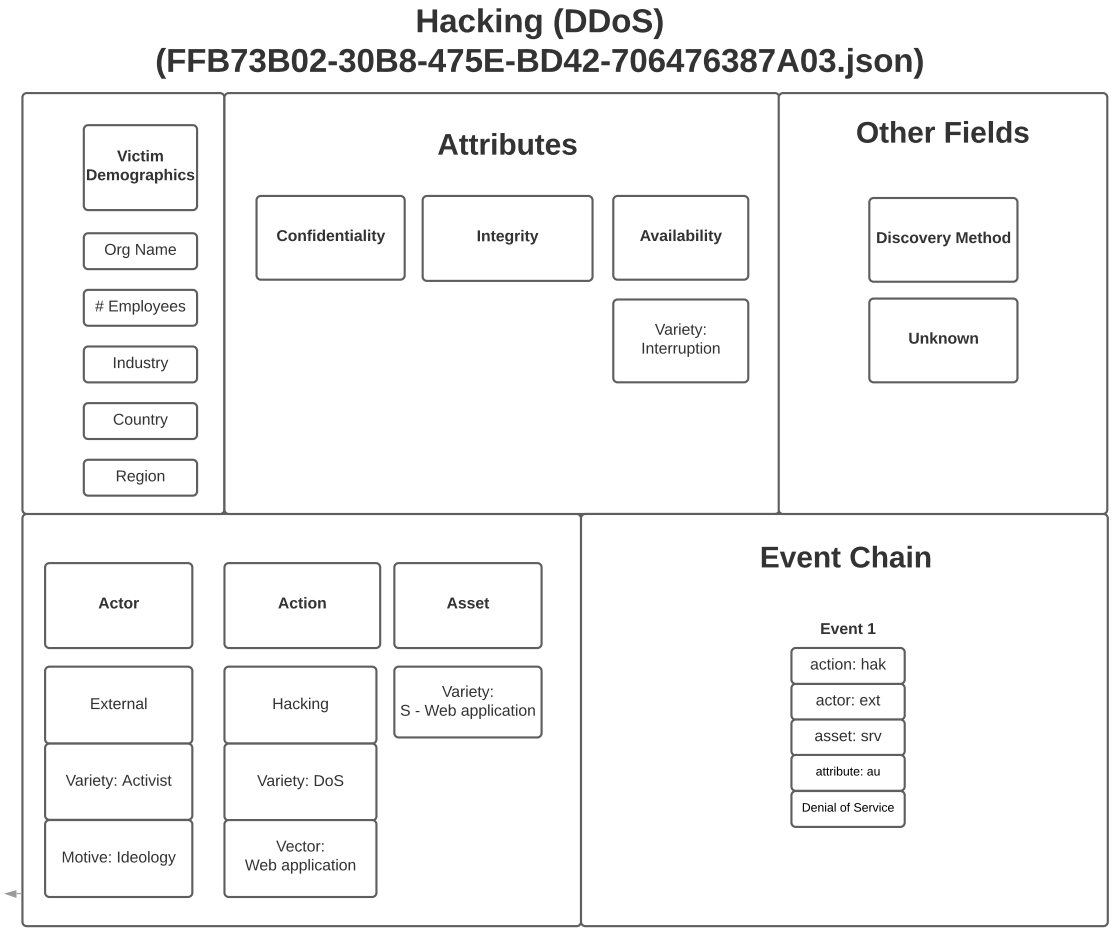
The event chain is two steps. First they hack into the web server, violating the integrity by changing the data on the site. Then they caused the availability interruption.

DDoS

A case where we don’t have a breach, but we have a security incident is the Denial of Service (DoS), or it’s big brother, the Distributed Denial of Service (DDoS) attack. This is where the asset is flooded with so much bogus traffic that it cannot perform the duties assigned to it for legitimate traffic. This is frequently employed by activists, but is also common in Asia between gambling website vendors. In our cases, it is the former we look to for an example of coding this kind of case.

“Anonymous DDoS Attack Against Israel Leverages Botnet Network.”

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/FFB73B02-30B8-475E-BD42-706476387A03.json



These cases are rarely breaches (although they can happen as smoke screens for someone trying to get in amongst the noise), and are largely just availability violations for the duration of the attack.

The variety is interruption, for the duration of the attack, or until the organization can mitigate it with various service offerings from vendors that handle these kinds of network attacks.

The discovery method in this case is listed as unknown, but it is common that they are noticed by employees being alerted by monitoring their network infrastructure.

The actor in this case is a member of the Anonymous hacking group, and was

carried out for ideology motivations. The action is hacking with a variety of DoS —which covers both the DoS and DDoS varieties. The vector is the web application, and they are typically attacked because they are a visible presence of the target on the internet.

The event chain is just the one availability event.

Malware Examples

Malware Overview

Our first example is a standard ransomware case. Until recently, these cases were treated as incidents, not confirmed data breaches, unless there was a compelling reason to treat them otherwise. However, since November 2019, certain groups of ransomware attackers have changed their tactics to take a copy of the data prior to triggering their encryption routine and then using the threat of data exposure as leverage to encourage victims to pay their ransom demands. So we will show an example of a standard ransomware case, as well as a case where data confidentiality has been confirmed to be compromised.

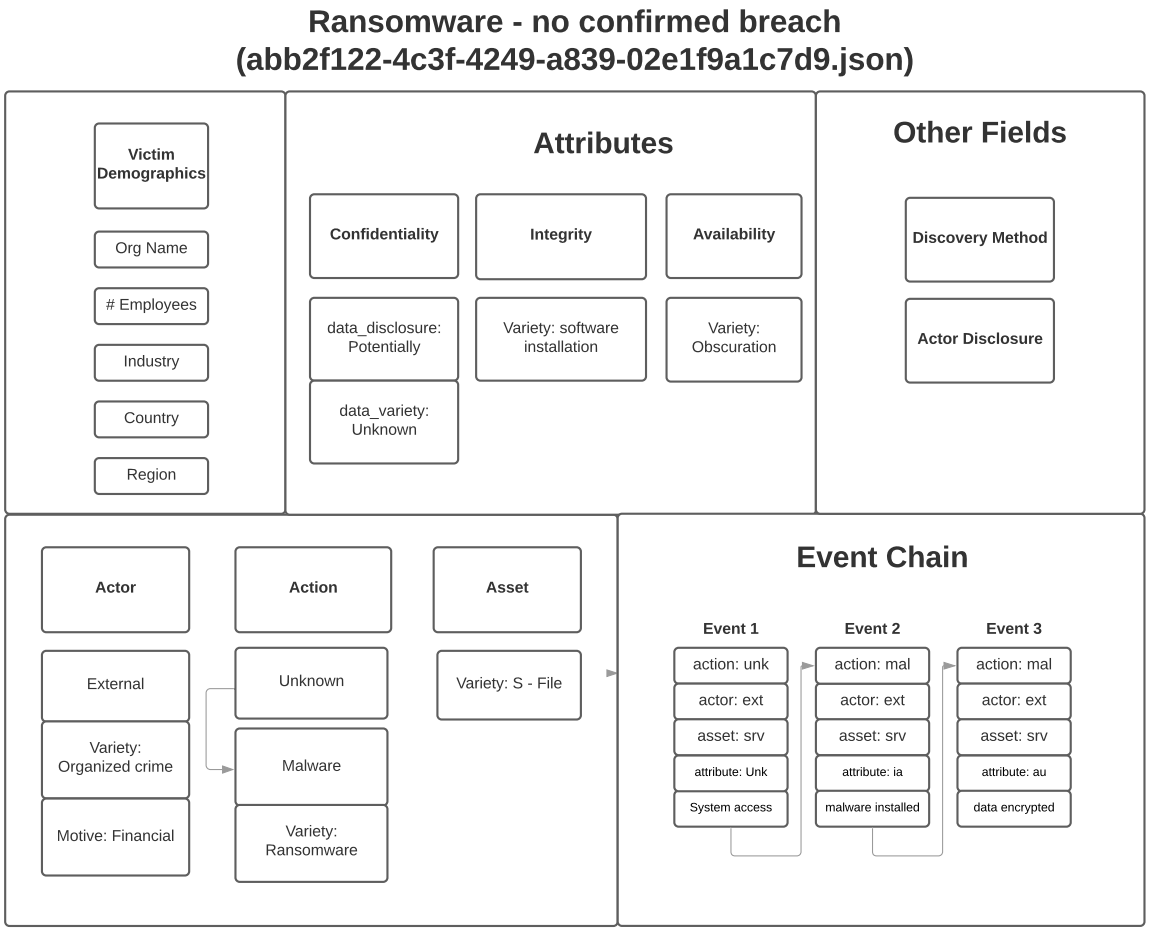
Ransomware

For Ransomware cases, we have a decision to make: was there a data breach or just the malware infection? We have split the two case types out to show how both should be handled. In general, if there is no indication of a confirmed compromise of the confidentiality aspect of the data, then the case should be coded as not being a data breach, but just a security incident.

Ransomware with no confirmed breach

Ransomware has been a popular money maker for bad guys on the internet. There is frequently a social action that starts it off, but frequently, we can't confirm that. The malware has to make it onto the system somehow though. So to account for that in our event chain and in the actions, we code an unknown event to represent the malware getting onto the systems.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/96159000ba2caf932ef1ca85d87888ce16bab979/data/json/validated/abb2f122-4c3f-4249-a839-02e1f9a1c7d9.json



The actor got the malware on the system, but what else did they do while they were there? Did they view any of the data on the system? Maybe. So we list data_disclosure as Potentially to account for this possibility. In the United States, the department of Health and Human Services (HHS) has given guidance that organizations experiencing a ransomware event should treat the case as though confidentiality has been compromised, and report it as such. So in Healthcare organizations, you see a high instance of ransomware cases being reported as breaches.

Ransomware cases are coded as financially motivated, since they're asking for money to decrypt the data. The external actors are coded as organized crime, as they have a process and are organized in the way they follow it.

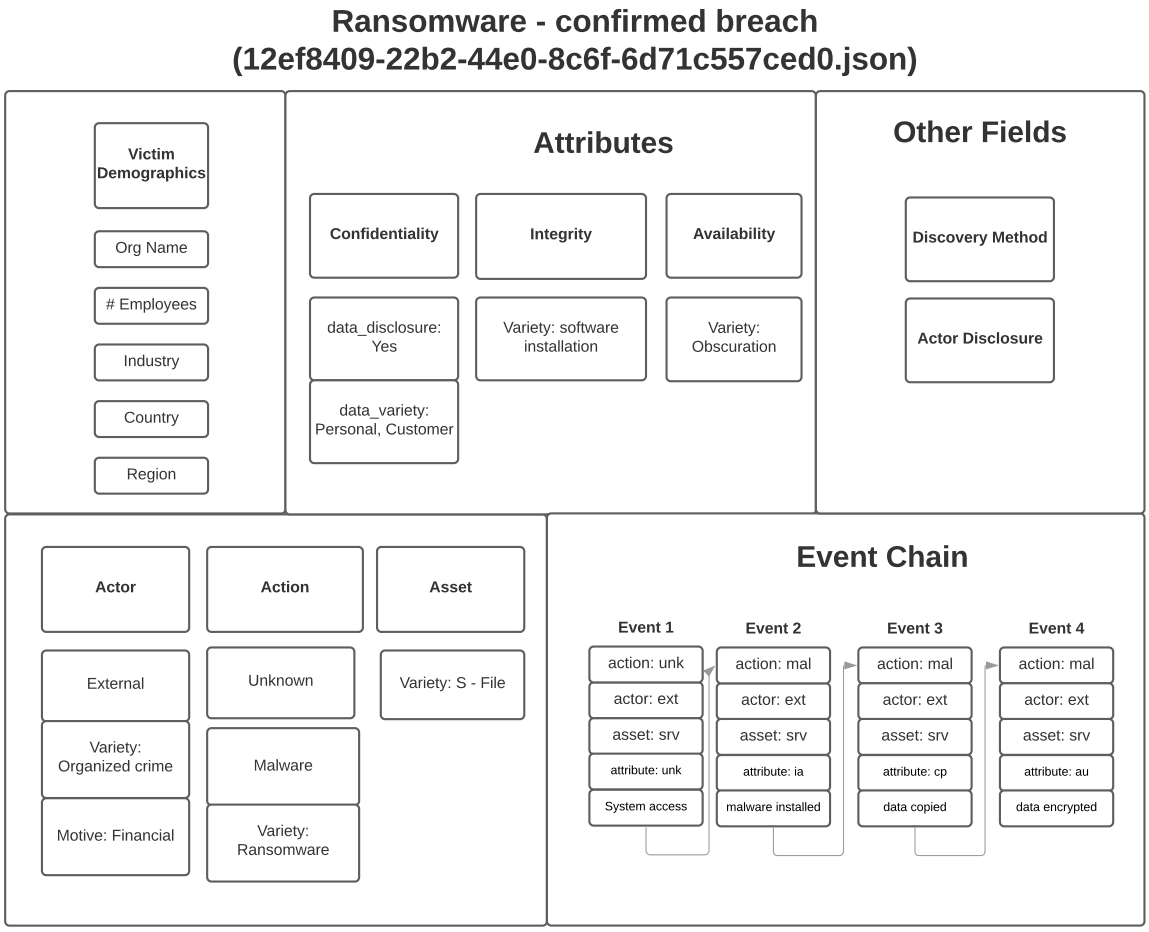
As mentioned above, we have an unknown action, and a malware action to account for how the malware got on the system. The variety in this case is a file server.

So looking at the event chain, you can see our unknown action of gaining system access, followed by the software installation (integrity violation) and then the malware is triggered, and the data is encrypted (availability violation). In this way, we attribute an event to each attribute that is compromised.

Ransomware with data breach

The diagram above shows the main coding style for this type of case. There are the typical victim demographics fields, which will vary case to case, so they are not filled in with values. Next, we have the CIA triad for the Attributes. For Confidentiality, we show data_disclosure as Potentially, as the data is at-risk, but we do not have confirmation of the data being stolen. The data variety is unknown, unless the source designates the data type at risk. If so, then naturally fill in any details you have. There is an integrity violation of software installation for the installation of the malware. There is also an availability violation as the data is obscured when it is encrypted. If we know details on how the malware got onto the system, include that as well. While we may suspect that there was a social action, such as phishing, if the source does not specify it, we cannot put it in the record.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/12ef8409-22b2-44e0-8c6f-6d71c557ced0.json



For the Actor, these are external actors, with financial motivation. The action shows the unknown action that led to the malware installation. The variety in this case was also a file server.

Note the event chain is more complex in this case. First we have our unknown action that led to system access. Could it be phishing? Could it be hacking? Yes, it could be lots of things, but since we don't know for certain what it was, this unknown action is our placeholder. This is followed by the installation of the ransomware, there is a step where the data is copied off, and then the actor triggers the encryption process.

This is an attack technique that has only been done starting in November of 2019. It was initiated by the Maze group, who were already well established in the ransomware game. Their change of tactic to take a copy of the data prior to encryption and use the threat of disclosure as leverage to get their victims to pay was so successful that

subsequent groups started to follow suit. Now here we are with this being a common place event.

Mining Cryptocurrency

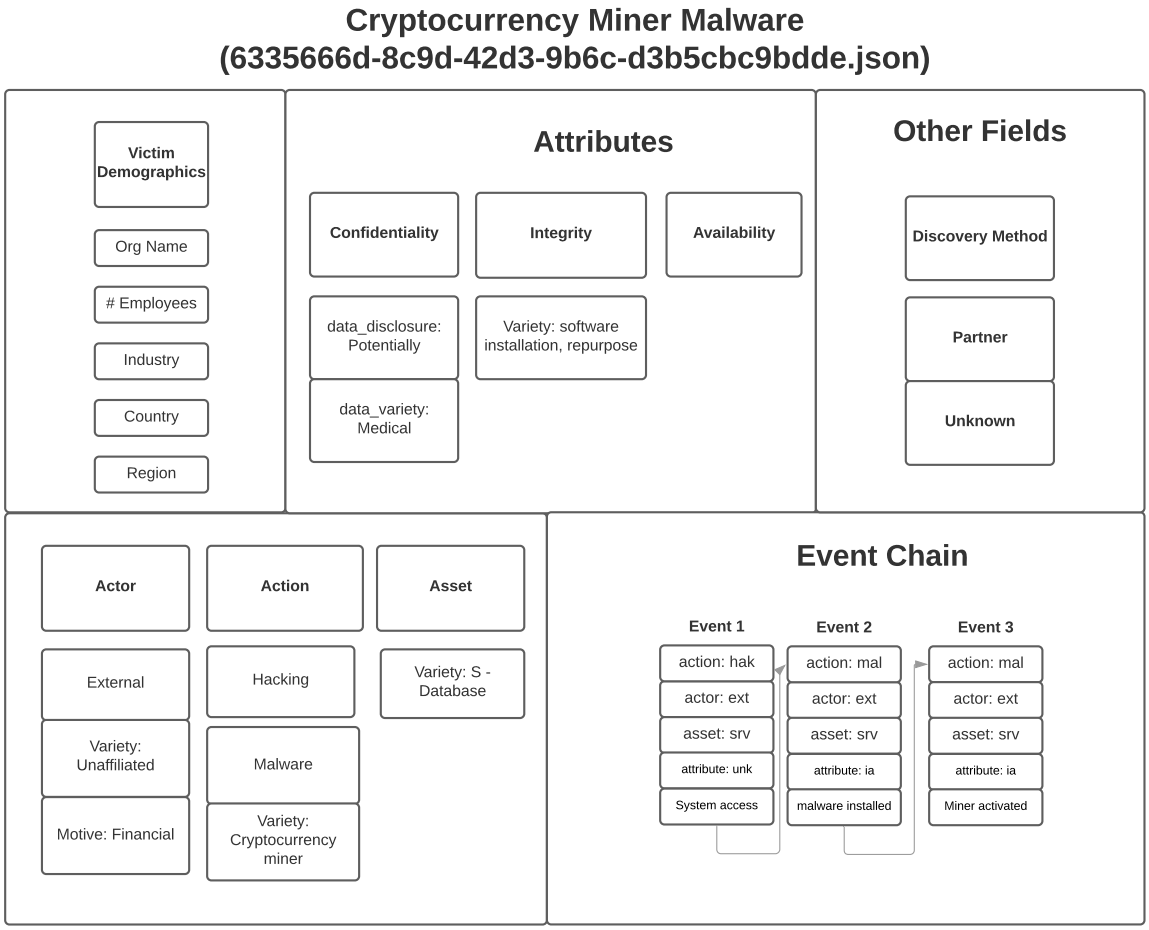
While ransomware is getting all the headlines these days, there are other kinds of malware encountered by those investigating security incidents and breaches. One such example is the cryptocurrency miner. In this case, a healthcare company's Electronic Health Records (EHR) system was hacked and this type of malware was deposited by the actor in the hopes of going unnoticed.

> "On November 27, 2017, we received a security incident report from our EMR system vendor indicating that unauthorized software had been installed on the server the vendor supports on our behalf. The unauthorized software was installed to generate digital currency, more commonly known as "cryptocurrency."
>
> Although the hospital's investigation is ongoing, they believe that an unauthorized individual accessed the server housing the EMR system to inject the software. The goal of the attack did not appear to be the acquisition or exfiltration of patients' personally identifiable information or protected health information, and the hospital has no evidence that PII or PHI was acquired or viewed. But as is the case so often, they could not definitively prove that there was no access or viewing, and so, they must notify patients.
>
> Information contained on the affected server included demographic

information such as patient names, addresses, dates of birth, and Social Security numbers, clinical information such as diagnosis and treatment information, and other information such as insurance billing information."

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/6335666d-8c9d-42d3-9b6c-d3b5cbc9bdde

s



We coded this one as potentially at-risk for confidentiality breach, based on the article indicating they couldn't be sure the data wasn't taken. The variety is of course medical records, given the nature of an EHR. While personal information such as Social Security numbers were also mentioned, we consider the main data type here as Medical.

The Integrity section is interesting, first we have the expected software installation of the malware. But we have another selection of repurpose—we use this when the intended use of the system is subverted for the cause of the bad actor. In this case, they repurposed the database to mine their cryptocurrency. We have also seen spammers start using email servers as their personal spamming systems. The repurpose selection allows us to account for this re-use.

There was no availability loss, although some may argue that there is system performance degradation when the miner is running. Since the article didn't mention anything about that, we will not make that assumption.

The articles indicate that the EHR vendor, who manages the system for the hospital, discovered the attack. Unfortunately, they didn't indicate how they found out past that, so we mark it as Partner, Unknown.

The Actor is external, with an obvious financial motivation. There was no indication they were in any way affiliated with the hospital. The Action is first an unknown hacking action that got them access to the EHR system, and then the installation and activation of the malware. The Asset targeted, as we mentioned before, was their records database.

The event chain is the three events—one unknown hacking, one malware installation and then the malware activation to account for the repurposing of the asset to the actor's benefit.

This kind of case doesn't vary all that much from a ransomware case, but we wanted to show you a different kind of malware attack to give you that example.

Misuse Examples

Single Actor Misuse

A trusted internal actor can wreak serious havoc on an organization's network and systems by misusing the access that has been granted to them just to get their job done. For most of these cases, it is just one person who, for whatever reason, has gone rogue and starts pulling down data for nefarious purposes.

Our example here is a bit unusual, in that is wasn't so much a case of an employee who stole data to take to another company, or someone who wanted to otherwise monetize their access. In this case, it was used for political gain.

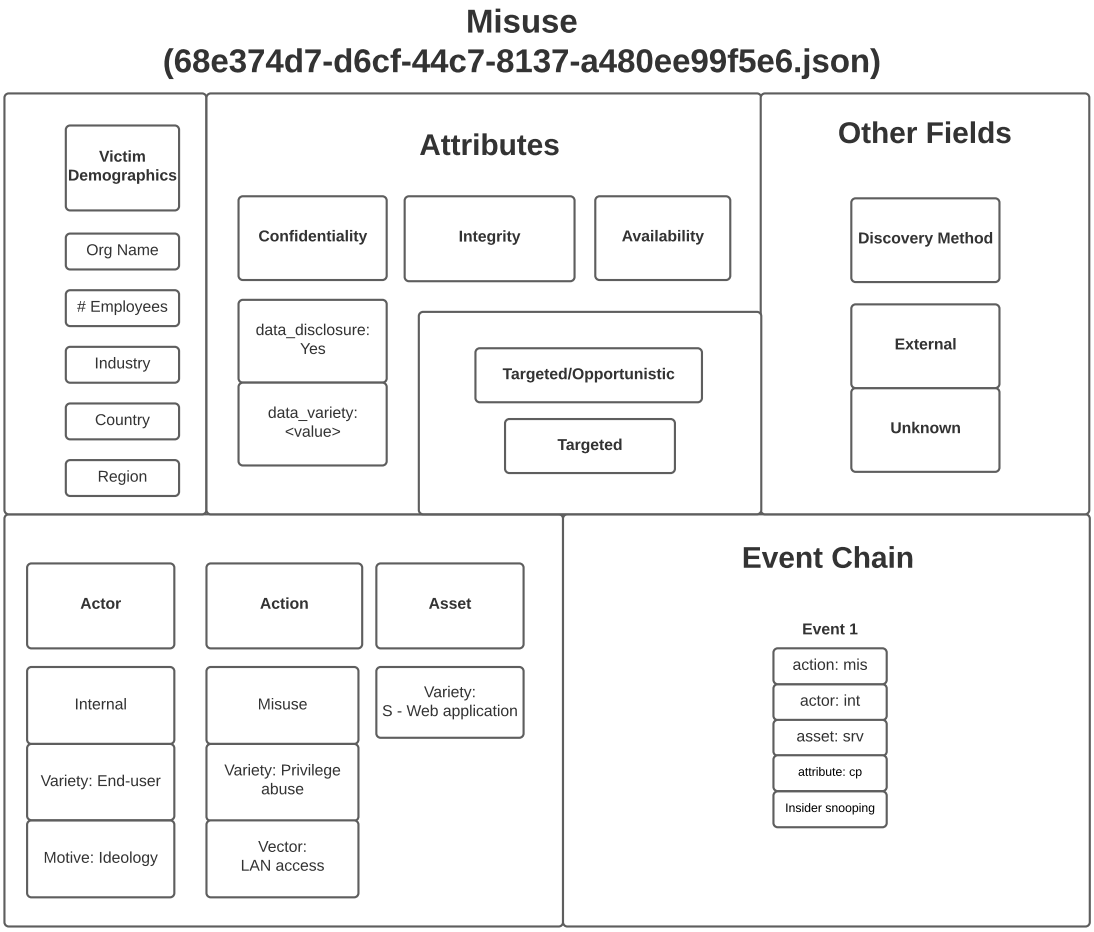
> "A former VA employee just pled guilty for computer fraud in an illegal scheme to access the medical records of a veteran running for Congress in an alleged smear campaign."

The employee was motivated to take pictures of medical records and share them with the someone who could use them against a political opponent.

> "One of the victims in question, Richard Ojeda (pictured above), is a retired Army Major who served three tours to Iraq and Afghanistan. At the time of the computer fraud, Ojeda was running for Congress while serving as a state lawmaker in West Virginia. A former VA employee illegally accessed the veteran's medical records, took photos of them, and shared those with a third party during the campaign."

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/

data/json/validated/68e374d7-d6cf-44c7-8137-a480ee99f5e6.json



So while most of this case is fairly straightforward in terms of coding, confidentiality was violated, the actor was internal, the action was privilege abuse, we have one thing here that we don't often talk about—whether the attack was targeted or opportunistic. This is something that we have debates on our team on a regular basis, because the line between them can be hard to determine. The attack here was clearly targeting one specific patient. The actor used their access not against hundreds of records, but against just this one. So we mark this as a targeted attack.

Much more frequently, these insider attacks are a matter of opportunity. They have all this access to data, and they use it to then steal the data. They didn't get

the job with the goal of gaining access to the data, although we have seen that on rare occasions. But because they used their access against that one specific target, the deliberate intent is sufficient to call this a targeted attack.

Multiple actors can also be involved in Misuse breaches, and for that, we have a case in the Coding Multiple Actors/Actions section.

Physical Examples

Tampering

Physical attacks are what they sound like—someone had hands on the hardware. Whether it is affixing a skimmer to a gas pump or ATM, or outright stealing an asset, nothing trumps having physical access to a device.

We will cover these two physical actions, as they are our most common Physical action types. The full list is as follows:

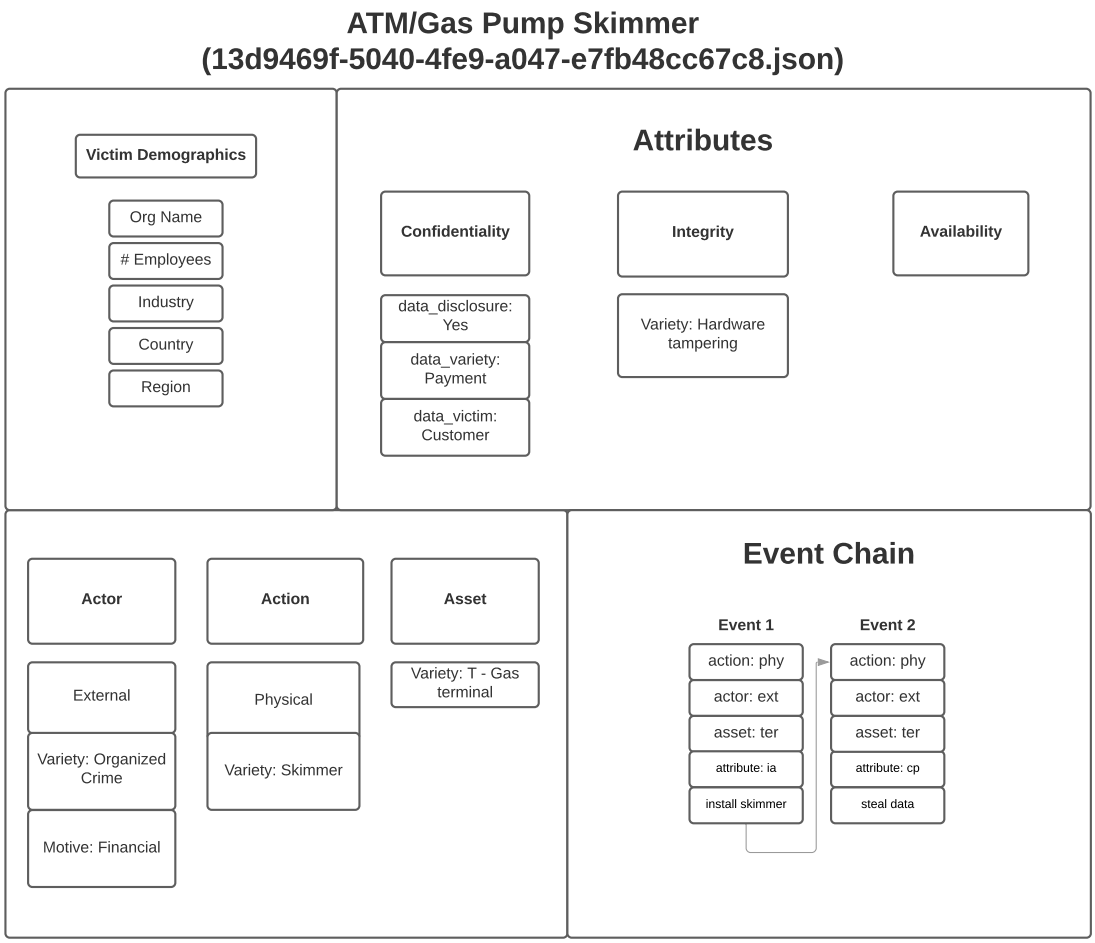
Assault
Sabotage
Snooping
Surveillance
Tampering
Theft
Wiretapping
Other
Unknown

We rarely see the other varieties in this Action category, but if you encounter them, you're ready to record them in VERIS.

ATM Skimming (Tampering)

This is the case diagram we looked at in the beginning of this document, so it should look familiar.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/13d9469f-5040-4fe9-a047-e7fb48cc67c8.json



One common mistake when coding this kind of case (whether it is an ATM, a gas pump or a point of sale device) is to forget to include the integrity violation. When actors affix hardware onto these systems, you are seeing hardware tampering. So in addition to the confidentiality compromise of the data, you need to account for the fact that the hardware has been changed by the actor.

The actor is most commonly external in these cases, usually organized crime.

A word on that—we do not mean the mafia when we say organized crime. We mean this is a criminal with a method they reuse time and again on multiple victims. They have a process and they work it. They are always financially motivated. The action is a physical action with a variety of skimmer. The asset is a gas terminal.

The event chain is two-step to account for both the attributes mentioned. First is the integrity violation of attaching the skimmer to the ATM. The second event is when the data is siphoned off.

Theft

People steal stuff, including stuff with data on it. Laptops are a favorite target, and there are still a number of organizations that haven't gotten the "encrypt all the portable things" memo. This case is one of them.

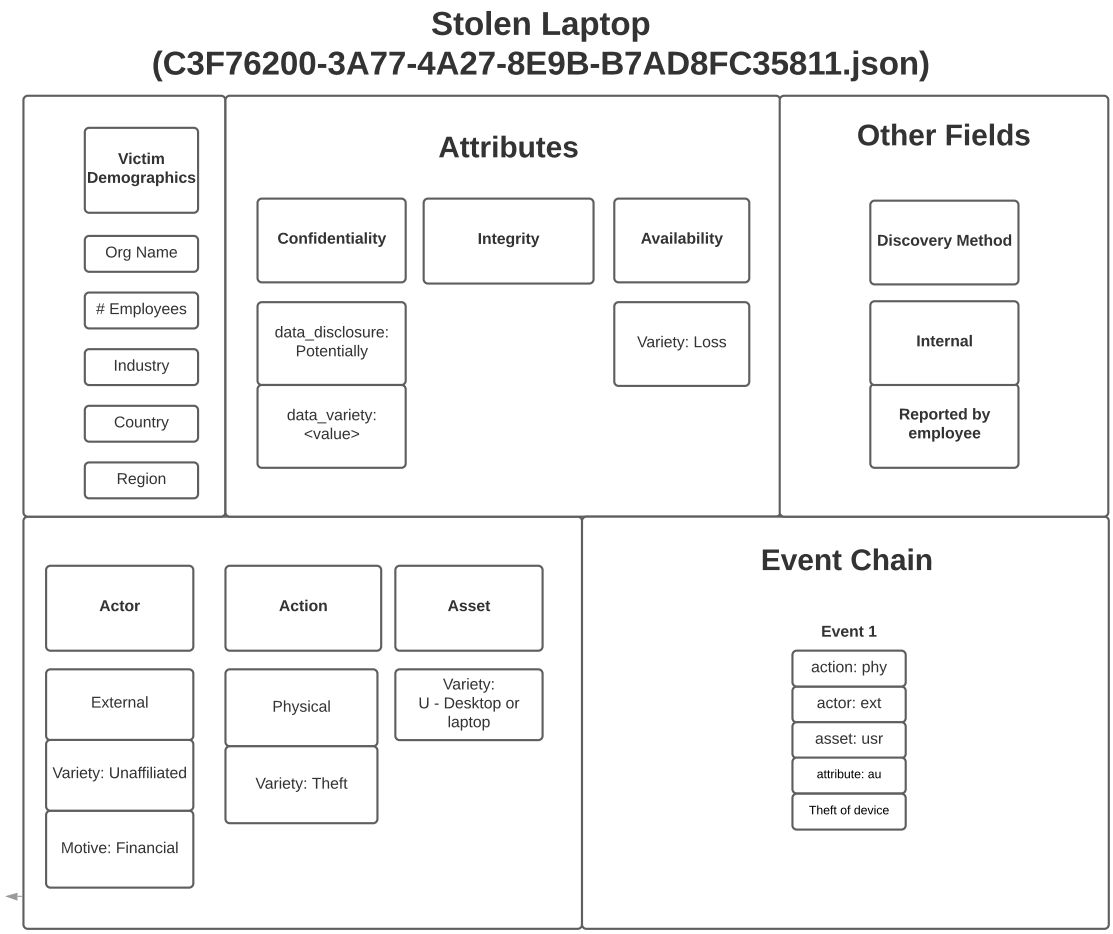
> "Dr. Olartino Dyoco sent data breach notification letters to patients after certain information was potentially exposed following an office break-in. Dyoco noticed on June 2, 2015, that his office had been burglarized, according to a copy of the breach notification letter. Several computers were stolen, Dyoco reported, containing information such as patient names, dates of birth, telephone numbers, insurance numbers, treatment codes, and billing information."

Sadly, this entity got the encryption memo only after the risk was brought home by their office break-in.

> ""The circumstances that resulted in this breach were unforeseeable, and

Dr. Dyoco assures you that he has heightened procedures and safeguards to prevent a recurrence of this situation," stated the letter, which was dated July 13, 2015. "He added levels of encryption to his computer systems, and advised his staff with regard to security training anything to avoid this situation in the future.""

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/C3F76200-3A77-4A27-8E9B-B7AD8FC35811.json



You may notice that we marked this case as data_disclosure "Potentially" rather than the definitive "Yes". The stolen laptop is the classic case we use to

illustrate the difference between a security incident and a confirmed data breach in the DBIR. To be a confirmed breach (and have data_disclosure marked "Yes"), there must be a confirmed compromise of the confidentiality aspect of the data. Since the victim organization no longer has custody of the laptop, that confirmation is very difficult to make for certain. The asset is protected with just a password, which leaves it in the "at-risk" category. Thus, this would not be a confirmed breach in our report.

The Availability violation is Loss, because the organization no longer has access to the asset. Sometimes the notices indicate the asset is recovered after a period of time, and there is a duration field where you can record this value. Most of the time, stolen devices are not reported as recovered, though.

The Discovery method for this kind of event is the employee returning to the car (or office, etc.) and discovering the break-in. In the webapp, we actually have a drop down value of "break-in discovered" to make it easy. This tends to be a rather quickly discovered type of incident, as evidence of broken windows and forced doors tend to be noticed rapidly.

The Actor is an external, unaffiliated person with a financial motive (these cases usually see the device resold, and the data is rarely the target, but the motive doesn't matter when you are required to report a breach). The Action is Physical with a variety of Theft. The Asset is the laptop or desktop(s) that were stolen. Since the sources do not specify whether these are desktops or laptops, we choose the parent "U - Desktop or laptop" as it is a parent of both "U - desktop" and "U - laptop". We have such hierarchical fields in VERIS for just this reason.

The event chain is a one-step diagram. It didn't take many steps for the person who stole it to get the asset(s) and make off with them.

Social Examples

Social Overview

Social actions, where the human is the target, have become so prevalent that there seems to be no escaping them. While phishing is the most prevalent, there are a more options for the enterprising social engineering actor to choose from. Here they are:

Baiting
Bribery
Elicitation
Extortion
Forgery
Influence
Scam
Phishing
Pretexting
Propaganda
Spam
Unknown
Other

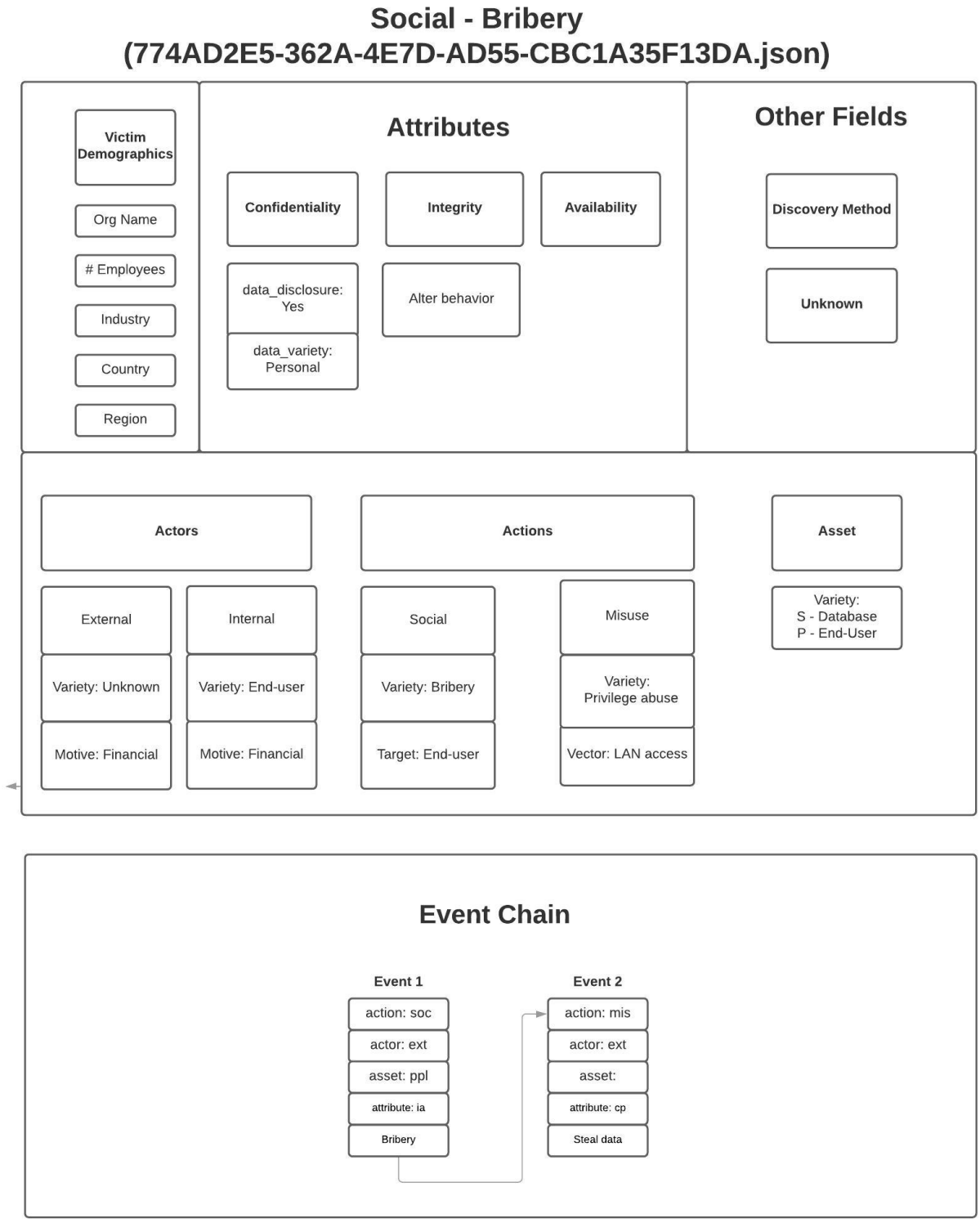
We will of course show a phishing example, but first we will visit Bribery, and what happens when someone outside your organization motivates someone inside it to do bad things.

Bribery

Far and away the most common social action in our dataset is phishing. However, there are other social actions we do see from time to time. Bribery is one of them, and here we have a case with internal and external actors, and a social action.

> "Two people have been charged with bribery offences following an investigation into the suspected leak of confidential data by a former employee of LV to a claims management company. Aisha Elliot, aged 23 from Yeovil was charged with offering a bribe and Stephen Karl Oates, aged 26 of Bournemouth was charged with receiving a bribe. Oates was an employee working with LV at the time of the alleged offence and Elliott worked for a claims management company."

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/774AD2E5-362A-4E7D-AD55-CBC1A35F13DA.json



This was an interesting case of collusion between two kinds of actors who worked together to steal data. Bribery is a social action that we don't see in data

breaches very often, but it does happen.

The data disclosure was confirmed, and the data variety was personal information. The social action comes with an integrity violation of alter behavior. We usually see this with phishing attacks, but really any kind of social engineering action that causes a person to do something different is a candidate for this variety.

We don't know how the breach was discovered, unfortunately, and that would have been quite helpful, given how few of these collusion cases we see come to light.

We have two Actors as mentioned above. The external actor and internal both have a financial motive. We don't know much about either of them, so the internal actor is just marked as an end-user. There are two Actions as well—the social action of the external actor offering a bribe to the employee of LV, and the subsequent misuse of their access (privilege abuse) by the employee to steal data and hand it over to the external actor.

The Assets were both the Person in terms of the internal actor who accepted the bribe, and the database they stole the customer information from.

The event chain is just those two actions of the social - bribery and the privilege abuse to get the data.

Phishing

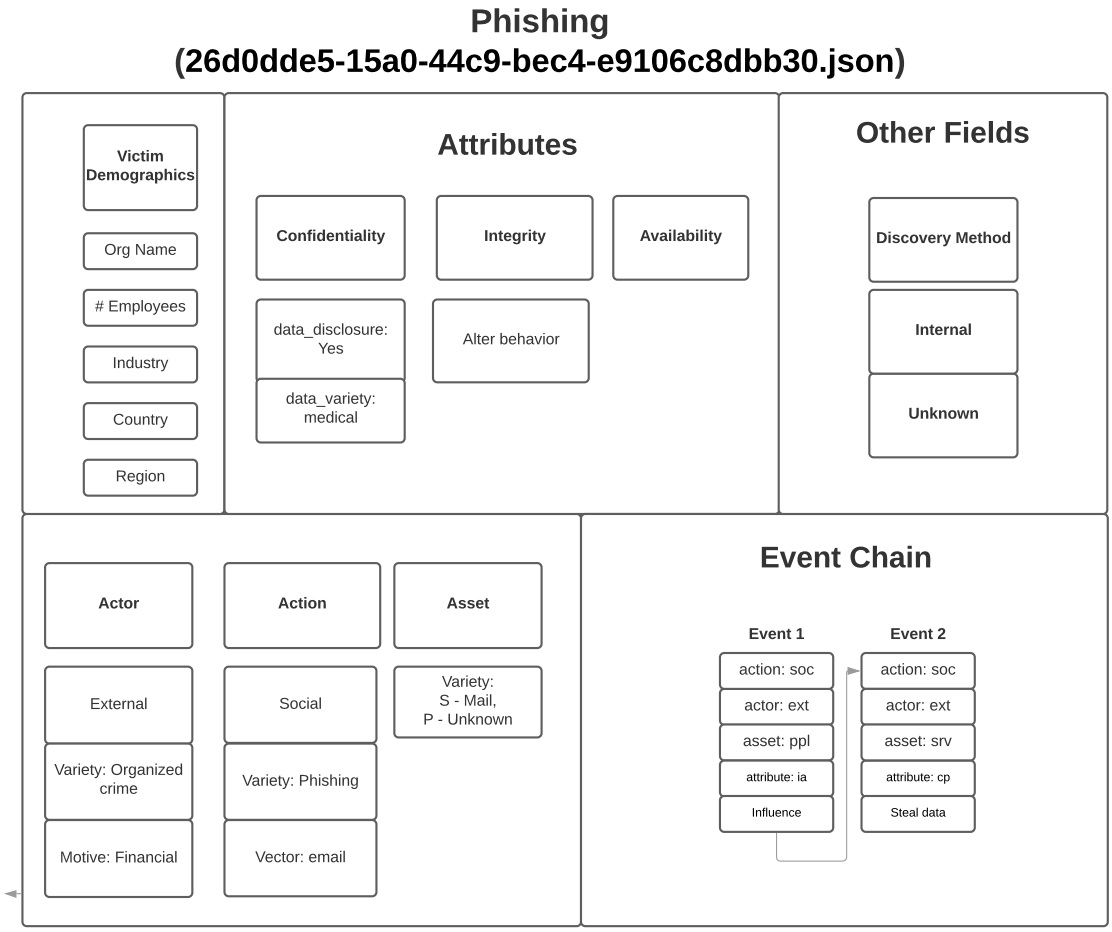
The most common social action by far is phishing in our dataset. The attack is against a Person asset, which may sound strange, but we wanted to include the people whose behavior is successfully altered by the attacker in making them do something they aren't supposed to do—whether it is giving up their credentials or paying an invoice to a thoughtfully provided new bank account.

"The University of North Carolina at Chapel Hill School of Medicine today

announced it is mailing notification letters to an estimated 3,716 persons whose information may have been affected in a cyber phishing incident involving some School of Medicine email accounts.

A leading independent forensic firm conducted a lengthy and extensive review that concluded on Sept. 13, 2019, and confirmed that an unauthorized third party gained access to several email accounts during the approximate timeframe of May 17, 2018, to June 18, 2018. This review confirmed that some patients' personal information was contained in the affected email accounts, possibly related to treatments received when they were seen by a UNC physician."

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/26d0dde5-15a0-44c9-bec4-e9106c8dbb30.json



The confidentiality was of course compromised, and the data type was medical. The integrity violation of the person falling for the phish is accounted for as Alter Behavior. The actor is external, with a variety of Organized crime, meaning the person attacking is organized and has a process they follow, not that we're talking about the mafia. Their motive was financial.

The assets listed are the mail server and the person who fell for the phish. Our event chain has the social action of influencing the person to alter their behavior, and then absconding with the data that was in the email accounts.

In some cases, we see attackers also using this kind of access to change the rules on the email boxes to keep their activities from showing the victim what

they are doing. They may also turn around and send out further phishing or pretext emails from the compromised accounts. We show how to handle those in the other examples in this guide.

Pretexting

The Business Email Compromise is a lucrative social engineering attack that has increased in frequency in recent years. It is usually based on a pretext social action—when someone invents a plausible story to get the target to take the bait.

> "In October 2017, someone infiltrated the email of the president of Buresh Building Systems and used the account to direct an employee to wire money to an account of K Henao Elite Inc. at SunTrust Bank in Florida, according to court records.
>
> Company officials later determined the president's email account had been accessed, and the perpetrators had set up systems within the account to hide the fraud-related messages when the president was using the account."

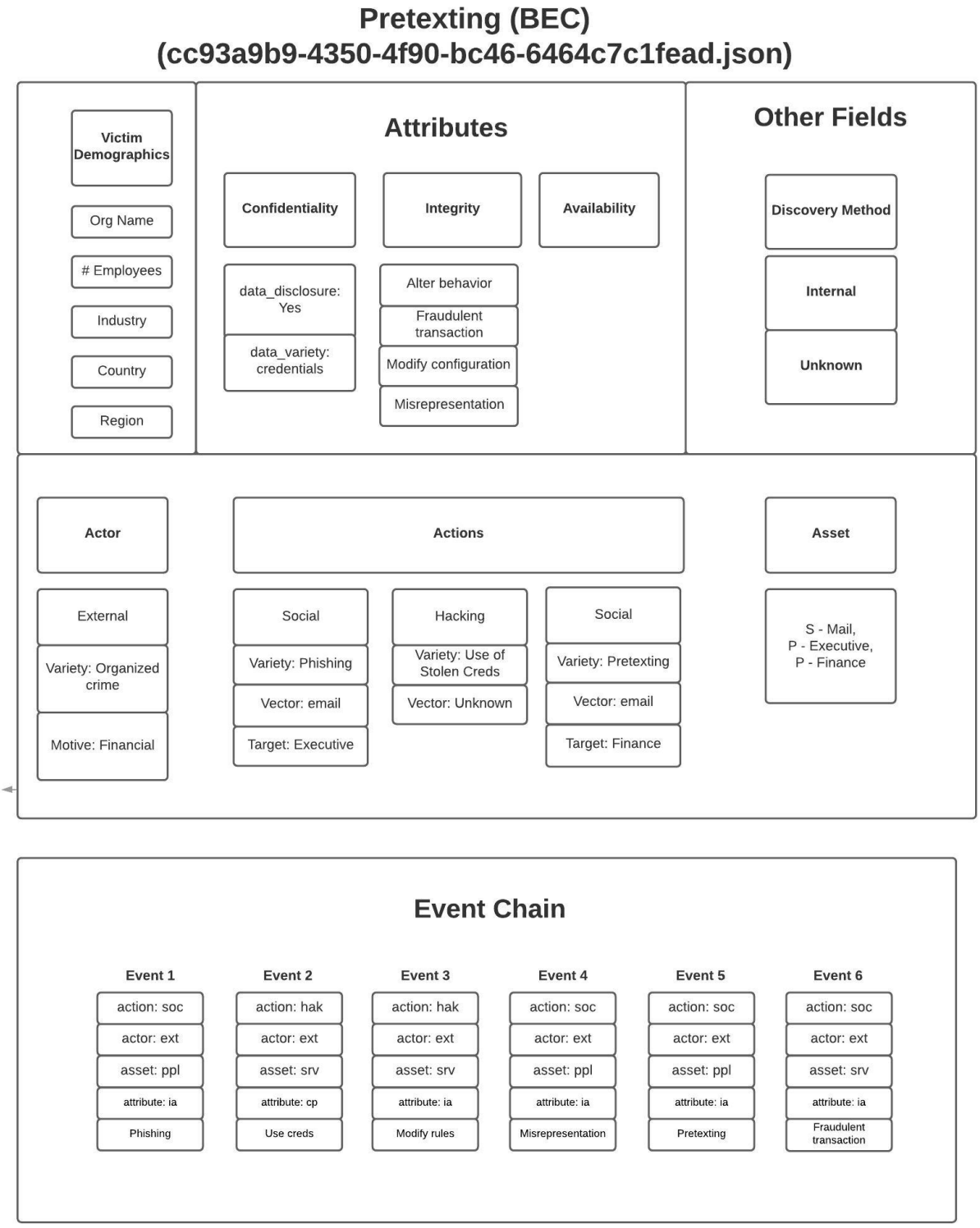
This turns out to be a fairly complex case. First we have the standard demographic info that we try to collect from all cases. We have a confidentiality violation with the stolen credentials from the executive from the first phishing.

> "The organization allegedly gained access to email accounts of business leaders through a "DocuSign" rouse — emails told users to sign into their accounts and reset their passwords. Then members of the group imitated the business officials, asking workers to wire money to bank accounts they

controlled and then sending the proceeds overseas."

So they got the executive's credentials via their DocuSign phish, and used them to get into the email server—that is our confidentiality violation. They then made changes to the email account's rules to keep the executive from knowing their email account was being used to send further social attacks. They misrepresented themselves as the legitimate owner of the account, and they used a pretext to get the employee in Finance to make a fraudulent transaction, sending them money to the account they controlled. So we have three integrity violations once they got the credentials.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/cc93a9b9-4350-4f90-bc46-6464c7c1fead.json.



We have a fair amount of information about the threat actor in this case, which is somewhat rare. Usually we hear more about what happened in the

course of the breach, but information about the perpetrator falls into the attribution problem. Here, since the case was being prosecuted, we have better data.

> "Henao is a native of Colombia who has been living in Port Saint Lucie, Florida, as a permanent resident. Authorities allege she was working for a criminal organization that operated in the fall of 2017 to defraud people throughout the United States."

While we aren't showing all the details in the diagram in the interest of space for an already complex case, we can record all of it in the webapp, so the json will reflect it. Here is an excerpt from the json:

```
"actor": {
   "external": {
     "country": [
       "CO"
     ],
     "motive": [
       "Financial"
     ],
     "notes": "Karina Henao",
     "region": [
       "019005"
     ],
     "variety": [
       "Organized crime"
     ]
   }
```

For the actions, there are basically three. First we have the original DocuSign phish, which is a social phishing attack targeting their CEO. Then we have the Hacking, Use of Stolen Credentials against their email infrastructure. Finally, the attacker pivots and sends out pretext emails masquerading as the CEO to have the money wire transferred.

Our assets follow suit, with the email server as the technical target, and the two people who were social engineering victims.

Finally, we go to our event chain—yes, three actions turned into six events. That takes into consideration the separate confidentiality and integrity violations, and splits them out individually. First we have the initial phish of the CEO. Then we see the stolen creds being used. Then they alter the email account's rules and misrepresent themselves as the CEO. Finally, they get that pretext attack that leads to the fraudulent money transaction.

Targeted vs Opportunistic

One of the things we track in security incidents is whether the attack was targeted or opportunistic in nature. This question is only relevant to deliberate malicious actions—errors are not applicable as they are accidental in nature. So the values associated with this question will include:

- NA (not applicable) - the action was accidental in nature (not malicious)
- Opportunistic - the victim was attacked because they exhibited a weakness the actor knew how to take advantage of.
- Targeted - the victim was chosen as a target and then the attacker

determined how they could successfully attack them.
- Unknown - the status of whether the attack was targeted or opportunistic is not known.

So realistically, this is a "chicken or egg" situation—did they pick the victim first and examine their weaknesses to determine how to attack them, or did they have an attack they knew how to exploit and found the victim was vulnerable to that attack. Did the choice of victim come first or did the choice of attack?

We see a lot of cases where someone has an attack they know how to use, and uses the internet to expand their ability to reach appropriate victims. These would be considered opportunistic attacks, given that the attack type was chosen first. If the attack is targeted, there is usually a reason the victim was chosen that helps play into the motive of the attack. In the Misuse example, we have a case where the victim was targeted because they were in politics and the actor wanted to discredit them by releasing their medical records.

It can be hard to determine which came first, and so we have a large number of unknowns in the data for this, but when we do have it, it is a good thing to record.

Coding Multiple Actors/Actions

Multiple Actors Overview

When different types of actors get together, it never means good news for the victim organization. As with multiple actions, VERIS allows for multiple actors to be active in a breach. VERIS has three actor types: Internal, External and Partner. When we see multiple actor types, there is sometimes a social component where the external actor recruited an existing employee, or even sent someone into an organization with the goal of getting hired and having access to data. Partner actors may also be either the recruiter or recruited.

Multiple Actor Misuse - When Collusion Comes Calling

This case has an external actor recruiting internal actors at two telecommunications companies (yes, ours was one) to use their access to gain customer data, which they then put to use for fraud.

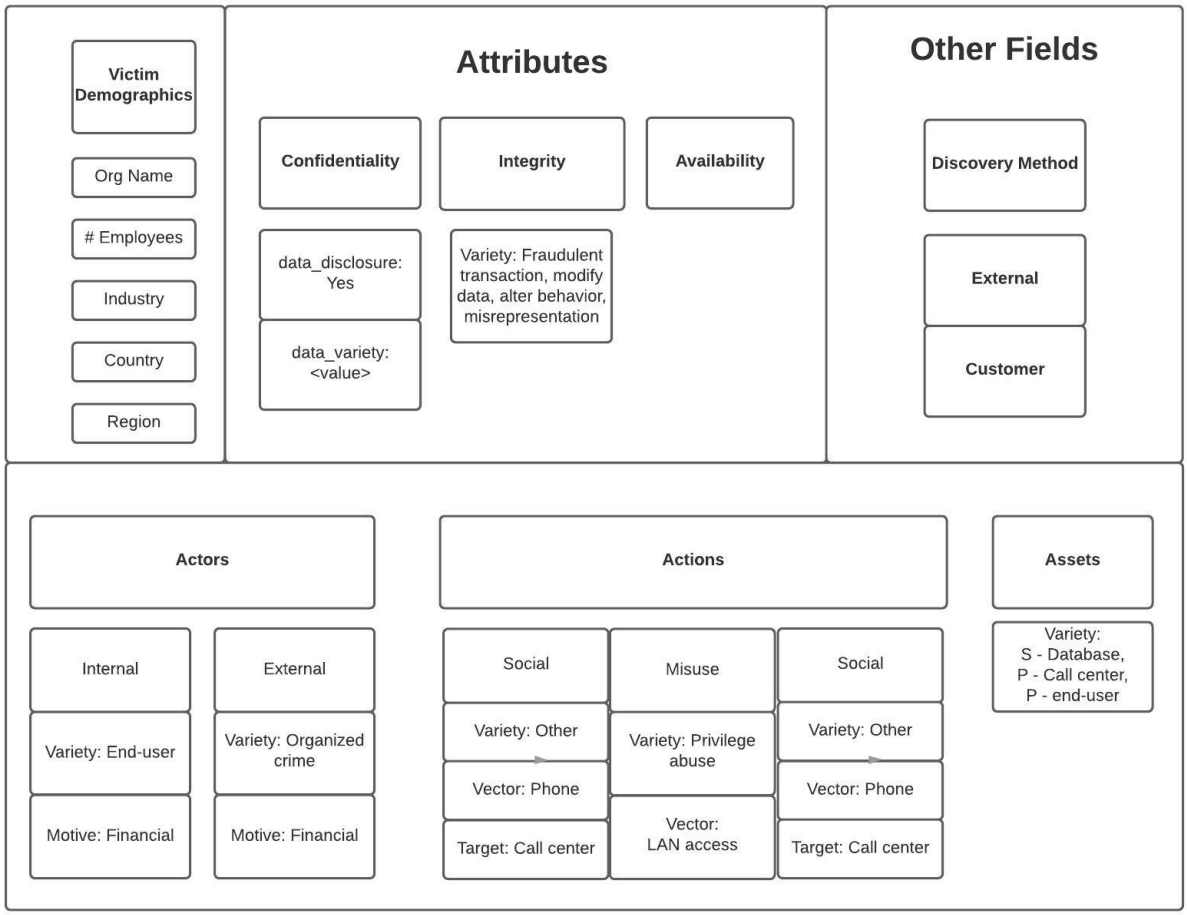
> "Internal actors at Verizon & AT&T collaborated with external actors to misuse privileges and cause loss of confidentiality of 860 customer PII records along with the external parties being added to the customer accounts and purchase mobile phones."

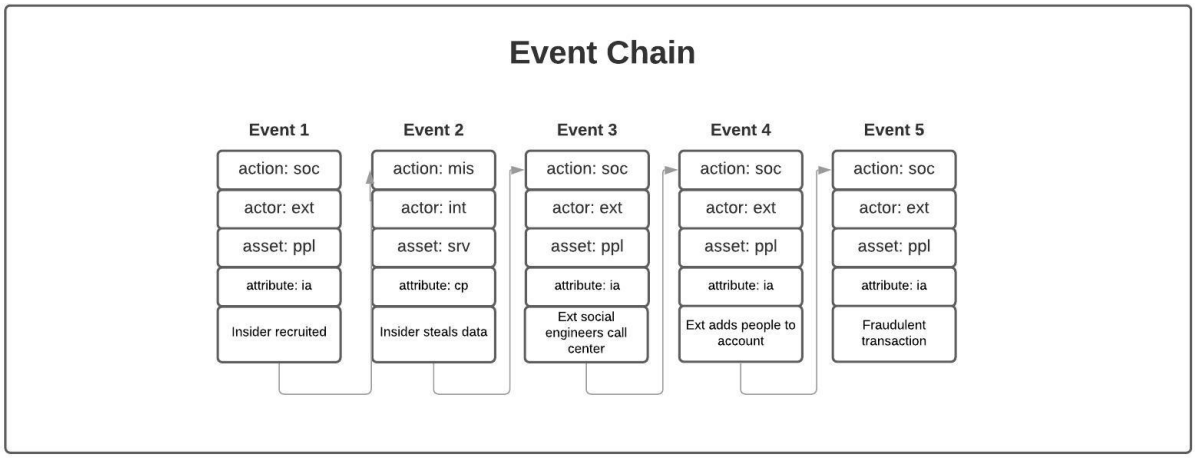
Given that this is an attack against two different companies, and involving internal actors at two different companies, it would be coded as separate incidents in the data. If you think about it, they had to compromise the internal actors at both organizations, and then separately call their respective Customer

Service departments and do their impersonation of the customer multiple times, so it makes sense that it would be separate breach events.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/AD2CC5B5-D6B4-4B62-A5E4-8B6ED36677C2.json

Multiple Actor Misuse (AD2CC5B5-D6B4-4B62-A5E4-8B6ED36677C2.json)





Social actions include bribery and phishing, but also less pleasant methods of gaining the assistance of reluctant participants—extortion and blackmail are also methods at the disposal of the unscrupulous person trying to get their way. They are frequently significant others or relatives of the employee, so that kind of relationship pressure can also be brought to bear.

For this case, the Attributes of Confidentiality and Integrity are both violated. Confidentiality of customer's personal information is stolen. The integrity violations begin with Alter behavior (to get them to collude with the outside actor), followed by misrepresentation when the external actors pretended to be the customer calling into Customer Service. Then we have modify data, as they added people to the customer's accounts, and finally we have the fraudulent transactions as they order phones.

The Discovery method is the customer, who later got a bill with the phone purchases against their account and reported the problem.

As indicated, we have an external and an internal actor, both with a financial motive. We have three Actions, beginning with the social action that happened when the external actor recruited the internal actor to start the whole scheme in motion. The next action is the insider misusing their access to steal data. Then we have the external actor taking that information and contacting the Call Center and impersonating the customer.

The Assets include both the employee who was recruited to participate in the incident, and the Call Center employee who was duped. The final asset is the customer records database where the information was originally kept.

We have quite a complex event chain—six steps in all. The first event is the social action recruiting the employee of the company. Next we have that employee stealing data. Then we have the external actors social engineering Customer Service, pretending to be the customer. The perpetrators add extra

people to the customer's account, and then finally, they order new phones.

So two actors, three actions and six steps in the event chain. Despite a pretty complex case, we were able to break down the elements and record them using VERIS without too much difficulty.

Hacking & Malware

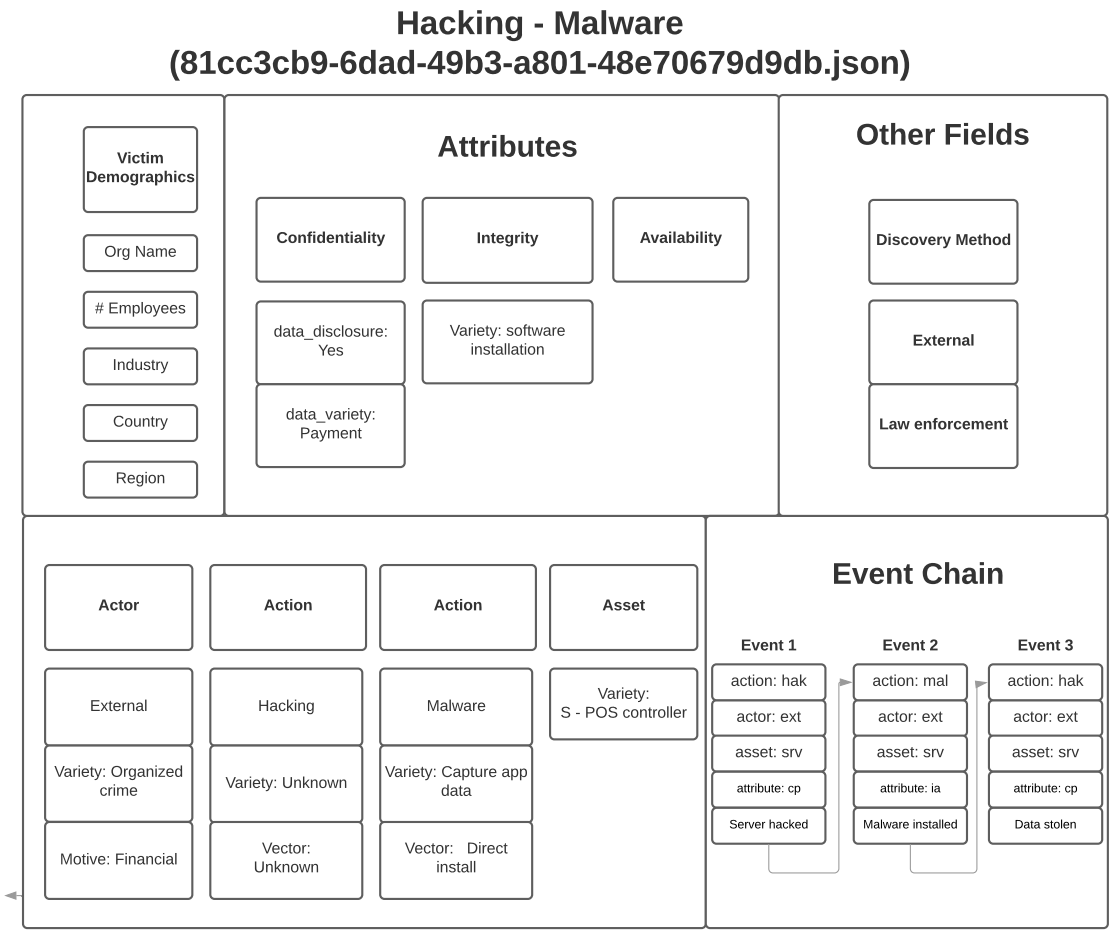
This example is a case where a financially motivated actor (Fin7) hacked into the point of sale server and installed malware to steal customer payment data. Here is the synopsis from one of the articles we cite in the record.

> "Burgerville has revealed a data breach impacting the chain which may have led to the theft of detailed credit card information belonging to customers. Malware was installed on Burgerville systems in order to scrape and steal customer data. Burgerville says that customer credit and debit card information, including names, card numbers, expiration dates, and CVV security numbers were stolen.The data breach has been attributed to Fin7, also known as Carbanak Group, an international hacking ring which has successfully launched cyberattacks against at least 100 US companies."

Confidentiality was confirmed compromised—they did in fact get an undisclosed number of customer credit/debit cards from this breach. Integrity was clearly violated, given the software installation of the malware.

Here is the link to the json file in the VCDB GitHub repository: https://

github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/81cc3cb9-6dad-49b3-a801-48e70679d9db.json



The breach was not discovered until Law Enforcement notified the victim organization. As is the case with externally discovered breaches, this one went on a long time—the attackers were in their systems a year before they were notified of the breach.

An external, financially motivated organized crime actor means they will have a process to get in and then monetize the data. We have an unknown variety of hacking. This was followed by installing the malware that was able to steal the data and send it along to the adversary. The asset is the Point of Sale

controller server.

The event chain is a bit more complex, with three events. The unknown hacking action is the first event. This is followed by the integrity violation of installing the malware, and then the confidentiality violation of stealing the payment card information.

Social - Hacking - Malware

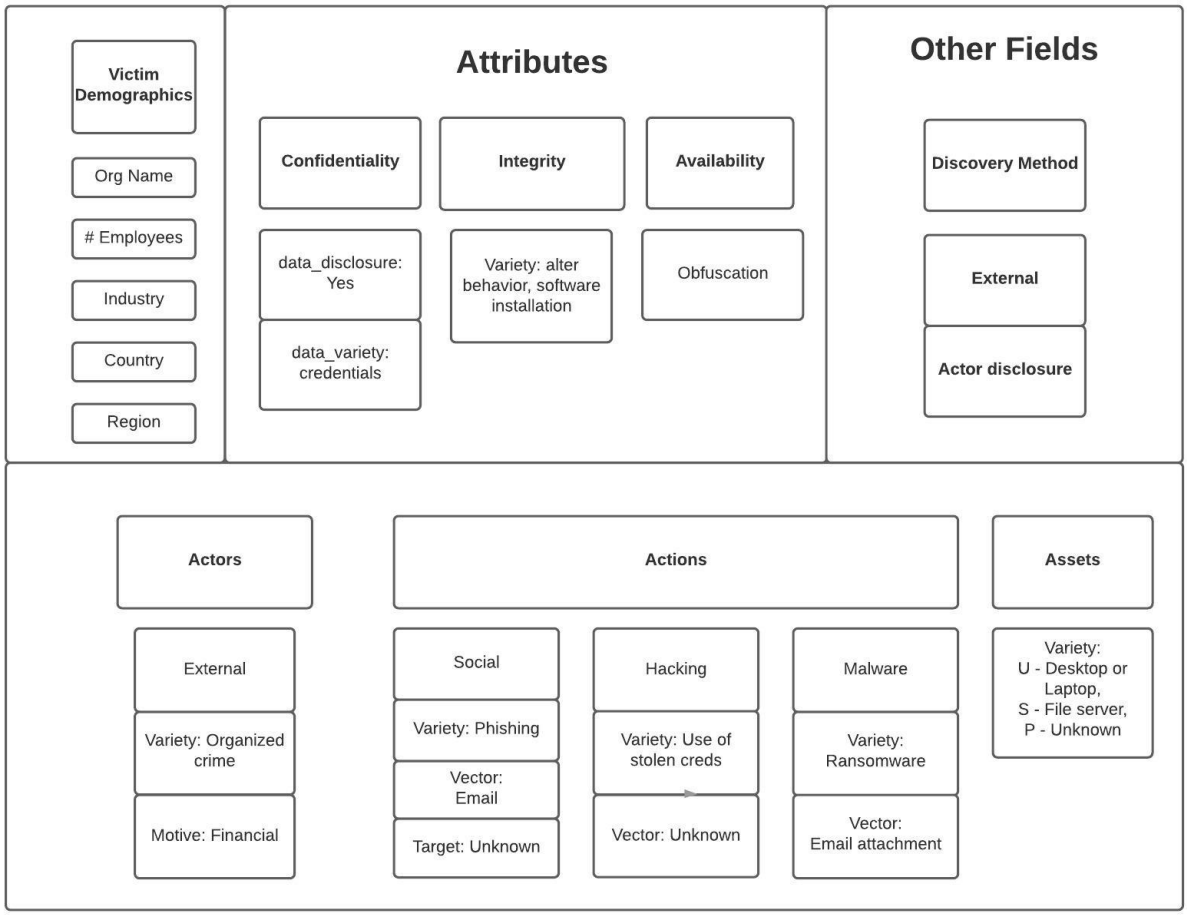
Sometimes the attackers pull out all the stops and use multiple actions in cases. This case shows how to code up a case that includes three different actions: social, hacking and malware for the trifecta of complex breaches.

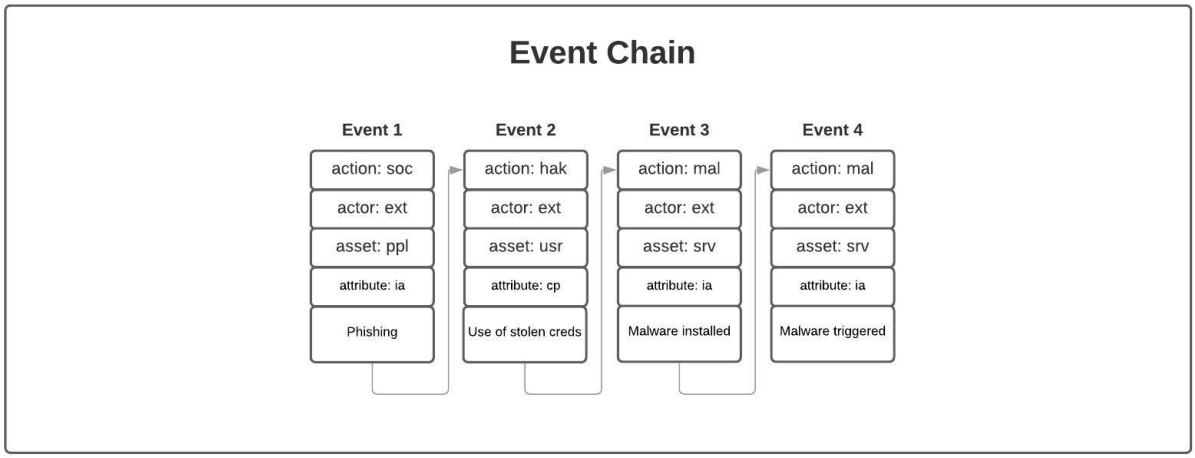
> "The ransomware malware attack came in as an attachment and cleaned our music libraries and ad spots that we had stored on hard-drives," Durham Radio sales VP Steve Macaulay said in a phone interview on the weekend.
>
> At no point were the station's computers hacked, just a harmless looking attachment from a known email address that turned out to be a trojan intruder that wiped drive sectors on the company's hard-drive. To prevent this occurring again, Durham now doubles up all of its information and secures it on a third-party cloud service. "

While it is nice to see a company able to recover quickly from this kind of attack, many organizations are not so well prepared.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/dc189ccd-8d88-4c86-a0f0-974dbbf80b6d.json

Social Hacking Malware
(dc189ccd-8d88-4c86-a0f0-974dbbf80b6d.json)





While the case looks pretty complex, coding it up is just a matter of going down the line and recording the values for all our boxes. Confidentiality was a confirmed breach, because they got the credentials. Integrity was both Alter behavior for the phish, and software installation for the ransomware. Availability took a hit as well, with obscuration once the encryption was triggered.

As with most ransomware cases, the discovery method is Actor disclosure, when the ransom note pops up on the screen.

The Actor was External, Organized crime with a Financial motive. The Actions include Social: Phishing, Hacking: Use of stolen credentials, and Malware: Ransomware with a vector of email attachment. The Assets involved were the Desktop or Laptop the initial phish came to and the associated person, then the fileserver, which the ransomware started encrypting almost immediately.

The event chain has four steps, accounting for the initial phish and subsequent use of the stolen credentials, and then the malware is dropped and encryption triggered.

So while this is a fairly complex case, involving multiple actions and assets, once you break it down, it isn't so hard to code.

Coding from the Partner Perspective

From the Partner Perspective Overview

Once you have been coding publicly disclosed data breaches for a while, you will start to see cases where it is a widely used third party service that was compromised, resulting in all of their customers having to report data breaches. Rather than coding each of those customers as a separate breach, we handle them a bit differently. We code them up from the partner's perspective rather than the customer's perspective, and we do that for one very important reason: we don't want to artificially inflate the data breach count for that time period. If you think about it, this is really just one actual breach—this isn't attackers having to go from company to company and breaking into each of their systems/bypassing each of their infrastructure controls one-by-one. No, this is one actor breaching one infrastructure, and scoring data on lots of different organizations in a business-to-business relationship. So to avoid making the problem bigger and more complex than it really is (and it is already complicated enough, thank you very much), we code it from the perspective of the partner, and list all of the organizations that were affected in the Secondary Victim_ID field, along with their industries.

Example 1: Click2Gov/Superion

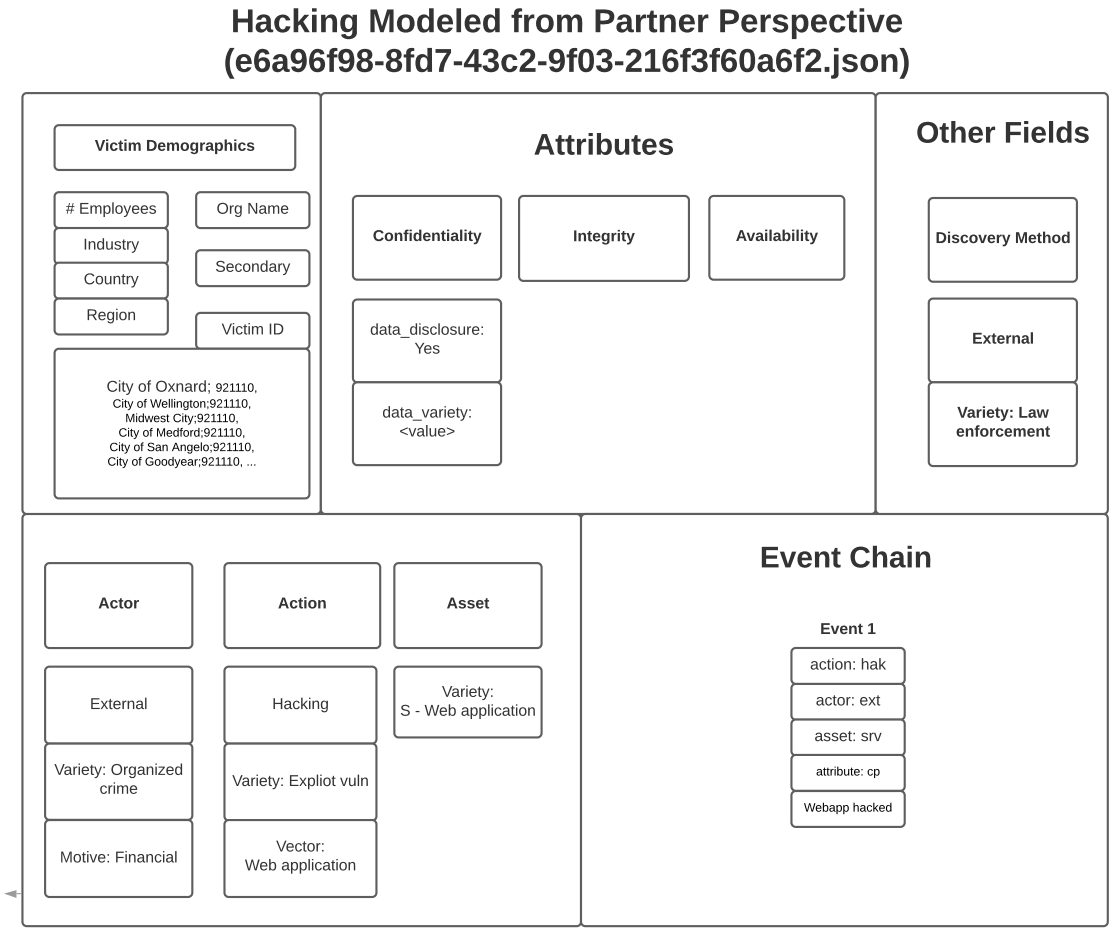
This case has become something of a legend in the VCDB dataset. The vendor serves a large number of government entities—particularly at the city level. They are used by a variety of city agencies to handle payment card processing and records keeping for the city constituents to access the services offered by the municipality. When this company was hacked, we first saw a wave of their

customers announcing the breach.

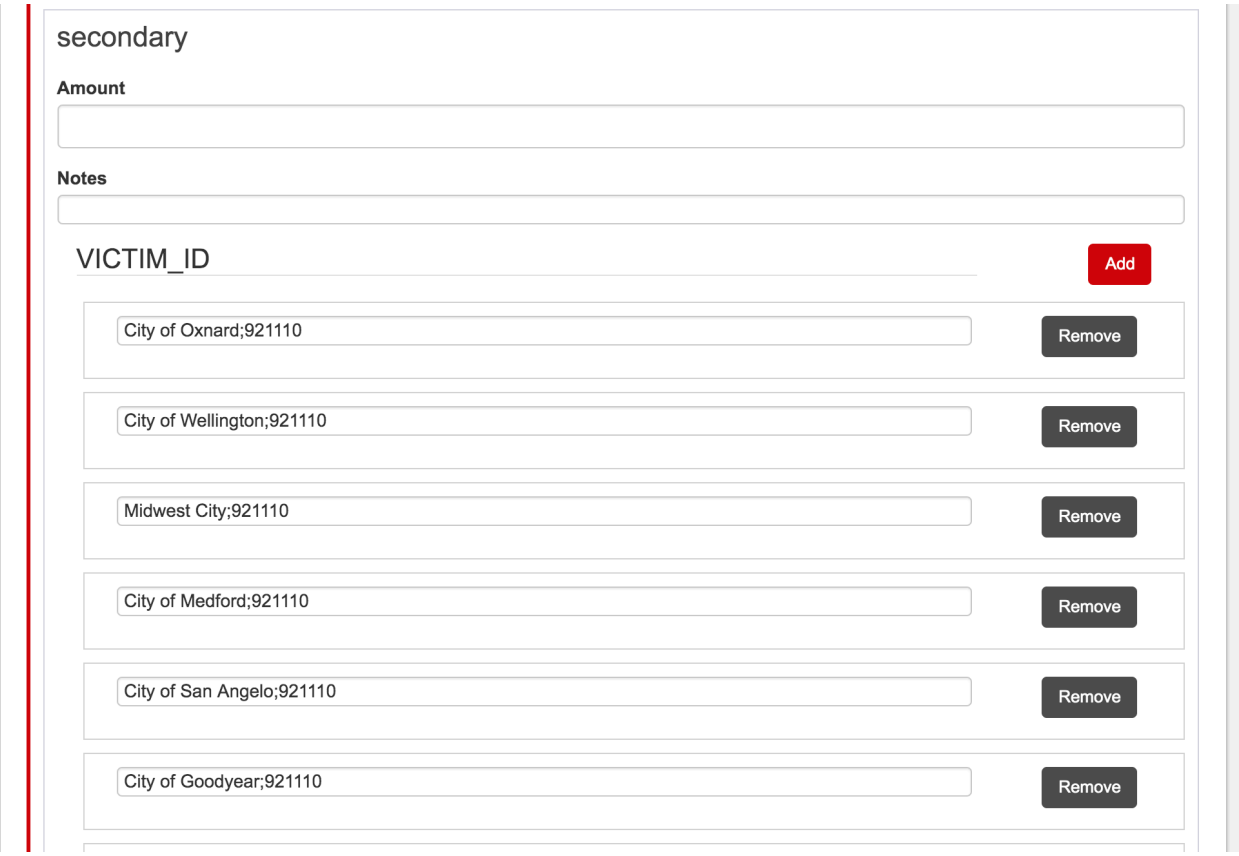
Usually for these partner breaches, we hold off on coding the case until there has been sufficient time for the affected customer organizations to all report the breach and notify their own customers that their data was compromised. This case is a bit of a puzzle to us, because while there was that first expected wave of notifications, after they had died down and we thought it was about time to code the case up, suddenly a second wave of notices hit. Was this the same breach only they found more victims? Did they take a phased approach to their notifications? We don't really know.

But wait, there's more. After that second wave died down finally, and we dared to hope we could code this up (we did actually code the first wave which is why you have a json record in our diagram title), suddenly there started up a new round of notices. Now we're waiting some more to code the second wave and some day we will get the third wave coded too. Given the separation of time, we are inclined to call this a separate set of incidents, but we as a team will have to argue that one out before we code it.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/e6a96f98-8fd7-43c2-9f03-216f3f60a6f2.json



You can see in the diagram that this is under the Victim Demographics section, and if you use the VERIS webapp, you will see under the Victim section, once you expand it out, and scroll down a bit, there is a secondary subsection. Under that, we have VICTIM_ID, and an add button. The form defaults to just one box, but you can add as many as you need to.



We list the victim organizations, and then their NAICS code, separated by a semi colon. This at least allows us to have those two pieces of information per victim organization. When viewed in the JSON file (after it has been generated by the webapp), you can see how it organizes the secondary victims list separate from the partner who was compromised.

```
"victim": {
        "country": [
            "US"
        ],
        "region": [
            "019021"
        ],
        "secondary": {
            "victim_id": [
                "City of Oxnard;921110",
                "City of Wellington;921110",
                "Midwest City;921110",
                "City of Medford;921110",
```

```
                "City of San Angelo;921110",
                "City of Goodyear;921110",
                "City of Thousand Oaks;921110",
                "City of Fond du Lac;921110",
                "City of Beaumont;921110",
                "City of Tyler;921110",
                "City of St. Petersburg;921110",
                "City of Lake Worth;921110",
                "City of Indio;921110",
                "City of Bakersfield;921110",
                "City of Waco;921110",
                "City of Midland;921110",
                "City of Bozeman;921110",
                "Bossier City;921110",
                "City of Ames;921110",
                "City of Topeka;921110",
                "City of ;921110",
                "City of Saint John;921110",
                "Hanover county;921110",
                "City of Pompano;921110"
            ]
        },
        "victim_id": "Superion (Click2Gov) Now DBA Central Square
Technologies",
        "employee_count": "1001 to 10000",
        "industry": "54151",
        "state": "FL"
    },
```


One benefit to handling the data this way is that when you are looking for the large partner (some call this large supply chain breaches) breaches, you look for those cases where there are values in the secondary victim_id field. Otherwise, it would be very difficult to isolate these kinds of cases in a VERIS dataset. This is largely because VERIS is an actor centric framework—so it focuses on who is performing the actions that led to the compromise. Even though it was data held by a partner, it was not a partner actor causing this breach directly.

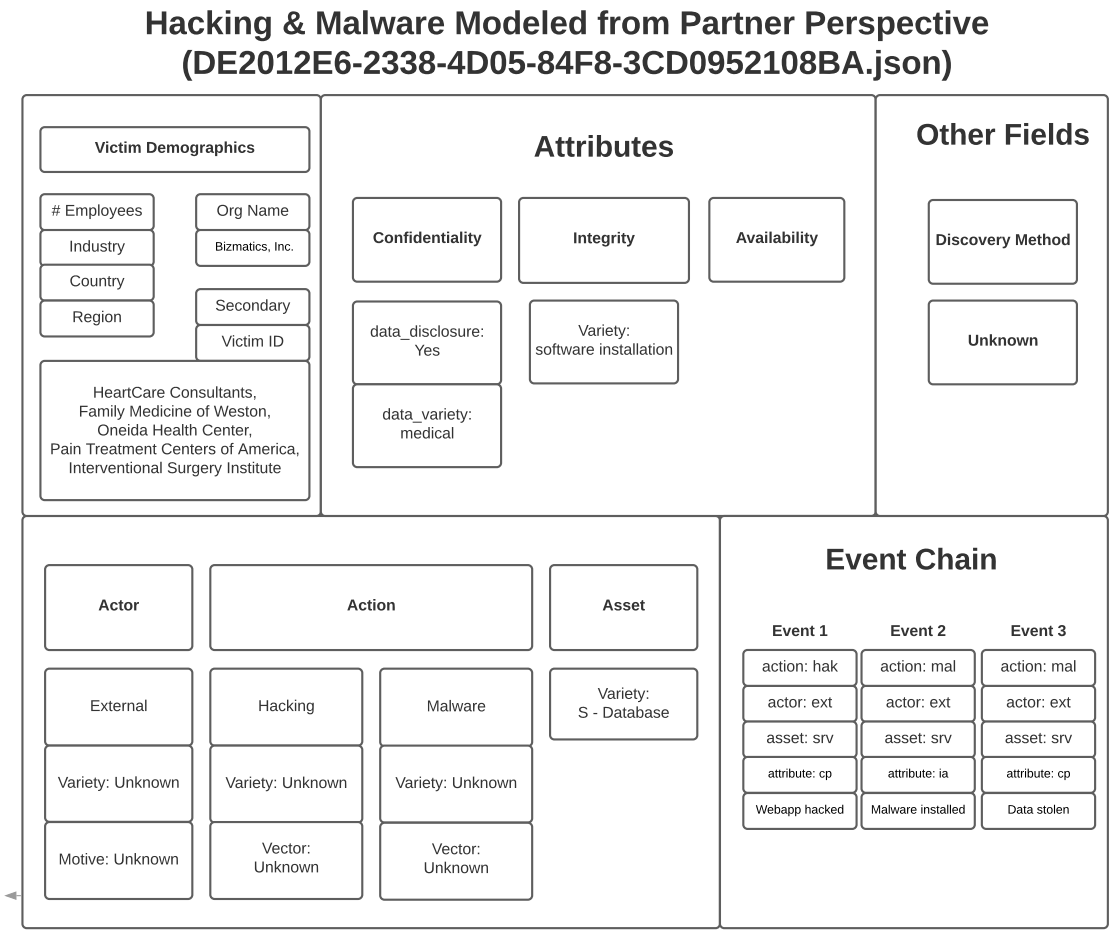
Example 2: Bizmatics

Many of the cases where a large partner is compromised, affecting many secondary organizational victims, involve hacking of one form or another. In our next example, we have the Bizmatics, Inc. case. They provide electronic health records software to medical customers of various types.

> "On March 30, 2016, Integrated Health Solutions, P.C. was informed by Bizmatics, Inc. (\"Bizmatics\"), our electronic medical records service provider, that it experienced unauthorized access to its records which may have included our patient records. This information may include name, address, social security number and health visit information."

Again, this involved a number of organizations, and the records were entered into VCDB separately. After it became apparent they were really all victims of the same partner breach, they were consolidated into one json record, coded from the partner organization's perspective.

Here is the link to the json file in the VCDB GitHub repository: https://github.com/vz-risk/VCDB/blob/a98bbdb01820eacaeef24e76f41e30c322b252bf/data/json/validated/DE2012E6-2338-4D05-84F8-3CD0952108BA.json.



I have listed a sampling of the organizational victims under the Secondary victim ID field. There were many more. The data disclosure was of medical information, given that this is a medical records system that is unsurprising. There was software installation, which is an integrity violation as well.

There was no indication in the references as to how this was discovered, unfortunately, so that is listed as Unknown.

The actor is external to the organization that was hacked—remember we do not consider this a partner actor breach—the person performing the actions was external to the victim organization. We don't know anything about their variety or motivation, so both are listed as unknowns.

The actions were hacking and malware. Not much detail is available—a common problem when we must rely on what interests the reporters covering the story, and the tidbits of information that the impacted organization provide in their notifications. So it is unknown hacking and unknown malware. We classify the electronic medical records system as a type of database for the Asset.

The event chain is three events—the unknown hacking action that got the adversary access to the server, and then the installation of the malware and subsequent theft of the data.

Conclusion

You've made it to the end, and now hopefully (if we've done our job), you feel much more confident in your ability to code cases using VERIS and potentially the VERIS Webapp.

Thank you for reading our Coding Style Guide

While this is a living document and will continue to evolve with the VERIS Framework, we hope it has been useful for people who are coding cases right now. If you have comments or questions, please feel free to email the DBIR team at DBIR@verizon.com and we will do our best to help out.

We would also love to hear how you are using VERIS in your own environments. Real world use cases are always of interest to us, and help us to understand how VERIS needs to evolve over time to remain relevant to people who do incident response and need to record these kinds of metrics.

Finally, if you would like to share data with us (pre-anonymized, of course), please reach out to us and let us know. The research we do is dependent on people like you sharing data for the greater good of the Information Security community.