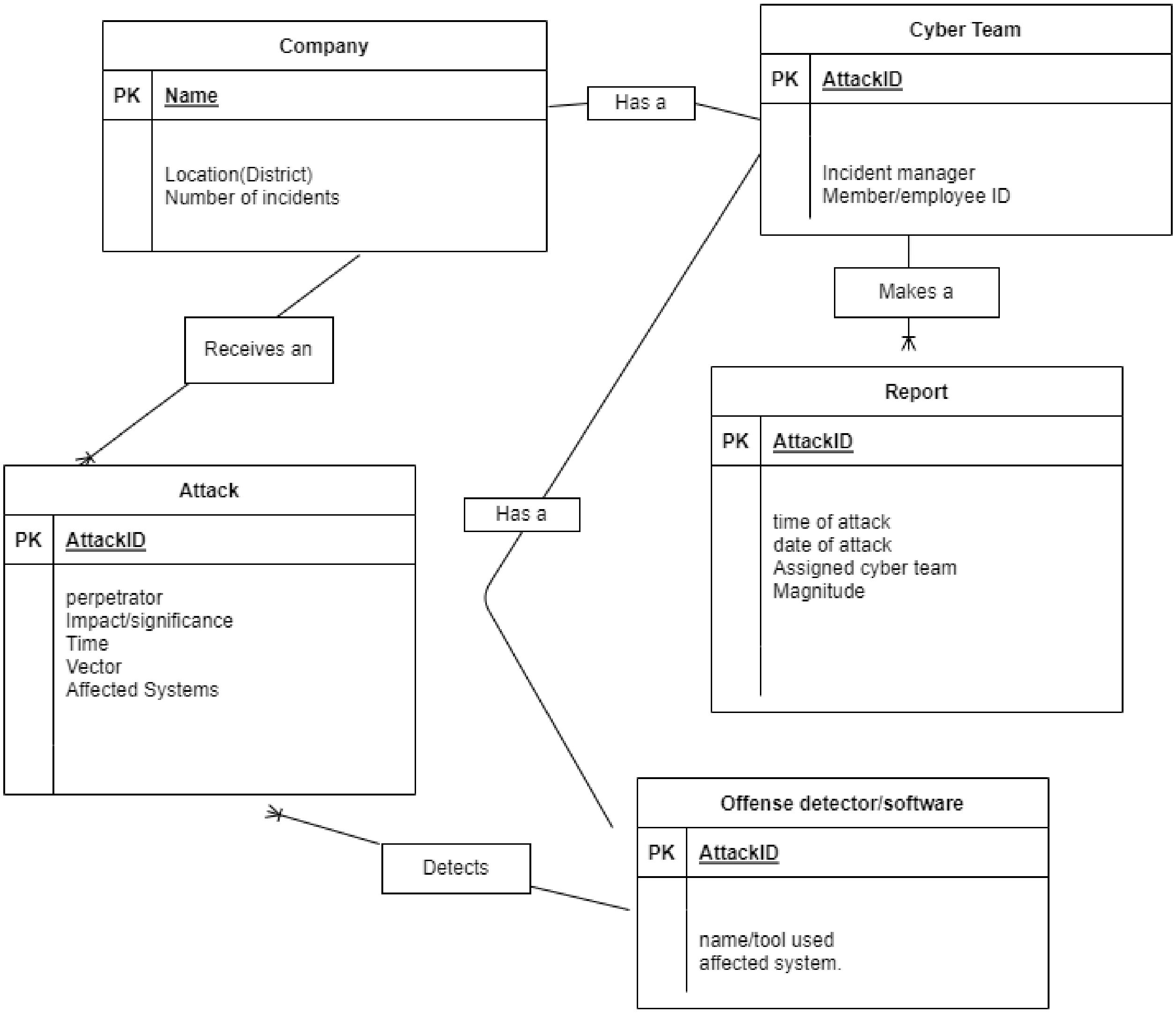
Entity Relation diagram



ERD into the Relation Model

Company{company name, num of incidents, location,}

Type: String.

Domain = (num of incidents, location, Company name).

Foreign Key: Location.

Primary Key: Company name

Attack{Perpetrator, Attack ID, impact/significance, time, vector of execution, affected systems}

Type: int and string.

Domain = Perpetrator, Attack ID, impact/significance, time, vector of execution, affected systems

Foreign Key: time

Primary Key: AttackID

Offense Detector{name of tools used, affected systems}.

Type: String.

Domain: Name of tools used, affected systems

Primary Key: Name of tool, affected systems.

Foreign Key: None.

Cyber team{employee ID, attack ID, incident manager}

Type: String and int.

Domain: (employee ID, attack ID, incident manager)

Primary Key: Incident Manager

Foreign Key: Attack ID

Report{Attack ID, time of attack, date of attack, assigned cyber team, magnitude}.

Type: Int and String

Domain: {Attack ID, time of attack, date of attack, assigned cyber team, magnitude}

Primary Key: Time of attack

Foreign Key: Attack ID

Identification of Functional Dependencies

In relational database theory, a functional dependency is a constraint between two sets of attributes in relation. For example, two attributes that would be included within this project are Time of Attack and Vector of Execution. If for every vector of execution there is one time of attack, then the two are said to be functionally dependent. Our database has two functional dependencies: The attack ID is functionally dependent on time, therefore and Company Name is functionally dependent on Location. Therefore, Attack ID -> Time and Company Name -> Location.

Normalization of Relations

The process of decomposing the tables extracted from the ERD translation into relations satisfying BCNF/4NF involves splitting, configuring and adding relations so that for every FD, x-> y, x is a superkey. In every relation in our database, the superkey will be the attackID. With this in mind we split the relation into an x->y such that x is a super key. In our database, the attackID will always be the superkey, so each relation will be split into AttackID, and the next attribute within the relation. We apply this same process to the other entities with PK attackID and the rest of its attributes.

Sample Data

Report

Attack ID	Time of Attack	Company	Assigned Cyber Team	Magnitude	Date of Attack
1	10 6:09 pm	VCU	Cyber Team 2	Severe	12/06/2002
2	10/13/2022	IBM	Cyber Team 1	Low	10/14/2022

Company

Name	Location	Number of incidents
IBM	Hampton	123,408
JB Hunt	Chantilly	1245,521
CarMax	Richmond	457,111

Cyber Team

Attack ID	Incident Manager	Employee ID
07	Bob Ross	01

Offense Detector

Name of Tool Used	Affected Systems
Qradar	Routers

Attack

Attack ID	Impact	Vector of Execution	Perpetrator	Affected Systems	Time
02	Severe	XSS	249.126.45.141	Computers	6:03 pm