****

**Technology stack : cyber security with IBM Qradar**

**Project title : advanced technology rules creation**

**threat detection**

**Team ID : LTVIP2024TMID14864**

**Team size : 5**

**Team leader : sarakanam vijaya durga**

**Team member : sushma seekolu**

**Team member : yellem sarvani**

**Team member : peyyala Sandhya**

**Team member : varipilli renuka**

**College name : ASD government degree college for**

**women (a)**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SNO** | **TITLE** | **PAGE NO** |
| 1 | Introduction | 1 to 4 |
| 2 | Abstract | 4 to 5 |
| 3 | Empathy Map Canvas | 6 to 6 |
| 4 | Brainstorming and Idea Prioritization | 7 to 9 |
| 5 | Stage-1 | 10 to 35 |
| 6 | Report on practice website | 36 |
| 7 | Report on Main website | 36 |
| 8 | Stage-2 | 37 to 47 |
| 9 | Stage-3 | 48 |
| 10 | conclusion | 49 |
| 11 | Future scope | 50 |
| 12 | references | 51 |

**INTRODUCTION**

With identity-based attacks on the rise, today’s businesses require the ability to detect when attackers exploit, misuse, or steal enterprise identities. This need is particularly true as organizations race to adopt the public cloud, and both human and non-human identities continue to increase exponentially. Given the penchant for attackers to use credentials and leverage Active Directory (AD), it is now critical to detect identity-based activity.

Identity Threat Detection and Response (ITDR) is a new security category adjacent to [Endpoint Detection and Response (EDR)](https://www.sentinelone.com/cybersecurity-101/what-is-endpoint-detection-and-response-edr/?utm_type=e&utm_target=aud-826478938444:kwd-306103145962&utm_device=c&utm_medium=paid-search&utm_source=google-paid&utm_campaign=nl-bau-platform-pmax&gclid=cjwkcajwuj2xbha3eiwamvjkvejsffwlrnotm2wzabws6vfwxujshliebsff2rpvcnfkkjqhbg57arocis4qavd_bwe), Extended Detection and Response (XDR), Network Detection and Response (NDR), and other detection solutions. While some might want to ask whether the industry needs yet another acronym amid cybersecurity’s seemingly endless alphabet soup, ITDR fills a significant gap in the [identity security](https://www.sentinelone.com/cybersecurity-101/identity-security-what-it-is-why-its-so-important/?utm_type=e&utm_target=aud-826478938444:kwd-306103145962&utm_device=c&utm_medium=paid-search&utm_source=google-paid&utm_campaign=nl-bau-platform-pmax&gclid=cjwkcajwuj2xbha3eiwamvjkvejsffwlrnotm2wzabws6vfwxujshliebsff2rpvcnfkkjqhbg57arocis4qavd_bwe) landscape. ITDR differentiates itself from identity protection systems in that it focuses on protecting credentials, privileges, cloud entitlements, and the systems that manage them. It represents an important step forward, marking the introduction of a new category of security tools.



Understanding Today’s Threats

The threat to identities is genuine, and given the damages occurring with their misuse, it should be a priority for every CISO. According to the 2021 Verizon Data Breach Investigations Report, credential data now factors into 61% of all breaches. More broadly, the “human element” factor into 85% of breaches, while phishing is present in 36% of them. These stats highlight that attackers consistently attempt to access valid credentials and use them to move throughout networks undetected. Credential misuse has also enabled the growth of attack tactics like Ransomwar2.0, with ransomware now making up 10% of all breaches (double what it was in 2019).

**Abstract**

Threat intelligence is the provision of evidence-based knowledge about existing or potential threats. Benefits of threat intelligence include improved efficiency and effectiveness in security operations in terms of detective and preventive capabilities. Successful threat intelligence within the cyber domain demands a knowledge base of threat information and an expressive way to represent this knowledge. This purpose is served by the use of taxonomies, sharing standards, and ontologies. This paper introduces the Cyber Threat Intelligence (CTI) model, which enables cyber defenders to explore their threat intelligence capabilities and understand their position against the ever-changing cyber threat landscape. In addition, we use our model to analyze and evaluate several existing taxonomies, sharing standards, and ontologies relevant to cyber threat intelligence. Our results show that the cyber security community lacks an ontology covering the complete spectrum of threat intelligence. To conclude, we argue the importance of developing a multi-layered cyber threat intelligence ontology based on the CTI model and the steps should be taken under consideration, which are the foundation of our future work.

**Empathy Canvas Map**

Despite the existence of various simple and complex data-

gathering instruments based on direct observation, none of

them stands out from the rest, since each has a specific

purpose. It is most important to manage to empathize with

the needs and requirements of users; in other words, to

have the ability to stand in someone else’s shoes in order

to feel what the other person feels without losing one’s

own identity, searching for aspects that people have in

common to facilitate the interaction. And what turns out to

be the most enriching in the HCI field is for empathy to

allow for ‘experiencing’ and observing the other’s reality

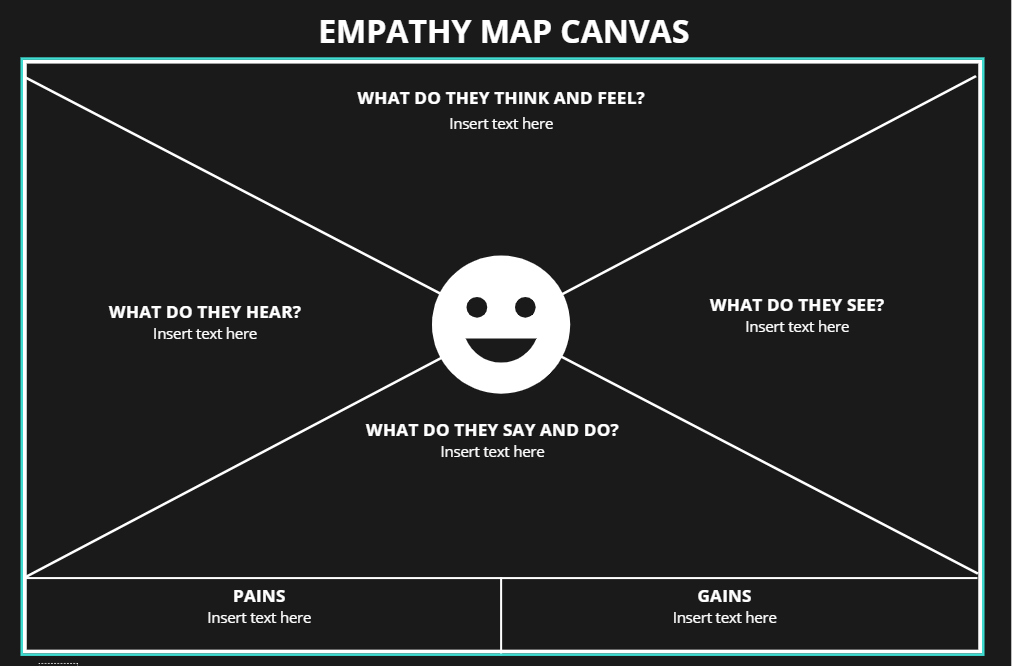
and get to know his feelings and perceptions.

Empathy map

The “Empathy Map” is an instrument that allows for

“standing in someone else’s shoes” based on 6 variables to

be considered in relation to the user: what he says, does,



**Brainstorming and Idea Prioritization**

### **1. Define goal and invite stakeholders**

The first step in the process is to clearly define the goal(s) of your brainstorming session. What problem(s) are you trying to solve? At this stage, you should also carefully select a list of stakeholders who will participate in your brainstorming session.

### **2. Define your problem statement**

The next phase is to create a problem statement that clearly identifies the issue. Frame your problem statement as a ‘How might we?’ question (e.g., ‘How might we shorten wait times for customer service calls?’).

### **3. Brainstorm!**

Have each stakeholder brainstorm as many ideas for possible solutions as they can in a given time.

Pro-tip: Use Mural’s private mode feature to avoid groupthink while brainstorming, allowing everyone to work independently even while synchronous.

### **4. Group ideas together by theme**

Once you’ve generated as many ideas as possible with your team, it’s time to look carefully at the results. What are the natural themes that emerge from your ideation? How should they be grouped together to inform potential solutions?

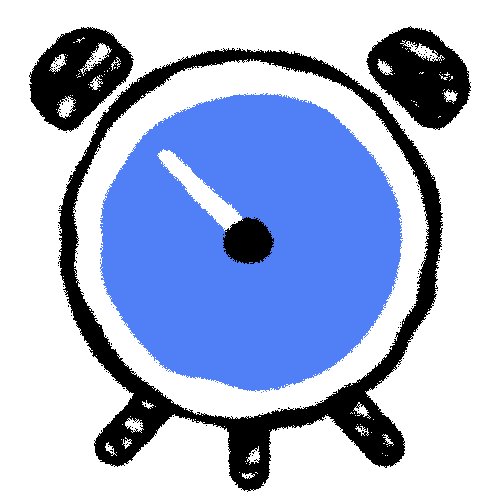
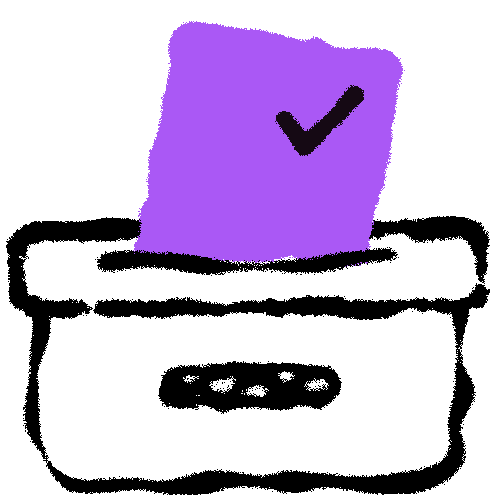
### **5. Prioritize solutions**

Now that you’ve refined your ideation into themes, you can effectively prioritize the results using the chart tracking importance vs. feasibility.



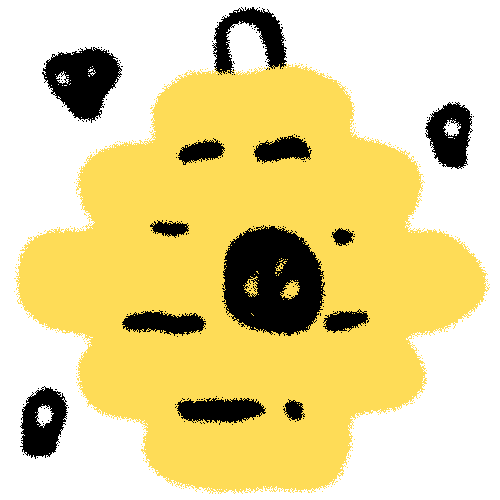
### **Sticky notes & text**

Add ideas, action items, and more as a sticky note or text box — then change the colors and cluster to identify patterns and new solutions.



### **Timer**

Keep collaboration moving forward with a timer to structure and time-box activities.

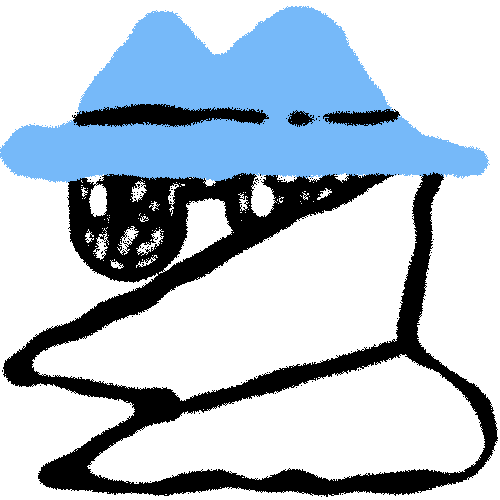


### **Real-time collaboration**

Add more productivity and engagement to meetings and calls with features to guide collaboration.

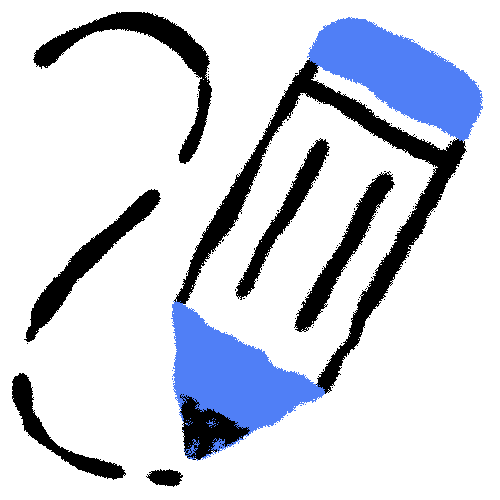
### **Anonymous voting**

Gain consensus and reach alignment quickly, either in real time or asynchronously.



### **Private mode**

Avoid groupthink and get authentic feedback by allowing collaborators to add content privately.



### **Drawing**

Visualize ideas, prototype, and quickly annotate for a better-than-a-whiteboard experience.

## 

## STAGE-1

## Title of the project:-advanced technologies rule creation for

## threat detection

## Overview:

The threat detection and response process includes:

* **Detection.** Security tools that monitor endpoints, identities, networks, apps, and clouds help surface risks and potential breaches. Security professionals also use [cyberthreat hunting](https://www.microsoft.com/en-us/security/business/security-101/what-is-cyber-threat-hunting" \t "_self) techniques to try to uncover emerging cyberthreats.
* **Investigation.** Once a risk is identified, people use AI and other tools to confirm the cyberthreat is real, determine how it happened, and assess what company assets are affected.
* **Containment.** To stop the spread of a cyberattack, cybersecurity teams isolate infected devices, identities, and networks from the rest of the organization’s assets.
* **Eradication.** Teams eliminate the root cause of a security incident with the goal of evicting the adversary completely from the environment and mitigating vulnerabilities that might put the organization at risk of a similar cyberattack.
* **Recovery.** After teams are reasonably confident that a cyberthreat or vulnerability has been removed, they bring any isolated systems back online.
* **Report.** Depending on the severity of the incident, security teams will document and brief leaders, executives, and/or the board on what happened and how it was resolved.

**Risk mitigation.**To prevent a similar breach from happening again and to improve response in the future, teams study the incident and identify changes to make to the environment and processes.

Successful insider threat programs proactively use a mitigation approach of detect and identify, assess, and manage to protect their organization. The foundation of the program’s success is the detection and identification of observable, concerning behaviors or activities.

Threat detection and identification is the process by which persons who might present an insider threat risk due to their observable, concerning behaviors come to the attention of an organization or insider threat team. Detecting and identifying potential insider threats requires both human and technological elements. An organization’s own personnel are an invaluable resource to observe behaviors of concern.

While virtually every person will experience stressful events, most do so without resorting to disruptive or destructive acts. For those insiders that turn to malicious activity, researchers have found that the acts are rarely spontaneous; instead, they are usually the result of a deliberate decision to act.

1. Behavior-based threat detection   
Behavior-based detection methods identify abnormal behavior that could indicate activity on devices or networks. With this threat detection model, baselines for normal behavior patterns, such as where a user usually logs in from, what time of day they are online, and what resources they access, are developed and regularly updated. If behavior deviates from established patterns, a flag is raised, sending an alert about potentially malicious activity.

2. Machine learning-based threat detection   
Vast amounts of data from a variety of sources, such as log files, security systems, and cloud services are processed through [machine learning](https://www.sailpoint.com/identity-library/how-ai-and-machine-learning-are-improving-cybersecurity/) models. Machine learning algorithms use statistics and probability to quickly recognize patterns that would be impossible for humans to detect. Generating insights from across the attack surface, machine learning threat detection plays a critical role in identifying unknown threats.

3. Signature-based threat detection   
Signature-based threat detection approaches scan network traffic for indicators of known threats (e.g., hashes, names of files, registry of key names, or strings that show up in a file). When a match is found, an alert is generated.

Threat detection and response is the practice of identifying any malicious activity that could compromise the network and then composing a proper response to mitigate or neutralize the threat before it can exploit any present [vulnerabilities](https://www.rapid7.com/fundamentals/vulnerabilities-exploits-threats/).

Within the context of an organization's security program, the concept of "threat detection" is multifaceted. Even the best security programs must plan for worst-case scenarios: when someone or something has slipped past their defensive and preventative technologies and becomes a threat.

Detection and response is where people join forces with technology to address a breach. A strong threat detection and response program combines people, processes, and technology to recognize signs of a breach as early as possible, and take appropriate actions.

### Detecting Threats

When it comes to detecting and mitigating threats, speed is crucial. Security programs must be able to detect threats quickly and efficiently so attackers don’t have enough time to root around in sensitive data. A business’s defensive programs can ideally stop a majority of previously seen threats, meaning they should know how to fight them.

These threats are considered "known" threats. However, there are additional “unknown” threats that an organization aims to detect. This means the organization hasn't encountered them before, perhaps because the attacker is using new methods or technologies.

Known threats can sometimes slip past even the best defensive measures, which is why most security organizations actively look for both known and unknown threats in their environment. So how can an organization try to detect both known and unknown threats?

## Leveraging Threat Intelligence

Threat intelligence is a way of looking at signature data from previously seen attacks and comparing it to enterprise data to identify threats. This makes it particularly effective at detecting known threats, but not unknown, threats. Known threats are those that are recognizable because the malware or attacker infrastructure has been identified as associated with malicious activity.

Unknown threats are those that haven't been identified in the wild (or are ever-changing), but threat intelligence suggests that threat actors are targeting a swath of vulnerable assets, weak credentials, or a specific industry vertical. [User behavior analytics (UBA)](https://www.rapid7.com/solutions/user-behavior-analytics/) are invaluable in helping to quickly identify anomalous behavior - possibly indicating an unknown threat - across your network. UBA tools establish a baseline for what is "normal" in a given environment, then leverage analytics (or in some cases, machine learning) to determine and alert when behavior is straying from that baseline.

[Attacker behavior analytics (ABA)](https://www.rapid7.com/solutions/attacker-behavior-analytics/) can expose the various tactics, techniques, and procedures (TTPs) by which attackers can gain access to your corporate network. TTPs include things like malware, cryptojacking (using your assets to mine cryptocurrency), and confidential data exfiltration.

During a breach, every moment an attacker is undetected is time for them to tunnel further into your environment. A combination of UBAs and ABAs offer a great starting point to ensure your [security operations center (SOC)](https://www.rapid7.com/fundamentals/security-operations-center/) is alerted to potential threats as early as possible in the [attack chain](https://www.rapid7.com/fundamentals/mitre-attack-framework/).

## Responding to Security Incidents

One of the most critical aspects to implementing a proper incident response framework is stakeholder buy-in and alignment, prior to launching the framework. No one likes surprises or questions-after-the-fact when important work is waiting to be done. Fundamental [incident response](https://www.rapid7.com/fundamentals/incident-response/) questions include:

* Do teams know who is responsible at each phase of incident response?
* Is the proper chain of communications well understood?
* Do team members know when and how to escalate issues as needed?

A great incident response plan and playbook minimizes the impact of a breach and ensures things run smoothly, even in a stressful breach scenario. If you're just getting started, some important considerations include:

* **Defining roles and duties for handling incidents**: These responsibilities, including contact information and backups, should be documented in a readily accessible channel.
* **Considering who to loop in**: Think beyond IT and security teams to document which cross-functional or third-party stakeholders – such as legal, PR, your board, or customers – should be looped in and when. Knowing who owns these various communications and how they should be executed will help ensure responses run smoothly and expectations are met along the way.

### Step 1: Scope your work

The first step in the threat modeling process is concerned with gaining an understanding of what you’re working on. This can involve:

* Drawing diagrams, often data flow diagrams.
* Identifying entry points to see where a potential attacker could interact with the application.
* Trying to identifying “assets”
* Identifying trust levels that represent the access rights that the application will grant to external entities.
* Reading a user story or creating one. (This step is sometimes extended to abuser stories, misuse cases, etc.)

Data flow diagrams ([DFDs](https://en.wikipedia.org/wiki/Data-flow_diagram)) are frequently used to show what we’re working on. The [DFDs](https://en.wikipedia.org/wiki/Data-flow_diagram) show the different paths through the system, highlighting the privilege or trust boundaries.

This is sometimes called “decompose the application,” which is an approach that consultants use when they’re brought in to do a threat model or architectural review. Consultants will often provide output in the form of a Threat Model document.

Tarandach has advocated for “Threat modeling every story,” which is work that developers, operations or SRE would do. When threat modeling every story, “decompose the application” is meaningless.

This step often calls out “assets”, which can be any of things you want to protect, stepping stones, or things attackers want. Often times those assets are out of scope for a project, and are a distraction. Other times, they’re hard to identify in advance of an attacker drawing attention to them.

### Step 2: Determine Threats

Critical to the identification of threats is using a threat categorization methodology. [STRIDE](https://en.wikipedia.org/wiki/STRIDE_%28security%29) is frequently used in threat modeling, and kill chains including MITRE ATT&CK are frequently used for operational threat modeling.

The goal of the threat categorization is to help identify threats both from the attacker ([STRIDE](https://en.wikipedia.org/wiki/STRIDE_%28security%29)). [DFDs](https://en.wikipedia.org/wiki/Data-flow_diagram) produced in step 1 help to identify the potential threat targets from the attacker’s perspective, such as data sources, processes, data flows, and interactions with users.

These threats can be organized further using threat trees; there is one tree for each threat goal. Common threat lists with examples can help in the identification of such threats. Use and abuse cases can illustrate how existing protective measures could be bypassed, or where a lack of such protection exists.

### Step 3: Determine Countermeasures and Mitigation

A vulnerability may be mitigated with the implementation of a countermeasure. Such countermeasures can be identified using threat-countermeasure mapping lists. Prioritization of countermeasures is a complex and contentious topic. Many approaches exist, and organizations need to select ones that will work for them. Frequently included factors are likelihood of attack, damage from an attack, and complexity or cost of fix.

The risk mitigation strategy might involve evaluating these threats from the business impact they pose. Once the possible impact is identified, options for addressing the risk include:

* Accept: decide that the business impact is acceptable, and document who has chosen to accept the risk
* Eliminate: remove components that make the vulnerability possible
* Mitigate: add checks or controls that reduce the risk impact, or the chances of its occurrence
* Transfer: Transfer risk to an insurer or customer.

The following sections examine these steps in depth and provide examples of the resulting threat model in a structured format.

### Step 4: Assess your work

First, determine if you’ve done the work. Are there records showing a diagram, a threats list and a control list.

## (Sample) Scope the work

This version frames the work as aimed at helping those not involved with the application’s development gain an understanding of the application and how it interacts with external entities. This goal is achieved by information gathering and documentation. The information gathering process is carried out using a clearly defined structure, which ensures the correct information is collected.

### (Sample) Threat Model Information

Waterfall-style or consultant-delivered threat model documents typically includes the the following:

1. **Application Name**: The name of the application examined.
2. **Application Version**: The version of the application examined.
3. **Description**: A high level description of the application.
4. **Document Owner**: The owner of the threat modeling document.
5. **Participants**: The participants involved in the threat modeling process for this application.
6. **Reviewer**: The reviewer(s) of the threat model.

Example:

## Threat Model Information (Sample)

**Application Version:** 1.0

**Description:** The college library website is the first implementation of a website to provide librarians and library patrons (students and college staff) with online services. As this is the first implementation of the website, the functionality will be limited. There will be three users of the application:

1. Students
2. Staff
3. Librarians

Staff and students will be able to log in and search for books, and staff members can request books. Librarians will be able to log in, add books, add users, and search for books.

**Document Owner:** David Lowry

**Participants:** David Rook

**Reviewer:** Eoin Keary

### External Dependencies (Sample)

External dependencies are items external to the code of the application that may pose a threat to the application. These items are typically still within the control of the organization, but possibly not within the control of the development team. The first area to consider when investigating external dependencies is the production environment and requirements.

It is useful to understand how the application is or is not intended to be run. For example, if the application is expected to be run on a server that has been hardened to the organization’s hardening standard and it is expected to sit behind a firewall, then this information should be documented in the external dependencies section.

External dependencies should be documented as follows:

1. **ID**: A unique ID assigned to the external dependency.
2. **Description**: A textual description of the external dependency.

Example:

## External Dependencies (Sample)

| **ID** | **Description** |
| --- | --- |
| 1 | The college library website will run on a Linux server running Apache. This server will be hardened per the college’s server hardening standard. This includes the installation of the latest operating system and application security patches. |
| 2 | The database server will be MySQL and it will run on a Linux server. This server will be hardened per the college’s server hardening standard. This will include the installation of the latest operating system and application security patches. |
| 3 | The connection between the web server and the database server will be over a private network. |
| 4 | The web server is behind a firewall and the only communication available is TLS. |

### Entry Points

Entry points define the interfaces through which potential attackers can interact with the application or supply it with data. In order for a potential attacker to attack an application, entry points must exist. Entry points in an application can be layered. For example, each web page in a web application may contain multiple entry points.

Entry points show where data enters the system (i.e. input fields, methods) and exit points are where it leaves the system (i.e. dynamic output, methods), respectively. Entry and exit points define a trust boundary (see [Trust Levels](https://owasp.org/www-community/Threat_Modeling_Process#trust-levels)).

Entry points should be documented as follows:

1. **ID**: A unique ID assigned to the entry point. This will be used to cross-reference the entry point with any threats or vulnerabilities that are identified. In the case of layered entry points, a major.minor notation should be used.
2. **Name**: A descriptive name identifying the entry point and its purpose.
3. **Description**: A textual description detailing the interaction or processing that occurs at the entry point.
4. **Trust Levels**: The level of access required at the entry point. These will be cross-referenced with the trust levels defined later in the document.

Example:

## Entry Points

| **ID** | **Name** | **Description** | **Trust Levels** |
| --- | --- | --- | --- |
| 1 | HTTPS Port | The college library website will be only be accessible via TLS. All pages within the college library website are layered on this entry point. | (1) Anonymous Web User (2) User with Valid Login Credentials (3) User with Invalid Login Credentials (4) Librarian |
| 1.1 | Library Main Page | The splash page for the college library website is the entry point for all users. | (1) Anonymous Web User (2) User with Valid Login Credentials (3) User with Invalid Login Credentials (4) Librarian |
| 1.2 | Login Page | Students, faculty members and librarians must log in to the college library website before they can carry out any of the use cases. | (1) Anonymous Web User (2) User with Login Credentials (3) User with Invalid Login Credentials (4) Librarian |
| 1.2.1 | Login Function | The login function accepts user supplied credentials and compares them with those in the database. | (2) User with Valid Login Credentials (3) User with Invalid Login Credentials (4) Librarian |
| 1.3 | Search Entry Page | The page used to enter a search query. | (2) User with Valid Login Credentials (4) Librarian |

### Exit Points

Exit points might prove useful when attacking the client: for example, cross-site-scripting vulnerabilities and information disclosure vulnerabilities both require an exit point for the attack to complete.

In the case of exit points from components handling confidential data (e.g. data access components), exit points lacking security controls to protect confidentiality and integrity can lead to disclosure of such confidential information to an unauthorized user.

In many cases threats enabled by exit points are related to the threats of the corresponding entry point. In the login example, error messages returned to the user via the exit point (the log in page) might allow for entry point attacks, such as account harvesting (e.g. username not found), or SQL injection (e.g. SQL exception errors).

### Assets

Many have something that one or more attackers are interested in; these items or areas of interest are often labelled “assets.” Assets can be both physical assets and abstract assets. For example, an asset of an application might be a list of clients and their personal information; this is a physical asset. An abstract asset might be the reputation of an organization. It is unclear how talking about the reputation of the organization aids in threat modeling.

Assets are documented in this sample threat model as follows:

1. **ID**: A unique ID is assigned to identify each asset. This will be used to cross-reference the asset with any threats or vulnerabilities that are identified.
2. **Name**: A descriptive name that clearly identifies the asset.
3. **Description**: A textual description of what the asset is and why it needs to be protected.
4. **Trust Levels**: The level of access required to access the entry point is documented here. These will be cross-referenced with the trust levels defined in the next step.

Example:

## Assets

| **ID** | **Name** | **Description** | **Trust Levels** |
| --- | --- | --- | --- |
| **1** | **Library Users and Librarian** | **Assets relating to students, faculty members, and librarians.** |  |
| 1.1 | User Login Details | The login credentials that a student or a faculty member will use to log into the College Library website. | (2) User with Valid Login Credentials (4) Librarian (5) Database Server Administrator (7) web server User Process (8) Database Read User (9) Database Read/Write User |
| 1.2 | Librarian Login Details | The login credentials that a Librarian will use to log into the College Library website. | (4) Librarian (5) Database Server Administrator (7) web server User Process (8) Database Read User (9) Database Read/Write User |
| 1.3 | Personal Data | The College Library website will store personal information relating to the students, faculty members, and librarians. | (4) Librarian (5) Database Server Administrator (6) Website Administrator (7) web server User Process (8) Database Read User (9) Database Read/Write User |
| **2** | **System** | **Assets relating to the underlying system.** |  |
| 2.1 | Availability of College Library Website | The College Library website should be available 24 hours a day and can be accessed by all students, college faculty members, and librarians. | (5) Database Server Administrator (6) Website Administrator |
| 2.2 | Ability to Execute Code as a web server User | This is the ability to execute source code on the web server as a web server user. | (6) Website Administrator (7) web server User Process |
| 2.3 | Ability to Execute SQL as a Database Read User | This is the ability to execute SQL select queries on the database, and thus retrieve any information stored within the College Library database. | (5) Database Server Administrator (8) Database Read User (9) Database Read/Write User |
| 2.4 | Ability to Execute SQL as a Database Read/Write User | This is the ability to execute SQL. Select, insert, and update queries on the database and thus have read and write access to any information stored within the College Library database. | (5) Database Server Administrator (9) Database Read/Write User |
| **3** | **Website** | **Assets relating to the College Library website.** |  |
| 3.1 | Login Session | This is the login session of a user to the College Library website. This user could be a student, a member of the college faculty, or a Librarian. | (2) User with Valid Login Credentials (4) Librarian |
| 3.2 | Access to the Database Server | Access to the database server allows you to administer the database, giving you full access to the database users and all data contained within the database. | (5) Database Server Administrator |
| 3.3 | Ability to Create Users | The ability to create users would allow an individual to create new users on the system. These could be student users, faculty member users, and librarian users. | (4) Librarian (6) Website Administrator |
| 3.4 | Access to Audit Data | The audit data shows all audit-able events that occurred within the College Library application by students, staff, and librarians. | (6) Website Administrator |

### Trust Levels

Trust levels represent the access rights that the application will grant to external entities. The trust levels are cross-referenced with the entry points and assets. This allows us to define the access rights or privileges required at each entry point, and those required to interact with each asset.

Trust levels are documented in the threat model as follows:

1. **ID**: A unique number is assigned to each trust level. This is used to cross-reference the trust level with the entry points and assets.
2. **Name**: A descriptive name that allows you to identify the external entities that have been granted this trust level.
3. **Description**: A textual description of the trust level detailing the external entity who has been granted the trust level.

Example:

## Trust Levels

| **ID** | **Name** | **Description** |
| --- | --- | --- |
| 1 | Anonymous Web User | A user who has connected to the college library website but has not provided valid credentials. |
| 2 | User with Valid Login Credentials | A user who has connected to the college library website and has logged in using valid login credentials. |
| 3 | User with Invalid Login Credentials | A user who has connected to the college library website and is attempting to log in using invalid login credentials. |
| 4 | Librarian | The librarian can create users on the library website and view their personal information. |
| 5 | Database Server Administrator | The database server administrator has read and write access to the database that is used by the college library website. |
| 6 | Website Administrator | The Website administrator can configure the college library website. |
| 7 | web server User Process | This is the process/user that the web server executes code as and authenticates itself against the database server as. |
| 8 | Database Read User | The database user account used to access the database for read access. |
| 9 | Database Read/Write User | The database user account used to access the database for read and write access. |

### Data Flow Diagrams

All of the information collected allows us to accurately model the application through the use of Data Flow Diagrams ([DFDs](https://en.wikipedia.org/wiki/Data-flow_diagram)). The [DFDs](https://en.wikipedia.org/wiki/Data-flow_diagram) will allow us to gain a better understanding of the application by providing a visual representation of how the application processes data.

Data flows show how data flows logically through the application, end to end. They allow the identification of affected components through critical points (e.g. data entering or leaving the system, storage of data) and the flow of control through these components.

The focus of the [DFDs](https://en.wikipedia.org/wiki/Data-flow_diagram) is on how data moves through the application and what happens to the data as it moves. [DFDs](https://en.wikipedia.org/wiki/Data-flow_diagram) are hierarchical in structure, so they can be used to decompose the application into subsystems and lower-level subsystems. The high-level [DFD](https://en.wikipedia.org/wiki/Data-flow_diagram) will allow us to clarify the scope of the application being modeled. The lower level iterations will allow us to focus on the specific processes involved when processing specific data.

There are a number of symbols that are used in [DFDs](https://en.wikipedia.org/wiki/Data-flow_diagram) for threat modeling. These are described below:

| **Symbol** | **Name** | **Description** |
| --- | --- | --- |
| Data Flow Diagram: External Entity | External Entity | The external entity shape is used to represent any entity outside the application that interacts with the application via an entry point. |
| Data Flow Diagram: Process | Process | The process shape represents a task that handles data within the application. The task may process the data or perform an action based on the data. |
| Data Flow Diagram: Multiple Process | Multiple Process | The multiple process shape is used to present a collection of subprocesses. The multiple process can be broken down into its subprocesses in another DFD. |
| Data Flow Diagram: Data Store | Data Store | The data store shape is used to represent locations where data is stored. Data stores do not modify the data, they only store data. |
| Data Flow Diagram: Data Flow | Data Flow | The data flow shape represents data movement within the application. The direction of the data movement is represented by the arrow. |
| Data Flow Diagram: Privilege Boundary | Privilege Boundary | The privilege boundary (or trust boundary) shape is used to represent the change of trust levels as the data flows through the application. Boundaries show any location where the level of trust changes. |

### Example Diagrams

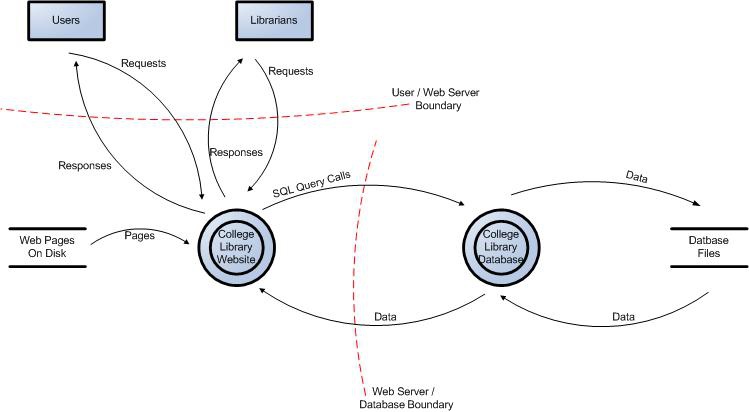


Figure 1: Data Flow Diagram for the College Library Website.

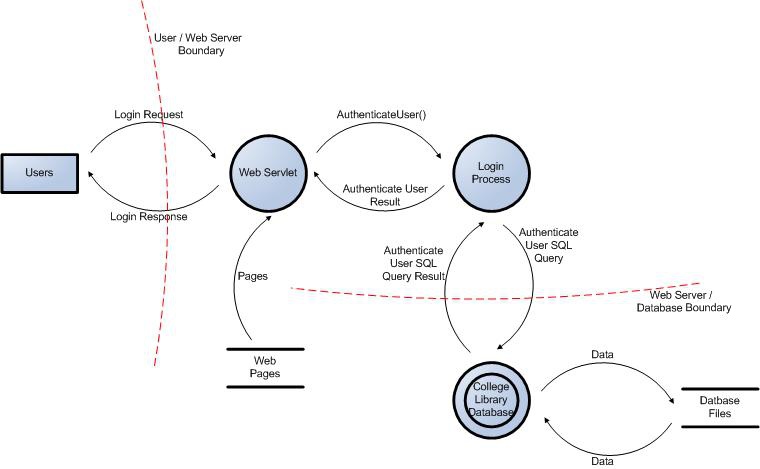


Figure 2: User Login Data Flow Diagram for the College Library Website.

## Determine Threats (Sample)

### STRIDE

A threat mnemonic such as [STRIDE](https://en.wikipedia.org/wiki/STRIDE_%28security%29) is useful in the identification of threats by prompting us to think about attacker steps such as:

* **S**poofing
* **T**ampering
* **R**epudiation
* **I**nformation Disclosure
* **D**enial of Service
* **E**levation of Privilege

A list of generic threats using [STRIDE](https://en.wikipedia.org/wiki/STRIDE_%28security%29) is provided in the following table along with their security controls:

## STRIDE Threat List

| **Type** | **Description** | **Security Control** |
| --- | --- | --- |
| Spoofing | Threat action aimed at accessing and use of another user’s credentials, such as username and password. | Authentication |
| Tampering | Threat action intending to maliciously change or modify persistent data, such as records in a database, and the alteration of data in transit between two computers over an open network, such as the Internet. | Integrity |
| Repudiation | Threat action aimed at performing prohibited operations in a system that lacks the ability to trace the operations. | Non-Repudiation |
| Information disclosure | Threat action intending to read a file that one was not granted access to, or to read data in transit. | Confidentiality |
| Denial of service | Threat action attempting to deny access to valid users, such as by making a web server temporarily unavailable or unusable. | Availability |
| Elevation of privilege | Threat action intending to gain privileged access to resources in order to gain unauthorized access to information or to compromise a system. | Authorization |

Threat lists based on the [STRIDE](https://en.wikipedia.org/wiki/STRIDE_%28security%29) model are useful in the identification of threats with regards to the attacker goals. For example, if the threat scenario is attacking the login, would the attacker brute force the password to break the authentication? If the threat scenario is to try to elevate privileges to gain another user’s privileges, would the attacker try to perform [forced browsing](https://owasp.org/www-community/attacks/Forced_browsing)?

### Threat Analysis

It is frequently claimed that “a prerequisite in the analysis of threats is the understanding of the generic definition of risk.” But this is not the case. You can analyze what can go wrong without being able to measure or quantify it.

Threat analysis is the identification of threats to the application, and involves the analysis of each aspect of the application’s functionality, architecture, and design. It is important to identify and classify potential weaknesses that could lead to an exploit.

From the defensive perspective, the identification of threats driven by security control categorization allows a threat analyst to focus on specific vulnerabilities. Typically, the process of threat identification involves going through iterative cycles where initially all the possible threats in the threat list that apply to each component are evaluated.

At the next iteration, threats are further analyzed by exploring the attack paths, the root causes for the threat to be exploited (e.g. vulnerabilities, depicted as orange blocks below), and the necessary mitigation controls (e.g. countermeasures, depicted as green blocks below). A threat tree as shown below is useful to perform such threat analysis.

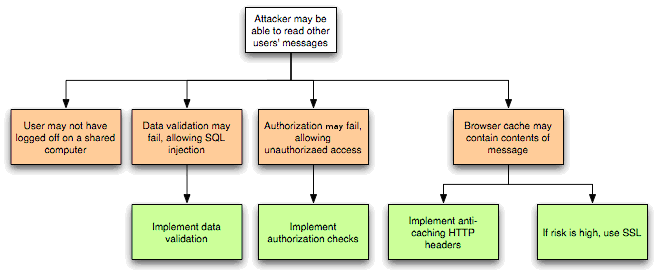


Figure 3: Threat Tree Diagram.

Once common threats, vulnerabilities, and attacks are assessed, a more focused threat analysis should take into consideration use and abuse cases. By thoroughly analyzing the use scenarios, weaknesses can be identified that could lead to the realization of a threat. Abuse cases should also be identified. These abuse cases can illustrate how existing protective measures could be bypassed, or where a lack of such protection exists.

A use and abuse case graph for authentication is shown below:

Figure 4: Use and Misuse Cases

The outcome of threat analysis is a determination of the types of threats posed to each component of the decomposed system. This can be documented using a threat categorization such as [STRIDE](https://en.wikipedia.org/wiki/STRIDE_%28security%29) or [ASF](https://pathlock.com/learn/what-are-application-security-frameworks/), the use of threat trees to determine how the threat can be exposed by a vulnerability, and use and misuse cases to further validate the lack of any countermeasures for mitigation.

### Ranking of Threats

Threats can be ranked from the perspective of risk factors. By determining the risk factor posed by the various identified threats, it is possible to create a prioritized list of threats to support a risk mitigation strategy, such as prioritizing the threats to be mitigated first. Different risk factors can be used to rank threats as High, Medium, or Low risk. In general, threat risk models use different factors to model risks such as those shown below:

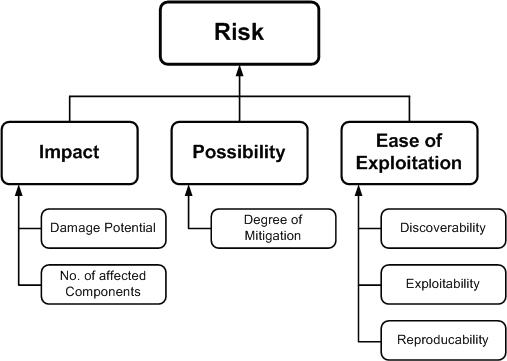


Figure 5: Ranking Risk Factors.

### Qualitative Risk Model

The following is a set of considerations for determining ease of exploitation:

1. Can an attacker exploit this remotely?
2. Does the attacker need to be authenticated?
3. Can the exploit be automated?

The impact mainly depends on the damage potential and its extent, such as the number of components that may be affected by a threat.

Questions to help determine the damage potential are:

1. Can an attacker completely take over and manipulate the system?
2. Can an attacker gain administration access to the system?
3. Can an attacker crash the system?
4. Can the attacker obtain access to sensitive information such as secrets or PII?

Questions to help determine the number of components that are affected by a threat:

1. How many connected data sources and systems can be impacted?
2. How many layers into infrastructure components can the threat agent traverse?

These examples help in the calculation of the overall risk values by assigning qualitative values such as High, Medium and Low to the likelihood and impact factors. In this case, using qualitative values, rather than numeric ones like in the case of the [DREAD](https://en.wikipedia.org/wiki/DREAD_(risk_assessment_model)) model, help avoid the ranking becoming overly subjective.

## Determine Countermeasures and Mitigation

The purpose of countermeasure identification is to determine if there is some kind of protective measure (e.g. security control, policies) that can prevent a threat from being realized. Vulnerabilities are then those threats that have no countermeasures. When threats have been categorized either with [STRIDE](https://en.wikipedia.org/wiki/STRIDE_%28security%29) or [ASF](https://pathlock.com/learn/what-are-application-security-frameworks/), it is possible to find appropriate countermeasures within the given category.

Provided below is a brief and limited checklist which is by no means an exhaustive list for identifying countermeasures for specific threats.

### STRIDE Threat & Mitigation Techniques

| **Threat Type** | **Mitigation Techniques** |
| --- | --- |
| Spoofing Identity | 1. Appropriate authentication 2. Protect secret data 3. Don’t store secrets |
| Tampering with data | 1. Appropriate authorization 2. Hashes 3. MACs 4. Digital signatures 5. Tamper resistant protocols |
| Repudiation | 1. Digital signatures 2. Timestamps 3. Audit trails |
| Information Disclosure | 1. Authorization 2. Privacy-enhanced protocols 3. Encryption 4. Protect secrets 5. Don’t store secrets |
| Denial of Service | 1. Appropriate authentication 2. Appropriate authorization 3. Filtering 4. Throttling 5. Quality of service |
| Elevation of privilege | 1. Run with least privilege |

Once threats and corresponding countermeasures are identified, it is possible to derive a threat profile with the following criteria:

1. **Non mitigated threats:** Threats which have no countermeasures and represent vulnerabilities that can be fully exploited and cause an impact.
2. **Partially mitigated threats:** Threats partially mitigated by one or more countermeasures and can only partially be exploited to cause a limited impact.
3. **Fully mitigated threats:** These threats have appropriate countermeasures in place and do not expose vulnerabilities.

## Complementing Code Review

Threat modeling is not an approach to reviewing code, but it does complement the security code review process. The inclusion of threat modeling early on in the Software Development Life Cycle ([SDLC](https://en.wikipedia.org/wiki/Systems_development_life_cycle)) can help to ensure that applications are being developed with appropriate security threat mitigations from the very beginning. This, combined with the documentation produced as part of the threat modeling process, can give code reviewers a greater understanding of the system. This allows the reviewer to see where the entry points to the application are and the associated threats with each entry point.

When source code analysis is performed outside the [SDLC](https://en.wikipedia.org/wiki/Systems_development_life_cycle), such as on existing applications, threat modeling helps to clarify the complexity of source code analysis. It promotes a depth-first approach instead of a breadth-first approach. In other words, instead of reviewing all source code with equal focus, you can prioritize the security code review of components where the threat modelling indicates higher-risk threats.

**STAGE-2**

You can enable Threat intelligence-based filtering for your firewall to alert and deny traffic from/to known malicious IP addresses, FQDNs, and URLs. The IP addresses, domains and URLs are sourced from the Microsoft Threat Intelligence feed, which includes multiple sources including the Microsoft Cyber Security team. [Intelligent Security Graph](https://www.microsoft.com/security/operations/intelligence) powers Microsoft threat intelligence and uses multiple services including Microsoft Defender for Cloud.

If you've enabled threat intelligence-based filtering, the firewall processes the associated rules before any of the NAT rules, network rules, or application rules.

When a rule triggers, you can choose to just log an alert, or you can choose alert and deny mode.

By default, threat intelligence-based filtering is in alert mode. You can’t turn off this feature or change the mode until the portal interface becomes available in your region.

You can define allowlists so threat intelligence doesn't filter traffic to any of the listed FQDNs, IP addresses, ranges, or subnets.

For a batch operation, you can upload a CSV file with list of IP addresses, ranges, and subnets.

## Logs

The following log excerpt shows a triggered rule:

JSONCopy

{

"category": "AzureFirewallNetworkRule",

"time": "2018-04-16T23:45:04.8295030Z",

"resourceId": "/SUBSCRIPTIONS/{subscriptionId}/RESOURCEGROUPS/{resourceGroupName}/PROVIDERS/MICROSOFT.NETWORK/AZUREFIREWALLS/{resourceName}",

"operationName": "AzureFirewallThreatIntelLog",

"properties": {

"msg": "HTTP request from 10.0.0.5:54074 to somemaliciousdomain.com:80. Action: Alert. ThreatIntel: Bot Networks"

}

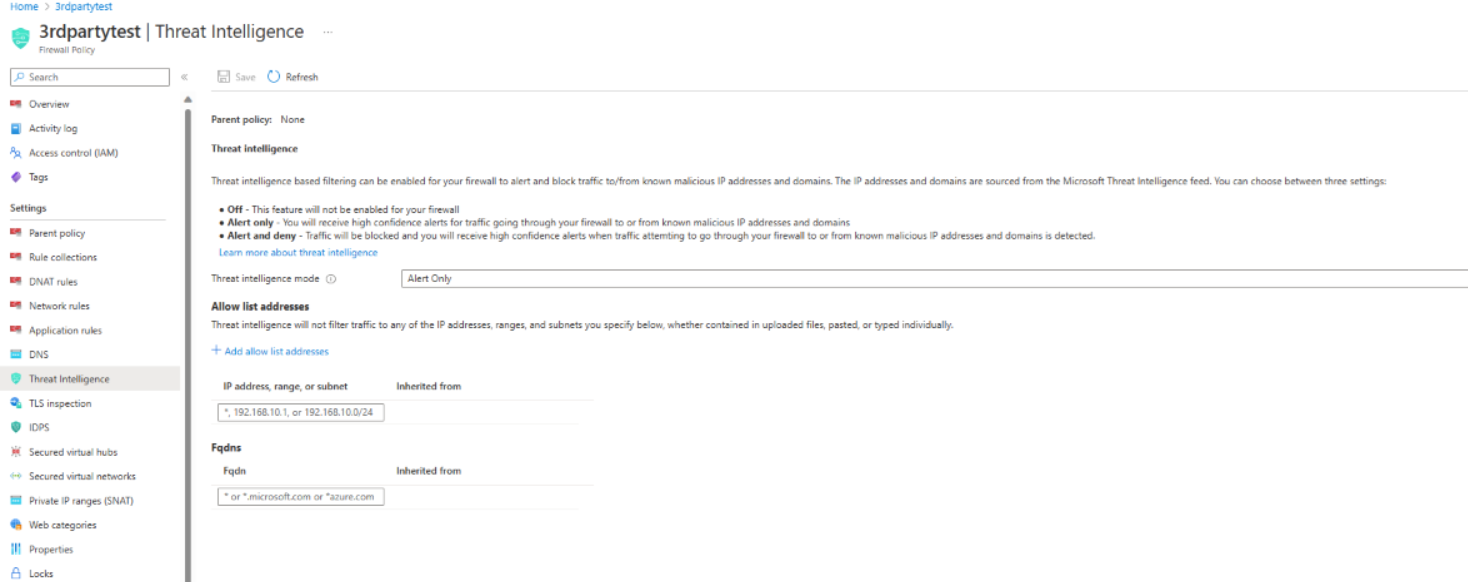
}

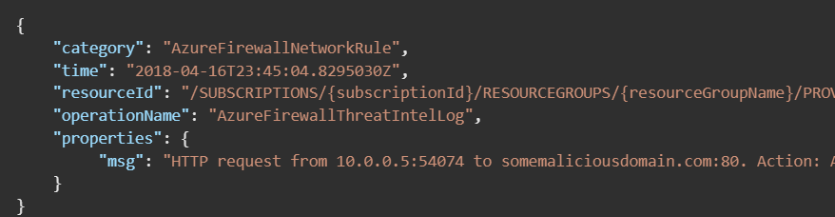
## Testing

* **Outbound testing** - Outbound traffic alerts should be a rare occurrence, as it means that your environment is compromised. To help test outbound alerts are working, a test FQDN exists that triggers an alert. Use testmaliciousdomain.eastus.cloudapp.azure.com for your outbound tests.

To prepare for your tests and to ensure you don't get a DNS resolution failure, configure the following items:

* + Add a dummy record to the hosts file on your test computer. For example, on a computer running Windows, you could add 1.2.3.4 testmaliciousdomain.eastus.cloudapp.azure.com to the C:\Windows\System32\drivers\etc\hosts file.
  + Ensure that the tested HTTP/S request is allowed using an application rule, not a network rule.
* **Inbound testing** - You can expect to see alerts on incoming traffic if the firewall has DNAT rules configured. You'll see alerts even if the firewall only allows specific sources on the DNAT rule and traffic is otherwise denied. Azure Firewall doesn't alert on all known port scanners; only on scanners that also engage in malicious activity.





## Rule processing using classic rules

Rule collections are processed according to the rule type in priority order, lower numbers to higher numbers from 100 to 65,000. A rule collection name can have only letters, numbers, underscores, periods, or hyphens. It must begin with a letter or number, and end with a letter, number, or underscore. The maximum name length is 80 characters.

It's best to initially space your rule collection priority numbers in 100 increments (100, 200, 300, and so on) so you have room to add more rule collections if needed.

## Rule processing using Firewall Policy

With Firewall Policy, rules are organized inside Rule Collections and Rule Collection Groups. Rule Collection Groups contain zero or more Rule Collections. Rule Collections are type NAT, Network, or Applications. You can define multiple Rule Collection types within a single Rule Group. You can define zero or more Rules in a Rule Collection. Rules in a Rule Collection must be of the same type (NAT, Network, or Application).

Rules are processed based on Rule Collection Group Priority and Rule Collection priority. Priority is any number between 100 (highest priority) to 65,000 (lowest priority). Highest priority Rule Collection Groups are processed first. Inside a rule collection group, Rule Collections with highest priority (lowest number) are processed first.

If a Firewall Policy is inherited from a parent policy, Rule Collection Groups in the parent policy always takes precedence regardless of the priority of a child policy.

**Note**

Application rules are always processed after Network rules, which are processed after DNAT rules regardless of Rule collection group or Rule collection priority and policy inheritance.

Here's an example policy:

Assuming BaseRCG1 is a rule collection group priority (200) that contains the rule collections: DNATRC1, DNATRC3,NetworkRC1.  
BaseRCG2 is a rule collection group priority (300) that contains the rule collections: AppRC2, NetworkRC2.  
ChildRCG1 is a rule collection group priority (200) that contains the rule collections: ChNetRC1, ChAppRC1.  
ChildRCG2 is a rule collection group that contains the rule collections: ChNetRC2, ChAppRC2,ChDNATRC3.

As per following table:

Expand table

| **Name** | **Type** | **Priority** | **Rules** | **Inherited from** |
| --- | --- | --- | --- | --- |
| BaseRCG1 | Rule collection group | 200 | 8 | Parent policy |
| DNATRC1 | DNAT rule collection | 600 | 7 | Parent policy |
| DNATRC3 | DNAT rule collection | 610 | 3 | Parent policy |
| NetworkRC1 | Network rule collection | 800 | 1 | Parent policy |
| BaseRCG2 | Rule collection group | 300 | 3 | Parent policy |
| AppRC2 | Application rule collection | 1200 | 2 | Parent policy |
| NetworkRC2 | Network rule collection | 1300 | 1 | Parent policy |
| ChildRCG1 | Rule collection group | 300 | 5 | - |
| ChNetRC1 | Network rule collection | 700 | 3 | - |
| ChAppRC1 | Application rule collection | 900 | 2 | - |
| ChildRCG2 | Rule collection group | 650 | 9 | - |
| ChNetRC2 | Network rule collection | 1100 | 2 | - |
| ChAppRC2 | Application rule collection | 2000 | 7 | - |
| ChDNATRC3 | DNAT rule collection | 3000 | 2 | - |

Initial Processing:

The process begins by examining the rule collection group (RCG) with the lowest number, which is BaseRCG1 with a priority of 200. Within this group, it searches for DNAT rule collections and evaluates them according to their priorities. In this case, DNATRC1 (priority 600) and DNATRC3 (priority 610) are found and processed accordingly.  
Next, it moves to the next RCG, BaseRCG2 (priority 200), but finds no DNAT rule collection.  
Following that, it proceeds to ChildRCG1 (priority 300), also without a DNAT rule collection.  
Finally, it checks ChildRCG2 (priority 650) and finds the ChDNATRC3 rule collection (priority 3000).

Iteration Within Rule Collection Groups:

Returning to BaseRCG1, the iteration continues, this time for NETWORK rules. Only NetworkRC1 (priority 800) is found.  
Then, it moves to BaseRCG2, where NetworkRC2 (priority 1300) is located.  
Moving on to ChildRCG1, it discovers ChNetRC1 (priority 700) as the NETWORK rule.  
Lastly, in ChildRCG2, it finds ChNetRC2 (priority 1100) as the NETWORK rule collection.

Final Iteration for APPLICATION Rules:

Returning to BaseRCG1, the process iterates for APPLICATION rules, but none are found.  
In BaseRCG2, it identifies AppRC2 (priority 1200) as the APPLICATION rule.  
In ChildRCG1, ChAppRC1 (priority 900) is found as the APPLICATION rule.  
Finally, in ChildRCG2, it locates ChAppRC2 (priority 2000) as the APPLICATION rule.

**In summary, the rule processing sequence is as follows: DNATRC1, DNATRC3, ChDNATRC3, NetworkRC1, NetworkRC2, ChNetRC1, ChNetRC2, AppRC2, ChAppRC1, ChAppRC2.**

This process involves analyzing rule collection groups by priority, and within each group, ordering the rules according to their priorities for each rule type (DNAT, NETWORK, and APPLICATION).

So first all the DNAT rules are processed from all the rule collection groups, analysing the rule collection groups by order of priority and ordering the DNAT rules within each rule collection group by order of priority. Then the same process for NETWORK rules, and finally for APPLICATION rules.

For more information about Firewall Policy rule sets, see [Azure Firewall Policy rule sets](https://learn.microsoft.com/en-us/azure/firewall/policy-rule-sets).

### Threat Intelligence

If you enable threat intelligence-based filtering, those rules are highest priority and are always processed first (before network and application rules). Threat-intelligence filtering may deny traffic before any configured rules are processed. For more information, see [Azure Firewall threat intelligence-based filtering](https://learn.microsoft.com/en-us/azure/firewall/threat-intel).

### IDPS

When IDPS is configured in Alert mode, the IDPS engine works in parallel to the rule processing logic and generates alerts on matching signatures for both inbound and outbound flows. For an IDPS signature match, an alert is logged in firewall logs. However, since the IDPS engine works in parallel to the rule processing engine, traffic denied or allowed by application/network rules may still generate another log entry.

When IDPS is configured in Alert and Deny mode, the IDPS engine is inline and activated after the rules processing engine. So both engines generate alerts and may block matching flows.

Session drops done by IDPS blocks the flow silently. So no RST is sent on the TCP level. Since IDPS inspects traffic always after the Network/Application rule has been matched (Allow/Deny) and marked in logs, another Drop message may be logged where IDPS decides to deny the session because of a signature match.

When TLS inspection is enabled both unencrypted and encrypted traffic is inspected.

## Outbound connectivity

### Network rules and applications rules

If you configure network rules and application rules, then network rules are applied in priority order before application rules. The rules are terminating. So, if a match is found in a network rule, no other rules are processed. If configured, IDPS is done on all traversed traffic and upon signature match, IDPS may alert or/and block suspicious traffic.

Application rules then evaluate the packet in priority order if there's no network rule match, and if the protocol is HTTP, HTTPS, or MSSQL.

For HTTP, Azure Firewall looks for an application rule match according to the Host header. For HTTPS, Azure Firewall looks for an application rule match according to SNI only.

In both HTTP and TLS inspected HTTPS cases, the firewall ignores the packet's destination IP address and uses the DNS resolved IP address from the Host header. The firewall expects to get port number in the Host header, otherwise it assumes the standard port 80. If there's a port mismatch between the actual TCP port and the port in the host header, the traffic is dropped. DNS resolution is done by Azure DNS or by a custom DNS if configured on the firewall.

**Note**

Both HTTP and HTTPS protocols (with TLS inspection) are always filled by Azure Firewall with XFF (X-Forwarded-For) header equal to the original source IP address.

When an application rule contains TLS inspection, the firewall rules engine process SNI, Host Header, and also the URL to match the rule.

If still no match is found within application rules, then the packet is evaluated against the infrastructure rule collection. If there's still no match, then the packet is denied by default.

**Note**

Network rules can be configured for TCP, UDP, ICMP, or Any IP protocol. Any IP protocol includes all the IP protocols as defined in the Internet Assigned Numbers Authority (IANA) Protocol Numbers document. If a destination port is explicitly configured, then the rule is translated to a TCP+UDP rule. Before November 9, 2020, Any meant TCP, or UDP, or ICMP. So, you might have configured a rule before that date with **Protocol = Any**, and **destination ports = '\*'**. If you don't intend to allow any IP protocol as currently defined, then modify the rule to explicitly configure the protocol(s) you want (TCP, UDP, or ICMP).

## Inbound connectivity

### DNAT rules and Network rules

Inbound Internet connectivity can be enabled by configuring Destination Network Address Translation (DNAT) as described in [Filter inbound traffic with Azure Firewall DNAT using the Azure portal](https://learn.microsoft.com/en-us/azure/firewall/tutorial-firewall-dnat). NAT rules are applied in priority before network rules. If a match is found, the traffic is translated according to the DNAT rule and allowed by the firewall. So the traffic isn't subject to any further processing by other network rules. For security reasons, the recommended approach is to add a specific Internet source to allow DNAT access to the network and avoid using wildcards.

Application rules aren't applied for inbound connections. So, if you want to filter inbound HTTP/S traffic, you should use Web Application Firewall (WAF). For more information, see [What is Azure Web Application Firewall](https://learn.microsoft.com/en-us/azure/web-application-firewall/overview)?

## Examples

The following examples show the results of some of these rule combinations.

### Example 1

Connection to google.com is allowed because of a matching network rule.

**Network rule**

* Action: Allow

Expand table

| **name** | **Protocol** | **Source type** | **Source** | **Destination type** | **Destination address** | **Destination ports** |
| --- | --- | --- | --- | --- | --- | --- |
| Allow-web | TCP | IP address | \* | IP address | \* | 80,443 |

**Application rule**

* Action: Deny

Expand table

| **name** | **Source type** | **Source** | **Protocol:Port** | **Target FQDNs** |
| --- | --- | --- | --- | --- |
| Deny-google | IP address | \* | http:80,https:443 | google.com |

**Result**

The connection to google.com is allowed because the packet matches the Allow-web network rule. Rule processing stops at this point.

### Example 2

SSH traffic is denied because a higher priority Deny network rule collection blocks it.

**Network rule collection 1**

* Name: Allow-collection
* Priority: 200
* Action: Allow

Expand table

| **name** | **Protocol** | **Source type** | **Source** | **Destination type** | **Destination address** | **Destination ports** |
| --- | --- | --- | --- | --- | --- | --- |
| Allow-SSH | TCP | IP address | \* | IP address | \* | 22 |

**Network rule collection 2**

* Name: Deny-collection
* Priority: 100
* Action: Deny

Expand table

| **name** | **Protocol** | **Source type** | **Source** | **Destination type** | **Destination address** | **Destination ports** |
| --- | --- | --- | --- | --- | --- | --- |
| Deny-SSH | TCP | IP address | \* | IP address | \* | 22 |

**Result**

SSH connections are denied because a higher priority network rule collection blocks it. Rule processing stops at this point.

## Rule changes

If you change a rule to deny previously allowed traffic, any relevant existing sessions are dropped.

## Three-way handshake behavior

As a stateful service, Azure Firewall completes a TCP three-way handshake for allowed traffic, from a source to the destination. For example, VNet-A to VNet-B.

Creating an allow rule from VNet-A to VNet-B doesn't mean that new initiated connections from VNet-B to VNet-A are allowed.

As a result, there's no need to create an explicit deny rule from VNet-B to VNet-A. If you create this deny rule, you interrupt the three-way handshake from the initial allow rule from VNet-A to VNet-B.

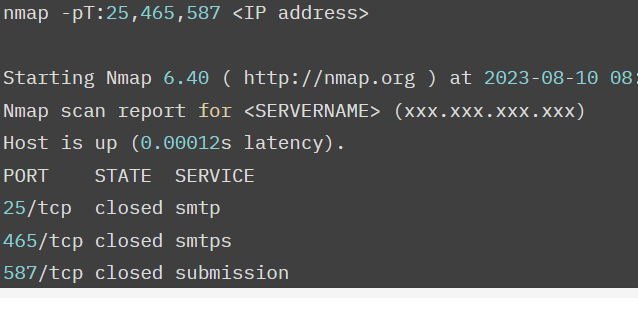
**STAGE-3**

**QRadar**

There is a way to test if a mail server is sending reports and offenses. This procedure allows you to log in to the mail server and run commands to help you determine whether the mail server is working properly.

**Procedure to test email services from QRadar:**

1. Connect to the QRadar Console by using SSH.
2. Scan your SMTP Server for open ports.

****

1. Connect to your email server from the QRadar Console that you are sending email from on the open port.

telnet IPADDRESS:PORT

Or

openssl s\_client -connect IP address:PORT -starttls smtp

**Note**: Change the IP address and PORT to the required server details.

1. At the email server's command prompt, type the EHLO command by using the name or IP address of the email server that is used by the QRadar appliance.

EHLO smtp.my\_mail\_server.com

1. Type the Mail from line:

MAIL FROM: administrator@qradar.com

1. Enter the email address that you want the email to be sent to:

RCPT TO: Account@email\_address.com

1. Enter the DATA command to begin entering the body of the email:

DATAEnter the body of the email. Type Enter and period**.** to end and send the email.**Hello,This is a test email  
.**Quit the session.quit

**CONCLUSION**

Cyber security is one of the most important aspects of the fast-paced growing digital world. The threats of it are hard to deny, so it is crucial to learn how to defend from them and teach others how to do it too. If you want to learn more about what is cyber security and how to deal with cyber criminals hop into our smart internz and become a hero in the digital platforms.

 cyber security awareness training for employees is not just a best practice; it is a necessity in today’s digital landscape. The benefits of such training are numerous, ranging from risk mitigation and data protection to compliance and cost savings. Businesses that prioritise cybersecurity education for their staff not only enhance their security posture but also contribute to a safer online environment for everyone. In the ever-evolving world of cybersecurity, knowledge is power, and investing in employee training is an investment in the future security and success of the organisation.

**Future scope**

With the domain of cyber security future scope expanding rapidly, you can expect numerous job opportunities in corporate companies. This also widens the future scope of cybersecurity for trained professionals. As a trained professional, you can pursue several career options.

If you have a certification from a reputable training institute, you can easily hope to join a corporate company, MNC, or an organization as:

**Security Analyst**

**Security Engineer**

**Security Architect**

**Security Administrator**

**Security Software Developer**

**Cryptographer**

**Chief Information Security Officer**

**Security Consultant or Security Specialist**

All these jobs are in high demand in India with high scope for cyber security and is at its all-time high now. The main reasons for high job demand stem from the fact that all corporate companies want to ensure that any intrusion is detected promptly, malware entry is forestalled, cloud security is maintained completely, all risks are analyzed and mitigated, and overall network security is maintained with total perfection.

### **REFERENCES**

[(PDF) Malware Detection & Classification using Machine Learning (researchgate.net)](https://www.researchgate.net/publication/349613618_Malware_Detection_Classification_using_Machine_Learning)

<https://learn.microsoft.com/en-us/azure/firewall/threat-intel>

**THANK YOU SMART INTERNZ**