HELPING THE WORLD MAKE SENSE OF DATA

neo4j

GRAPHS FOR CYBERSECURITY

Dave Voutila, Gal Bello, Tara Jana (TJ), and Deb Cameron

Table of Contents

Introduction

Cybersecurity in the Time of Pandemic

Cyberattacks had been on the rise for years, with nation state threat actors and foreign hacking collectives joining in, devoting more time and resources to attacks. At the same time, the shortfall in talented staff available to fight these threats also increased.

Then the COVID-19 pandemic struck, forcing many businesses to shut down their physical buildings, creating an unprecedented demand and pressure on digital platforms as businesses shifted to securing work from home at scale.

Cyberattacks skyrocketed in 2020. The FBI saw a never before seen 300 percent increase in cybercrime reports in 2020. Between January and April 2020, there was a 238 percent rise in cyberattacks on financial institutions.

Software supply chain attacks, which exploit vulnerabilities in embedded open source libraries, increased 430 percent in 2020 and continued their meteoric rise with a 650 percent increase in 2021.

This is not news to those in cybersecurity. After all, it is a field characterized by extreme, unmanageable complexity. Something seemingly straightforward like Incident analysis requires pulling together data and logs from many platforms and tools. To effectively mitigate cybersecurity risks, we need advanced data solutions that empower us to correlate and analyze connections at a real-world scale.

The Odds Are Stacked Against Defenders

Security professionals have long lamented the difficulty of defending against cyberintruders. Consider the following:

- Defenders have a bigger job. They must protect against every possible attack and patch every possible vulnerability. Attackers need to find only one opening to land and expand.
- Defenders have many responsibilities. Attackers are often hyperfocused on finding exploits.

A Graph for Cybersecurity Flips the Script

Defenders traditionally rely on lists: alerts and logs from software tools. Such an approach blocks defenders from gaining a holistic view of their systems and creates blind spots.

Meanwhile, attackers are opportunistic. Attackers find a weakness – no matter how small – then exploit it to gain access to more of your network. They think of your network as a graph. If they get access to one node, they can build an attack graph from that node, working toward the most valuable systems and data.

Defenders can enhance their security posture by building a complete graph of their infrastructure and enriching it over time. Here defenders are at an advantage (for once). After all, the attackers' "graph" is the developing understanding of the network structure as they discover it. The defenders' graph should take into its sweep the entire infrastructure, creating a digital twin that is as complete as they choose to make it.

> *Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win.*
>
> John Lambert
> Distinguished Engineer and General Manager,
> Microsoft Threat Intelligence Center

Graphs Give Defenders the Advantage

	Defenders	Attackers	Advantage
Scope of effort	Defend **every device and component** on the corporate network	**One vulnerability**, one open port, one phishing click	**Attackers**
Available time and effort	**Scarce:** Understaffed and overworked	**Dedicated**: Focused and funded	**Attackers**
Think in graphs?	**Traditionally, no**, but this is changing	**Yes**	**Attackers and savvy defenders**
View of the graph	**Complete**: Create a digital twin of the entire environment	**Piecemeal**: Discover graph based on point of entry	**Defenders**

Why Graph Databases for Cybersecurity?

Graph databases easily capture the complexity of IT infrastructure and security tools. Graphs are the most natural way to process data because they provide a high-fidelity model of the real world.

Assess Your Need for Graphs for Cybersecurity

- Does your organization maintain a live representation of your network structure for analysis?
- Are you aware of the most likely attack paths threatening your crown jewels?
- What is your organization's plan for handling security breaches?
- What contextual information does your security team have to deal with it?
- Does your organization have processes to review user access rights and tools to help visualize a user's security groups versus the rights they need?

A graph data model represents intricate networks of entities and their relationships, and uncovers patterns that are difficult to detect using traditional representations such as tables. Tables are good for collecting and processing data but they miss relationships between data points.

Graph databases are a strong fit for cybersecurity: they integrate many data sources, incorporate large data volumes, and easily reveal dependencies.

Security data comes from many sources; large organizations have an average of 75 security tools deployed. Each tool generates alerts and logs. And security tools are just one type of data source; many applications and services generate log files that are relevant to cybersecurity.

Alerts and logs produce voluminous data. Add to this the relationships in and across all that data, the dependencies and paths from one resource to another.

Modeling your infrastructure as a graph enables you to:

- Identify your most valuable assets (your "crown jewels") and target security investments
- Generate alerts for relevant teams about the impact of incidents across systems
- Identify suspicious behavior, reducing the mean time to detection and uncovering insider threats
- Analyze and rationalize identity and access management to enforce the principle of least privilege

The advantage of graph databases increases with the size and complexity of the data. With a graph database like Neo4j you gain a unified visualization of the attack surface and the ability to conduct ongoing cyber risk assessment simply by connecting your resources and users with the activities on your system. You can have predictive, preemptive, and proactive threat identification and cyber risk management with clear attack paths and reachability routes. You can protect systems, detect anomalies in real time, respond with confidence to any incidents and recover quickly.

There are many ways to use graph databases to enhance cybersecurity. This paper focuses on one essential use case: creating a digital twin of your infrastructure by capturing it in a knowledge graph. That might sound like a big project, but with graphs you can start small, realize value, and expand your graph to provide more value over time.

This paper has three parts:

- **Part 1** describes the current threat landscape to provide context for those who are not cybersecurity afficianados.
- **Part 2** offers policies and strategies to consider, with some simple graph queries to show you how you might implement them.
- **Part 3** covers the basics of creating a digital twin of your IT infrastructure, including code samples. All the code in this paper can be found in its Github repository.

The paper concludes with next steps, followed by an appendix with links to useful open source tools.

Part 1: Cyberthreats, Vulnerabilities, and Risk

This section provides an overview of threats, tradeoffs related to cloud versus on-premise deployment, and a look at sources and impacts from security breaches.

Those who are well-versed in security may wish to skip to Part 2 to learn more about how graphs aid cybersecurity.

Cybersecurity Threats

Social engineering refers to manipulating people into performing actions against their interests or those of their organization. For example, an attacker may trick people into giving out sensitive information over the phone or leave a memory stick containing malware in a company parking lot, hoping that a curious employee will connect it to a computer. A seemingly small initial breach can result in significant subsequent attacks. Phishing is just one type of social engineering, and yet, according to a PwC report, it accounts for over half of all reported security incidents.

Ransomware involves removing a person or company's access to their data or systems. The attacker offers to reinstate access if a ransom is paid and may threaten to publish extracted data if a ransom payment is not made. It is a financial and potentially a reputational risk if customer systems are unavailable or sensitive data is published. Ransomware is reportedly the top cybersecurity risk for small and medium-sized businesses. For example, disruption from a ransomware attack in 2017 is estimated to have cost the UK's National Health Service £92,000,000.

Distributed denial-of-service (DDoS) attacks are carried out using hijacked computers that send so many requests to a service that it cannot carry out its regular work. "Denial of service" means that the computer systems will no longer function, leading to potential severe loss. Cisco estimated that by 2023, the number of DDoS attacks would increase to as many as 15.4 million per year.

Risks from third-party vendors. Most companies use third-party vendors for commodity services so they can focus on their core business. services, providing a competitive advantage. However, this benefit runs increased security risks when third-party software is compromised and exposes your systems or data. From 2017 to 2019, the number of data breaches caused by third-party vendors increased by 35 percent. Recently, an exploit on Fujitsu's ProjectWeb leaked 76,000 addresses from the Ministry of Land, Infrastructure, Transport, and Tourism in Japan.

Security Tradeoffs: On-Premise versus Cloud

Hosting systems yourself offers maximum control, but there's a tradeoff – full responsibility. The hardware, the network, and everything running on it must be staffed, designed, built, and maintained.

Keeping software and systems you manage secure requires constant vigilance and discipline. You must monitor for new vulnerabilities in all software, operating systems, utilities, and hardware that your organization uses. Patching is essential.

While cloud services can bring huge advantages, their users undeniably give a third-party organization some control of their systems. However, there is less visibility of how the systems work and their dependencies and, therefore, less ability to assess and mitigate risks.

Often your team needs to provision cloud systems, and each cloud provider offers its own configuration tools. A lack of in-depth experience may lead to misconfigurations and potential security holes. You can mitigate this by getting help from cloud vendors or partners, but then you have the overhead of granting third parties access to your cloud systems.

If a cloud system is hosted in a different country, it may not be subject to the data protection laws that you must comply with. And you will be responsible for any fines for noncompliance, not the cloud provider.

Organizations need to do a cost-benefit analysis of different options on the spectrum of control and flexibility and find the optimal point for each use case. One option is to run a hybrid model where you manage systems handling crown jewels with a high breach cost in-house while you can run less critical resources in the cloud after putting careful policies in place.

Sources of Security Breaches

External breaches exploiting vulnerabilities are the classic attack vector, but an organization can be vulnerable to breaches by insiders even with perfect security in place. According to a 2019 Verizon report, internal parties were involved in over a third of breaches. They can provide information to external parties to enable their access or can directly leak data requiring privileged access. That said, insider involvement may be intentional or unintentional (as with phishing and other forms of social engineering).

Impacts from Cybersecurity Breaches

The impact of cybersecurity breaches is manifold, from loss of productivity to reputation damage to fines, all of which directly or indirectly impact the bottom line.

Downtime creates a loss of critical organization capabilities, an inability for customers to access systems, which often means an immediate loss of sales.

A Forbes report found that 46 percent of organizations suffer **reputation damage** following a data breach. Data protection regulations oblige companies that suffer a data breach to inform their customers, and the media may report on it, further impacting your brand's reputation.

Remediating any type of breach comes with **financial costs.** The average cost of a data breach in 2021 was $4.24 million. On top of these costs, many countries and industries have regulations requiring organizations to secure personal data they hold. Violations of these regulations come with stiff **fines and penalties.**

Companies operating in the European Union can face fines of up to 4 percent of annual gross revenues if they violate the General Data Protection Regulation (GDPR). Google was fined $50,000,000 in 2019 for such offenses. GDPR has become a model for similar regulations worldwide, such as the California Consumer Privacy Act and Brazil's LGPD.

In the U.S., HIPAA protects personal healthcare information (PHI). Companies must not disclose such information to anyone except a patient and the patient's authorized representative without their consent. If breached, the fine can be up to $25,000 per violation category per year. In 2020, Premera Blue Cross was fined $6,850,000 for unauthorized access to the PHI of more than 10 million individuals.

PCI DSS is an industry standard rather than a regulation, but it fills a similar role for card issuers and any merchant storing, processing, or transmitting cardholder data. It protects payment card information internationally by requiring accredited companies in breach of its rules to pay fines up to $100,000 monthly. The Equifax data breach in 2017 compromised 147 million Americans' social security numbers, birth dates, addresses, driver's license numbers, and credit card numbers. The settlement cost $425 million.

Part 2: Cybersecurity Policy and Strategy

Effective Security Posture

Security posture refers to your awareness of your assets, your processes to monitor and maintain their security, and your ability to detect, handle, and recover from any attacks.

The process of monitoring your assets should operate continuously and, where possible, automatically. Creating a graph of your infrastructure and setting up automated alerts is a solid approach.

Automated reports can be run regularly to flag, for example:

- Any new security perimeter nodes (connected to the public internet)
- Any nodes connected directly to the perimeter without any security infrastructure in between
- New attack paths to the most important resources (crown jewels)
- Nodes in the infrastructure that may be impacted by newly discovered security vulnerabilities, prioritized by their centrality in the network

This part of the paper includes some sample queries in Cypher Query Language that show how easy it is to analyze your security posture. In addition to serving as practical examples, these queries may inspire you to create a knowledge graph of your IT infrastructure, as described in Part 3.

Cybersecurity Audits

A cybersecurity audit is a broad review of the organization, its IT infrastructure, and its processes to identify the threats, vulnerabilities, and risks it is subject to. An independent third party typically performs the review.

Generally, an audit should start with a review of the organization's policies around mitigating cyber risk, data storage and management, and incident handling and response. Next, the audit should consider staff training and awareness of policies, technical controls that monitor for breaches, and the organization's physical security practices.

An effective cybersecurity audit requires identification of the organization's crown jewels, possible attack vectors, and appropriate security posture and processes. Even if audits are done well, if they are infrequent, vulnerabilities may be exploited.

With your security infrastructure in a graph, you can verify that you have addressed issues raised in an audit and automatically monitor and alert to protect your systems between audits.

Defense in Depth

Defense in depth means not relying on a thin security perimeter such as a VPN to maintain security but instead having multiple layers of security.

Visualizing infrastructure as a graph is a natural way to discover the weakest points in your perimeter so that you can better protect them.

A zero-trust approach takes this to another level by abandoning the concept of trusting resources inside the corporate network and mistrusting those outside it. Instead, treat all systems as inherently untrusted, requiring them to authenticate. A graph of your infrastructure can enable you to check that all the systems in your architecture follow the zero trust principle. For example, suppose there is a connection between two resources without a node representing a security check between them (such as a key-pair). A query that finds this pattern shows you which elements in your network need additional security to follow the principle of zero trust.

Budget and Investment in Cybersecurity

Many security teams struggle to get the funding they need to secure their systems properly. If the likelihood and the potential cost of a breach are well understood, you can present proposals with a projected return on investment. Security staff need the tools and training to do their job effectively. Having a graph of your infrastructure gives you visibility to find vulnerable systems, enabling you to estimate what it will take to remediate the risks, as well as the potential cost and impact if those resources are compromised.

Training

Research from Tessian and Stanford in 2022 found that 88 percent of cybersecurity breaches are a result of human error. Training employees is, therefore, a critical part of reducing the risks and costs associated with those errors.

A large share of this 88 percent are social engineering attacks where people are tricked into exposing data or giving access to unauthorized people. While technical measures

can help in some instances, training is the only way to reduce the risk of these types of attacks. Employees need to understand the risks of providing information over the phone, being tailgated as they walk through doors, using public wifi with company devices, and being overheard in public spaces.

Cybersecurity staff need to be trained to analyze events that security systems flag. In addition, the constant change in the cybersecurity world means that these staff will likely need ongoing training and time dedicated to keeping their knowledge up to date.

Policies and Procedures

Having written policies and procedures is essential. Here we highlight a few policy areas, including query samples where applicable.

Least Privilege

Companies should implement the principle of least privilege, which means giving employees only the permissions they need to do their jobs and no more. If an account is compromised or a user leaks data, enforcing this principle minimizes the scale of the loss.

A common way to manage the principle of least privilege is to create groups with permissions and assign users to those groups. The difficulty with such a policy is ensuring it is applied consistently, as users join, change roles, or leave the organization.

Creating a graph of your users along with their security group memberships enables you to run queries to flag users who are not assigned to a security group.

In the Cypher query language, this statement finds users who are not in a security group:

```
MATCH (user:User) WHERE NOT (user)-[:IN_GROUP]->(group:SecurityGroup)
RETURN user
```

Patching

Keeping systems patched is a constant challenge. Companies should set up **automatic updates** for systems and users' devices to minimize the risk of compromise from unpatched vulnerabilities. In addition, devices' storage can be encrypted so that if a device is stolen, attackers are less able to access the data on it.

Software Installation Policy

A s**oftware installation policy** can minimize the chance of users installing software with vulnerabilities, backdoors, or viruses.

If you create a graph of that specifies software that is allowed by policy, you can use a Cypher query to find software that is not in compliance with your policy:

```
MATCH (sw:InstalledSoftware) WHERE NOT (sw)<-[:ALLOWED_BY_POLICY]
   -(p:Policy)
RETURN sw
```

Authentication Policies

A **password strength policy** can be enforced to ensure that passwords are reasonable in length or complexity, that they are not reused, and so on.

Users should use **multi-factor authentication (MFA).** MFA typically requires entering a code sent to one of the user's devices as an additional step to verify their identity.

The following query finds users who do not have MFA enabled:

```
MATCH (u:User) WHERE NOT (u)<-[:HAS_POLICY_ENABLED]-(policy:MfaPolicy)
RETURN u
```

Backups

Automated processes should **back up all data** regularly or store it in the cloud by default to reduce the risk of data loss if data is accidentally deleted. This query finds data stores that do not have a backup:

```
MATCH (d:DataStore) WHERE NOT (d)-[:HAS]->(b:Backup)
RETURN d
```

Vendor Security Evaluations

Policies must also apply to vendors. Suppose a third party suffers a data breach involving your data. From your customers' perspective, that may be indistinguishable from your own company suffering a breach, and you may be liable for losses and fines. In a Ponemon report from 2021 on third-party security, 63 percent of respondents said that they rely entirely on the reputation of third parties when choosing a supplier and don't perform their own evaluations of suppliers' security practices. You need a vendor management policy that includes initial and periodic security evaluations.

Part 3: Building a Digital Twin for Cybersecurity

This section provides an overview of building a knowledge graph with information to support you at all stages of an attack - from before it begins to until it ends as well as for subsequent incident analysis.

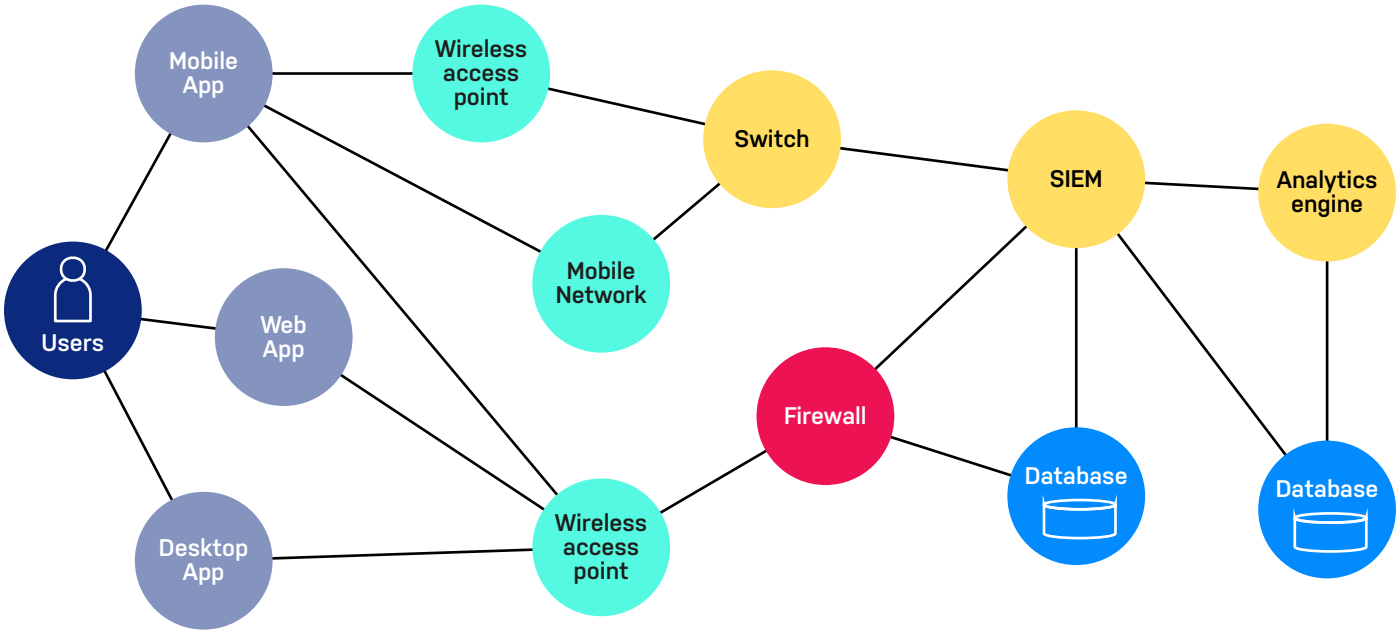
Such a knowledge graph creates a digital twin of your environment, enabling you to represent all or part of your network data in a holistic view. This view is very useful for cybersecurity analysts to query and take action on. In addition, the knowledge graph can be analyzed by data scientists, who build models to detect malicious activities.

The task of automating and systematizing cybersecurity relies on processing the organization's security data and bringing it into the graph. The digital footprint might include:

- Which systems connect to which systems
- Which systems are open to the Internet
- Users and the groups they belong to
- What permissions are given to members of the groups
- Your policies and the systems to which they are applied
- The systems that are most important to protect (the crown jewels)

Your graph may include events like:

- User access events
- Application resources usage
- Devices connected
- Service health



A sample cybersecurity knowledge graph

Creating a Graph of Network Infrastructure

Networks are graphs by nature, comprised of connections. Storing this information in a graph database enables actionable insight and analysis.

It is easier than you might think to create a graph of your network. For example, the major cloud providers offer metadata API services for describing the infrastructure that a customer has in place. This data can then be easily ingested into a graph database like Neo4j. For example, Google Cloud offers the Cloud Asset Inventory; Microsoft Azure has the Azure Resource Manager and Microsoft Graph API; AWS offers its AWS Organizations API.

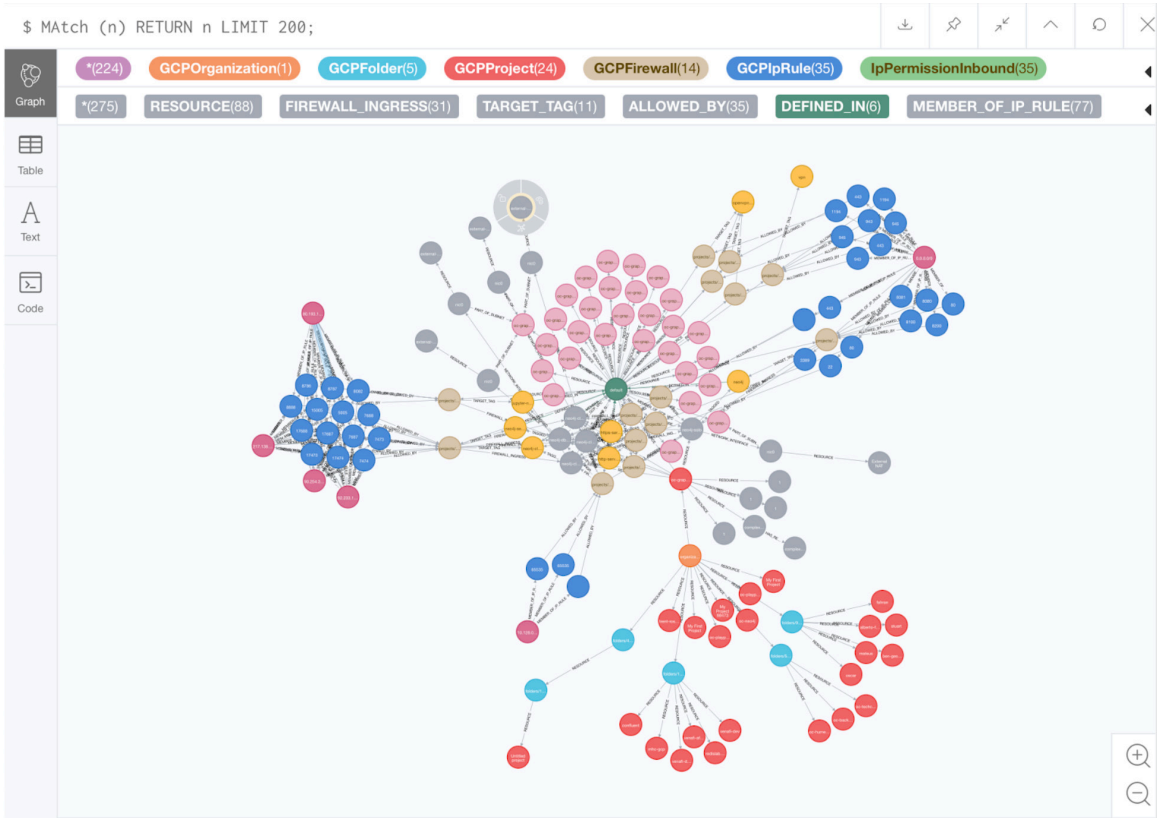
Alternatively, some tools create a network graph for you, such as Cartography, an open source tool built on Neo4j and described in the appendix for this paper.

For on-premise platforms, you can investigate the system resources – in VMWare or HyperV, for example – using the vendor's provided APIs.

Once the information is in the graph database, it is straightforward to run queries and visualize the network structure.

This simple Cypher query returns 200 nodes, as shown in the following screenshot.

```
MATCH (n) RETURN n LIMIT 200
```



Part of a cybersecurity knowledge graph

With your network structure loaded into the graph, you can identify the crown jewels that need the most protection. Give special labels or properties to nodes for valuable resources. You can then easily query them and track dependencies. Add relationships that mark critical access paths (our sample graph includes relationships such as `FIREWALL_INGRESS` for example).

Adding Software Information to the Graph

If you are already using configuration management tools, you can use their APIs to ingest the versions and health reports of the software and operating systems they monitor. If you are not using configuration management tools, you can start with Nessus or Nmap, which can scan networks and devices and combine that information with software activity logs to provide information such as versions, libraries, ports they use, who is accessing them, and which resources they rely on. You then ingest this information into your graph.

Enriching the Graph with Threat Intelligence

MITRE, NIST, and others have created vulnerability dataabases that are used industry-wide. By ingesting this data into your graph, you can see where the vulnerabilities impact critical resources and reveal potential attack paths to those resources.

The ATT&CK-D3FEND matrices, developed by MITRE, are a freely available set of tactics and exploits that attackers have been known to use and a corresponding set of defensive measures used to counter them. Once you visualize your infrastructure as a graph, you can see the set of attack paths, including information from ATT&CK. You can then link to the defense tactics from D3FEND. GraphKer, an open source tool discussed in the appendix, simplifies the process of loading threat intelligence into Neo4j.

Visualizing Vulnerabilities and Attack Paths

Attack paths take the attackers' perspective and show the path of potential multistage attacks along with the vulnerabilities used at each stage. This process goes beyond a static list of vulnerabilities and looks at how an attacker could use them.

An attack path analysis can identify different paths an attacker may take through your infrastructure to reach those assets. All the attack paths on a network together form an attack graph. Visualizing the paths as a graph is intuitive, especially with a graph data visualization tool such as Neo4j Bloom.

Assessment, Planning, and Preparation

A digital twin of your IT environment has tactical advantages. You can programmatically assess the impact of changes to your environment before implementing them.

Having an approval gate to check changes is critical to protecting the system. Depending on the rules you implement, you can automatically reject or roll back suspicious changes or raise tickets to review them manually.

Analyzing the Digital Twin Using Graph Algorithms

One approach to security is to find the types of vulnerabilities that affect the largest share of your infrastructure. Then, assessing the likelihood of these assets being breached will reveal where to focus resources to achieve a tremendous increase in security.

A more visionary approach looks at each node's context in the network to decide on the most important nodes to focus on.

The Cypher examples in this section make some assumptions:

- You have created a graph of your IT infrastructure and classified some nodes as externally facing.
- You know the node IDs of your most valuable assets (your crown jewels).
- You have installed Neo4j Graph Data Science so you can run graph algorithms to analyze your graph as a whole.

In this Cypher example (available in the code repository for this white paper), we call a shortest path algorithm to discover the shortest path to a valuable resource called target node. We build a graph of all the paths from the Internet to a specific crown jewel and add a `POSSIBLE_ATTACK` relationship.

```
# build a graph of all shortest attack paths to a specific crown jewel
MATCH (external:ExternalFacingNode)
CALL gds.shortestPath.dijkstra.stream("attackPathGraph",
{ sourceNode: id(external),
  targetNode: targetId,
  path: true})
YIELD sourceNode,targetNode,nodeIds
# create POSSIBLE_ATTACK relationships
WITH nodeIds
UNWIND apoc.coll.pairsMin(gds.util.asNodes(nodeIds)) AS pair
WITH pair[0] AS toNode, pair[1] AS fromNode
MERGE (toNode) <- [:POSSIBLE_ATTACK] - (fromNode)
```

We then get all the pairs of nodes in that attack graph and score them using the betweenness centrality algorithm. We then write those scores back to the database:

```
CALL gds.graph.create('centralityGraph', '*',
    { relType: {
    type: '*',
    orientation: 'UNDIRECTED',
    properties: {}
    }}, {});
CALL gds.betweenness.stream('centralityGraph', {})
    YIELD nodeId, score
    WITH gds.util.asNode(nodeId), score AS betweenness
    RETURN n.name, betweenness ORDER BY betweenness DESC
```

Each node is returned with a score based on how many possible attack paths go through it.

Attaching a probability to each hop in the attack paths enables you to use the graph as a vulnerability tree and predict which attacks are the most likely to succeed.

A score combining the two numbers we have calculated – the node's centrality and the likelihood of attacks being carried out – would be a powerful metric for prioritizing efforts.

Threat Intelligence and Prediction

Certain events increase the likelihood of attacks. For example, when a company is mentioned in the news, it is more likely to come onto attackers' radar.

Similarly, as new vulnerabilities are disclosed, attackers will test them out. Suppose you have set up a system to ingest vulnerabilities from vulnerability databases automatically. Then, as described earlier, you can quickly build reports and create alerts about the potential impact on the crown jewels and take proactive security measures to mitigate those threats.

Detection

Unfortunately, it is impossible to guarantee complete security, so detecting breaches is critical if any measures to limit the attack are to be taken.

With your data loaded into a graph database, you can automatically run queries to flag specific patterns of events on the network.

You may detect **suspicious actions** by looking for unusual patterns. For example, you may want to monitor how many attempts an IP address has made to log into a system or systems or how many logins you are seeing from a country people do not normally log in from. A graph showing accounts and IPs logged into them would likely make it easy to identify such patterns.

You can uncover **data leaks** by monitoring access to files from unexpected accounts and IP addresses. For example, suppose a user rarely looks at files on the network but suddenly accesses many files using IP addresses or devices that they have not used before. Such events may indicate that an account has been compromised, and you can detect this automatically.

If you detect that your network is being scanned and analyzed, this can indicate the start of an attack. In addition, if you know which resources have been scanned, you know where the attack is most likely to hit and can take additional measures to protect or obfuscate that target.

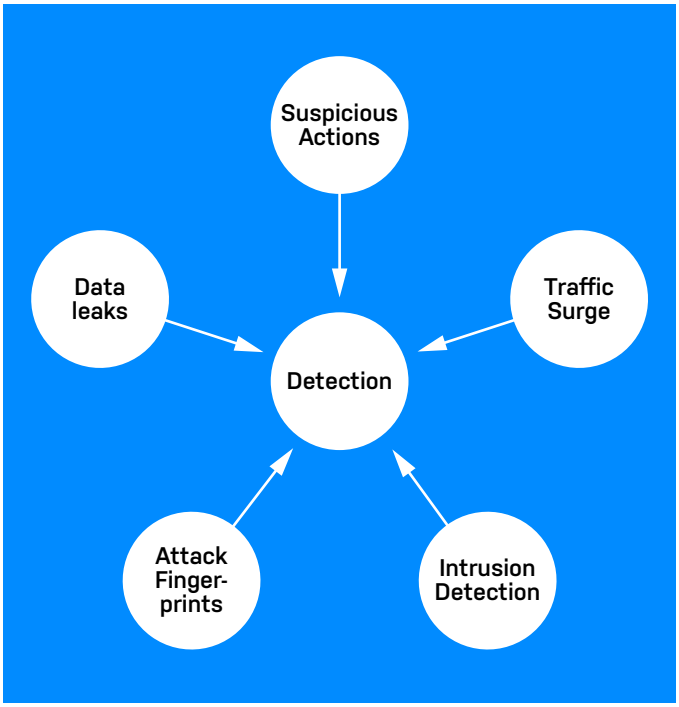
If you pick up **attack fingerprints** that have been seen before, this may mean that a similar attack strategy is being followed – perhaps even by the same attacker.

Intrusion detection systems (IDS) monitor your network for signs of intrusion. Load IDS data into the graph to see whether an attack is in progress.

An unexplained network **traffic surge** may be the start of a DDoS attack.

Once you detect an attack in progress, you can use the graph to find out more about the attackers, including their IP addresses.

Correlating these signals, you can more easily detect attacks and mitigate any future attacks.



Sources of input for detecting breaches.

Investigation and Response

Once you find suspicious events, the top priority is to discover whether they are innocuous or represent an actual attack.

Suppose you can holistically view the suspicious events together on a graph. Then, you can find the commonalities between them and assess the scale of the attack and where it might be originating from on the network. Graphs further empower you to examine all of the contextual information around these events, such as the user accounts and devices involved.

Graph visualizations show the critical information that will be required to determine how to stop the attack - potentially by blocking user accounts or access from specific IP address ranges.

Cyberattacks are usually a chain of system compromises. If you were to model all the cyberattacks you see and their chains of compromises, that model would naturally take the form of a graph. When you are suffering a cyberattack, predicting the attackers' next move is a case of matching the latest attack with a node on the graph and seeing which events most often came next.

With system dependencies modeled in the network graph, you can run a simple query to find out which other systems will be affected when one system is compromised or taken offline by a DDoS attack for example:

```
MATCH (compromisedSystem) <- [:DEPENDS_UPON*] - (system:System)
RETURN system
```

Next, the security staff can alert the teams responsible for the systems under attack. To query for the teams maintaining the systems dependent on the compromised system, you may run a query like this one:

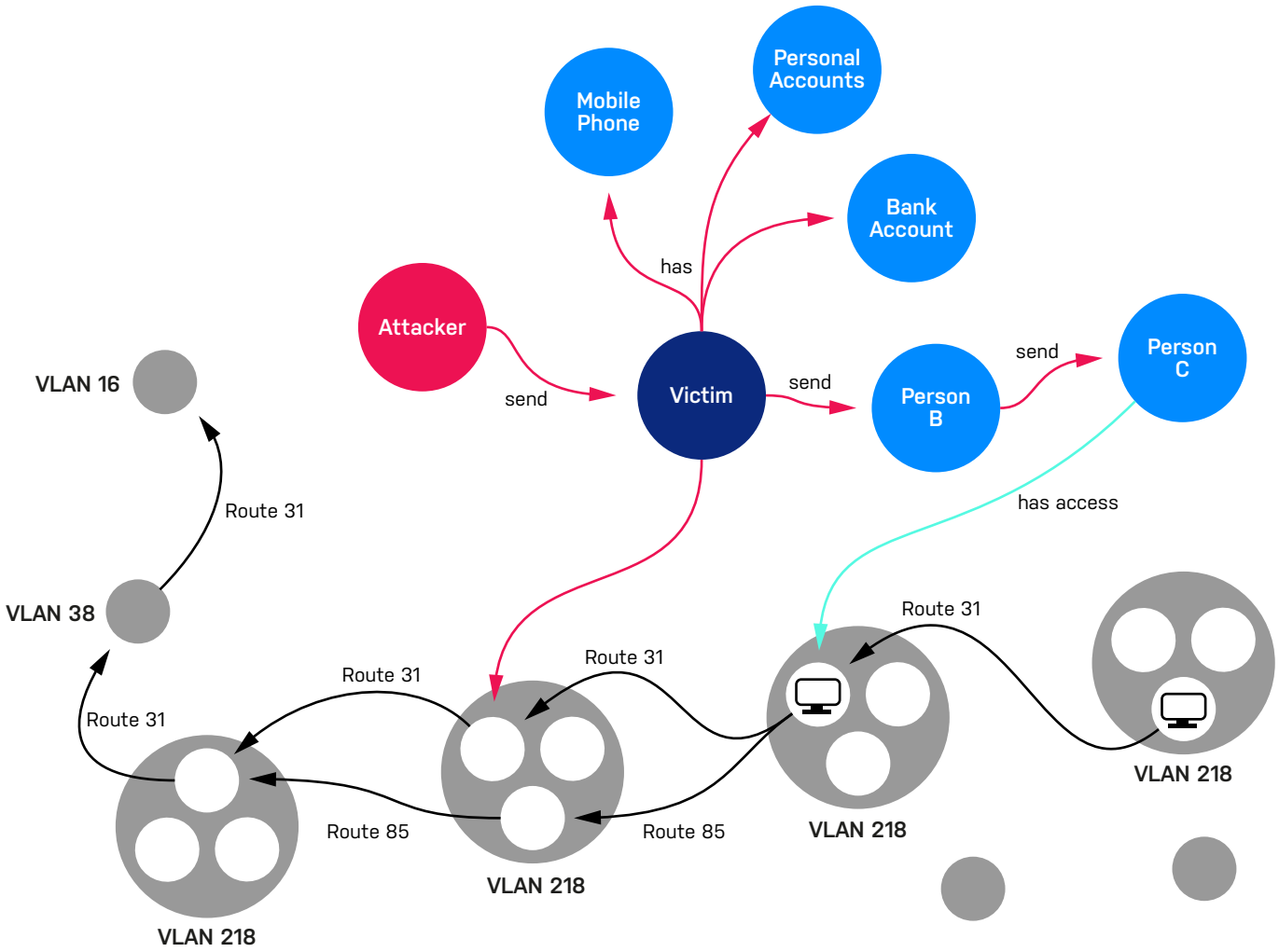
```
MATCH (compromisedSystem) <- [:DEPENDS_UPON*] - (system:System)
<- [:SUPPORTED_BY] - (team:Team)
RETURN team
```

Finding Compromised Systems after a Phishing Attack

A great strength of graph databases is the ability to explore connections deeply. For example, if we start with a phishing email, we can find all the emails that the user sent in that time, get the receiving accounts, and then get all the systems connected to see the possible extent of the phishing attack.

```
MATCH (phishingEmail) - [:WAS_SENT_TO] - (user:User)
    -[message:SENT_EMAIL_TO|SENT_SLACK_MESSAGE_TO|SENT_IM_TO*0..6]
    - (recipient) - [connection:CONNECTED_TO_SYSTEM]
    - (potentiallyCompromisedSystem: System)
WHERE connection.date > phishingEmail.sentDate
RETURN potentiallyCompromisedSystem
```

This query finds all **potentially impacted** entities that are connected to the compromised system by six hops or less.



Finding systems compromised by a phishing attack.

Grouping Incidents Using Community Detection

Security incidents can be loaded into a graph database and grouped as communities using community detection algorithms like Louvain. There are many cases where alerts might be related. They might indicate:

- The same attack such as a DDoS attack in progress
- The same event picked up by different detection mechanisms
- Multiple parts of one attack, like multiple hosts being probed by the same machine or phishing emails sent to several recipients

Given a graph database with a group of alerts, this Cypher code runs the Louvain graph algorithm to find communities.

```
CALL gds.graph.create({ 'communities', 'Alert',
       {
        relType: {
          type: '*',
          orientation: 'UNDIRECTED',
          properties: {}
        }
      }
    });
    CALL gds.louvain.write('communities',
      { writeProperty: 'communityId'}
    );
```

After running Louvain on the alerts, you can analyze the events further. We can create an `AlertGroup` object representing each community and link the alerts to their community using the `communityId` field that the algorithm populated:

```
MATCH (a:Alert)
MERGE (a) -> [:IN_GROUP] -> (ag:AlertGroup {name: a.communityId}
```

The graph now has the same objects as if we had run the following Cypher:

```
MERGE (g1:AlertGroup {name: '1'})
MERGE (g2:AlertGroup {name: '2'})
MERGE (:Alert {name: 'a', communityId: 2, from: 'evilcorp'}) –
[:IN_GROUP] -> (g2)
MERGE (:Alert {name: 'b',  communityId: 2}) - [:IN_GROUP] -> (g2)
MERGE (:Alert {name: 'c',  communityId: 2}) - [:IN_GROUP] -> (g2)
MERGE (:Alert {name: 'd',  communityId: 2}) - [:IN_GROUP] -> (g2)
MERGE (:Alert {name: 'e',  communityId: 1, from: 'nicecorp'})
- [:IN_GROUP] -> (g1)
```

Now we can use the information on the alerts to decide how much we trust the group as a whole, and write that information to the group:

```
MATCH (group:AlertGroup) - [:IN_GROUP] - (e:Alert {from: 'nicecorp'})
SET group.trustworthy='high';
MATCH (group:AlertGroup) - [:IN_GROUP] - (e:Alert {from: 'evilcorp'})
SET group.trustworthy='low';
```

To view this information:

```
MATCH (a:Alert) - [] - (ag:AlertGroup ) WHERE ag.trustworthy IS NOT NULL
RETURN a, ag
```

The complete code for this example (and all the examples in this paper) is available in its Github repository.

As we have seen, creating and analyzing a graph of your infrastructure – a digital twin – is one of the most effective measures you can take for improving your cybersecurity posture and managing the endless, dynamic complexity of cybersecurity vulnerabilities and threats.

Of course, when you create a digital twin of your infrastructure, you will find it serves many other purposes. Here are a few:

- Lending Tree sought to create a real-time view of its microservices and dependencies; soon they also used the tool to predict their monthly cloud usage
- Vanguard refactored their application from Java monoliths to microservices, mapping it all in Neo4j to improve software quality and enforce best practices
- The UK Department for Education uses Neo4j AuraDB to map and modernize its IT landscape, consolidating software licensing and paying for their subscription with their savings

Graphs eat complexity for breakfast, and there is no area more complex than the ever-morphing cybersecurity threats we face today.

Why Neo4j for Cybersecurity

The Neo4j Graph Data Platform offers a way to model, manage, and transform ever-changing cybersecurity landscapes.

The powerhouse at the core of the platform is the Neo4j Graph Database. Fully ACID compliant, Neo4j is a native graph database, optimized for storing data as a graph structure. Neo4j connects data as it is stored, enabling it to traverse connections orders of magnitude faster.

Neo4j provides security at the database level. Role-based access control empowers organizations to create rules about sensitive data and know that those rules will be applied across all applications and uses of Neo4j. Administrators declare role-based permissions at the schema level and all data is protected in accordance with those permissions.

Neo4j includes powerful developer tools to help you efficiently write, profile, and debug queries as well as visualize and explore your data. As we have seen, Cypher is simple and powerful, not to mention faster and far more compact than SQL. With the Neo4j GraphQL Library, you can generate Cypher from GraphQL, further empowering front-end developers. Connectors bring in streaming data from Kafka and Spark.

Neo4j Graph Data Science powers analysis and insights from graph datasets with billions of nodes. Neo4j Graph Data Science offers an enterprise-ready analytics workspace, more than 65 ready to run graph algorithms, and easy-to-use machine learning pipelines.

Neo4j Bloom enables novices and experts alike to visually investigate and explore their graph data from different perspectives. Rich graph visualizations drive understanding and alignment of technical and business staff, enabling rapid progress.

Storing your data in an enterprise-ready graph database like Neo4j empowers you to solve your most pressing problems and then expand your use case as needed. Answer any query in milliseconds, visualize your graph to see patterns, and use Neo4j Graph Data Science to identify anomalies and vulnerabilities proactively before attackers exploit them.

Get started right away with Neo4j or learn more about Neo4j for cybersecurity by attending a Connections event. or email us at cybersecurity@neotechnology.com.

Appendix: Open Source Tools and Github Repository

Loading Infrastructure Into Neo4j

Here are two open source tools that enable you to load your infrastructure into Neo4j.

Cartography is an open source tool for auditing cloud infrastructure. It scans the public clouds, Kubernetes setups, Github accounts, Okta installations, and PagerDuty and puts the information into a graph in Neo4j. If you are looking for a simple starting point, Cartography may be a good tool to help you load your organization's network into Neo4j.

Bloodhound is an auditing and examination solution that queries everything in your Active Directory domain and loads it into a Neo4j database. It includes prewritten queries to explore your domain and look for security weaknesses. For example, you can explore routes on your domain and find routes from nodes marked as compromised to the domain administrator nodes.

Loading Threat Intelligence Into Neo4j

The landscape of cybersecurity vulnerabilities, weaknesses, and attack patterns is continually evolving. MITRE and NIST both publish open data describing these items, and GraphKer is designed to make it easy to visualize and analyze this data.

GraphKer represents every public record of every CVE, CWE, CAPEC and CPE provided by MITRE and NIST and connects them using Neo4j.

This Python tool scrapes the MITRE and NIST websites and loads the data into Neo4j. The data is also regularly published as a Neo4j database backup file, which you can easily load into Neo4j. You can then quickly examine the data using an exploration tool like Neo4j Bloom or query it using Cypher.

You can link this data with the data in the graph you have describing your network. For example, you could link software running on your machines to the related vulnerability database entries, and update the data automatically so you can easily see the latest vulnerabilities and proactively apply relevant fixes.

Github Repository

If you would like to put the code in this paper to work, down load it from the Github repository for this paper at https://github.com/neo4j-examples/cybersecurity-demo