Assessing the Health of a Network Under Attack

Pedro Marques
Applied Research
British Telecom
Ipswich, United Kingdom
pedro.marques@bt.com

Alfie Beard
Applied Research
British Telecom
Ipswich, United Kingdom
alfie.beard@bt.com

Jonathan Roscoe
Applied Research
British Telecom
Ipswich, United Kingdom
jonathan.roscoe@bt.com

***Abstract*—When faced with a malware outbreak, the health of a computer network is hard to quantify. Calculating the number of infected nodes is a straightforward approach, but it fails to capture intricacies of the devices that make up the network. The choice between which of two network states is preferable might not correlate directly with the number of infected nodes in each, as different nodes carry different importance to the overall function of the network. In this paper we propose a method of assessing the health of a network under attack from a malware outbreak. The proposed method allows for a quantitative measure of how well a network is handling a malware outbreak, as well as the comparison between different network states and the ranking of possible mitigating actions. The method proposed can be adapted to different networks, with its usefulness increasing with the amount of data available for a given network.**

Keywords—Network security, network modelling, security framework.

I. Introduction

Malware outbreaks continue to pose a significant challenge for security professionals tasked with protecting computer networks. The number of cyber incidents is continuously on the rise, with the frequency of enterprise ransomware incidents increasingly making headlines in news reports [2, 10]. When a malware outbreak is detected, the most straightforward approach to take is to simply shut down the network. Such a drastic action eliminates the threat entirely, but this is seldom an option for large enterprises, as these face a large number of security alerts, many of which false positives [7].

Instead during a cyber response, it may be preferable to "tolerate" a possible malware outbreak, through the enacting of different mitigating actions that affect how a malicious agent can spread across the network, and how much damage they are able to cause. These actions can include removing or isolating specific devices on the network, disallowing specific ports, services, applications or users from operating throughout the network, or deploying additional firewall protections, among others. These actions will often have an impact on the performance and quality of a network, for example, by preventing users from carrying out their normal business operations. Thus, a careful balance needs to be struck between defending against a network intrusion and the extent to which normal network functionality is compromised.

Security professionals must take decisions that balance confidentiality, integrity and availability [5]. These decisions however, are often hard to justify, due to a lack of meaningful and direct metrics to assess the state of a network. The goal is to keep the network as "healthy" as possible, preserving as much of the original network functionality, whilst minimising the impact of a malware intrusion. From a high-level point of view, a network with a large number of infected devices is an unhealthy network. Likewise, a network operating under a large amount of restrictions from enacted defensive actions is also an unhealthy network.

In this paper, we propose an approach for calculating a "health score" for a computer network during the course of a security event. This health score comes in the form of a percentage metric, allowing for an easy assessment of a network's condition at any one point in time, as well as the comparison of different network states relative to each other. The ability to compare different network states' health is especially important, as it allows for the comparison of different mitigating actions. We show how coupled with a malware simulation engine, such an ability allows for a more accurate forecasting of the potential damage and reach of a malware, and consequently lead to a more robust defence of a network by allowing for the selection of the best possible mitigating action.

This paper is organised as follows: in section II we give a brief overview of related work, and the current implementations of the most similar techniques. Section III gives an overview of the framework we propose for calculating the health score of a computer network. We exemplify the use of the framework within our threat response tool, in section IV, and further discuss the nuances and requirements for extending the framework to accommodate different scenarios in section V. Finally, section VI concludes with a discussion of the limitations of our proposal, as well as calls for additional works on the subject.

II. Related work

Attack-graphs are often employed for the defence of networks. Such graphs represent the possible paths an attacker can take to reach a certain target or goal within a given network, constructed from the vulnerability information present about the devices and their connections. There is a large body of research on the use of attack-graphs [1, 4, 6] for both detection and defence, and these provide security professionals a way of identifying critically vulnerable paths an attacker might use to traverse a network. Attack-graphs can be used to extrapolate an overall security score to a device, set of devices or the overall network. The authors of [9] propose a method for defining a risk score based on the Common Vulnerability Scoring System (CVSS) of vulnerabilities found on devices and the connections between them. Whilst not explicit, such a method could be extended to provide an overall network risk score. However, this risk score would be a constant value of a network's risk, dependent on the attributes present at a fixed point in time and would not reflect any current state of a network during a malware outbreak (i.e., it would not take into account devices already compromised by an attacker).

Explicit work on classifying the health of a network during an outbreak is not prominent in the literature. Most notably, Gujral et. al detail a network health monitoring system

979-8-3503-3036-6/24/$31.00 ©2024 IEEE

achieved through the monitoring of end-user inbound and outbound network traffic, anomalous fingerprints and known vulnerabilities [3]. Such a system provides real-time analysis of the health of the network, but requires explicit network traffic data and vulnerability knowledge, which can limit its applicability. Similar works have been patented before [8, 11] however, these approaches rely on the monitoring of network activity and well-known security events, which again limit their applicability towards generic network environments.

III. Framework

The health of a network is directly dependent on the state of the devices that it incorporates, and whether or not each device is operating as intended across the multiple network layers. Different devices contribute different amounts to the health of a network. For example, when considering an industrial SCADA network, it would be reasonable to assume that devices which directly control large-scale machinery, such as turbines, are of extreme importance and that, from the perspective of a network administrator, are critical points to defend in the network. In contrast, printer devices are less important in the overall network, meaning that temporary restrictions on these are generally more acceptable.

The importance of a device is a double-edged sword, as their ability to be impactful for the healthy operations of the network is also equally impactful for the destructive purposes of a malware. A basic example of this is a central router with connections to multiple outer devices. This central router is of great importance to the overall communications within the network. However, if it becomes compromised, then an attacker gains access to a large number of new victims, as well as potentially sensitive information being transmitted over the network.

The impact of a device on the overall network can also change over time, depending on a multitude of reasons. Continuing from the previous example, a central router's impact to the correct operation of the network is in part related to the number of connections it has with other devices. Restricting the number of connections a device has access to lowers its overall impact on the network's health. This is the case whether the device is currently infected or not. In other words, non-infected devices are less important to the normal operation of the network if they have fewer links to other devices, and an infected device's negative contribution to the health score is similarly diminished due to the same circumstance - i.e., it has less linked devices that it can try to exploit or gain information from. Additionally certain nodes' importance's are time-sensitive - for example, databases might be accessed more often at specific times of day.

The main challenge of our framework is assigning a contribution value to each device on a network. We devise this process as a modular approach, capable of being extended in tandem with the amount of data available for a particular network and the devices within it.

Our framework uses an abstract representation of a computer network, that can have varying levels of complexity depending on the outcomes and intelligence available. At the most basic level, the topology of a network - even if incomplete - is something that is almost always available to a security professional. With this being the case, it is common to find computer networks modelled as structured graphs. In such representations, the various devices that make up a network (e.g., servers, host machines, routers, switches, sensors, etc) are represented as nodes in the graph. Any links between the nodes represent possible communication channels between devices[1]. Figure 1 shows how a typical computer network might be modelled in this manner.

The first approach for calculating the impact of each device on a network is to calculate the centrality[2] of each node. In the absence of additional data, nodes with higher centrality values - i.e., nodes with a higher number of connections - are more important to the operation of a network. All else being equal, it is straightforward to assume that a node with a single connection is less impactful and easier to replace than a node with multiple connections.

Consider the examples shown in figures 2 and 3, where we present two possible states of the same network. In figure 2 the node in the centre of the network is infected, while in figure 3 two of the outer nodes are infected. With a more traditional security view we might consider that a higher number of infected nodes is a worse state for our network. However, this might not necessarily be the case if the nodes in question are of little importance to the overall network's function. Lacking any details of the devices past where they lay on the network, it's not unreasonable to assume that the central node being infected is sub-optimal compared to the alternative[3]. Thus, we assign for each node a contribution value equal to its centrality.

After determining the contribution value of a node, the next step is to identify whether the node is infected or not. As we described previously, each infected node still contributes the same absolute amount to the overall network health, but in being in the control of a malicious actor, it can use this leverage to attack the network. In a sense, we simply negate the value of contribution for each node that is infected.

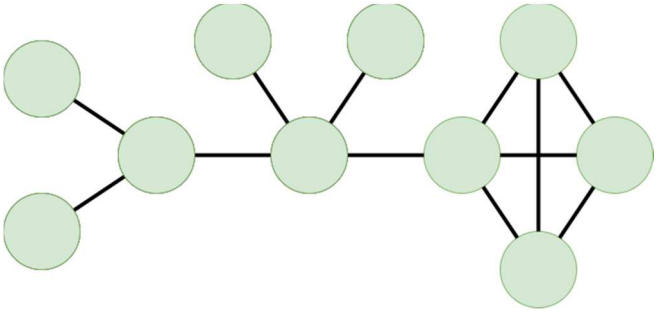

Figure 1: Example graph for a computer network

[1] Note that links can denote 'direct' paths between devices, but sequences of links can also denote communication channels that involve one or more 'steps', such as traffic going through a firewall when travelling between two end-user devices.

[2] There are different ways of calculating the centrality of nodes, for the purposes of this paper we will be using degree centrality.

[3] The central node has direct access to four other nodes which can be potential victims, whilst the two outer nodes only have access to a single potential victim.

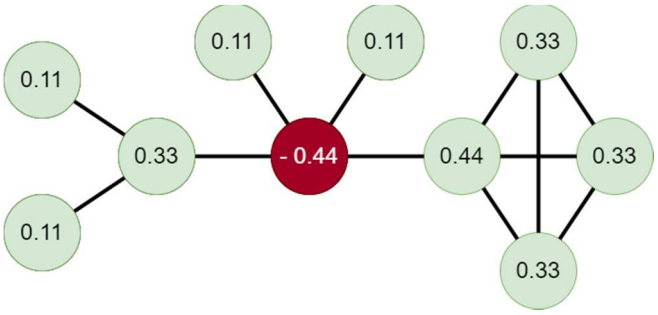

Figure 2: Network with center node infected

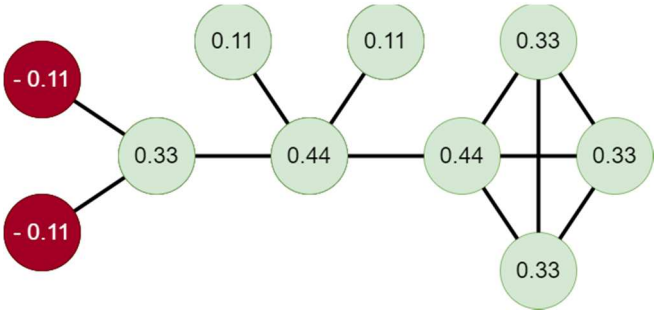

Figure 3: Network with two outer nodes infected

Following the calculation of each node's contribution value, the health score for the overall network is simply the sum of all of its nodes' contributions. For the examples in figures 2 and 3, the contribution values are shown as the labels on each node, and the overall network health score is 1.76 and 2.2, respectively.

As an absolute value, the health score metric calculated above is enough to compare different states of the same network, where greater values indicate healthier network states. However, on its own the score is unintuitive, as it can vary wildly between different networks and states of the same network. Knowing that the network state presented in figure 2 has a health score of 1.76 tells an analyst nothing without the context of knowing the limits of this score. In order to increase the metric's explainability, we must first determine what the best and worst possible states for the network are. These limit states are straightforward to calculate if we have no additional data on the network. The best possible state is one where all of the nodes are not infected, while the worst possible state is one where all the nodes are infected. For the network states shown in figures 2 and 3, the best possible health score would be 2.64 and the worst possible health score would be -2.64. Therefore, the network states have a normalized health score of 83.3\% and 91.6\%, respectively.

So far, we have calculated a contribution value for each node by using only its centrality. While this is a decent start if we have no additional data about the network, the centrality of a node is far from a realistic representation of how important nodes in a network truly are. In a perfect world, a network administrator would be able to assign a final contribution value to each device a priori, perfectly encapsulating a node's true importance to the network. However, besides this being a far-fetched ideal, it also suffers from the problem that it does not update during the course of a security incident. As different events occur on the network, either from actions taken by a security professional, or based on actions from the malware itself, the contribution of each node changes.

For example, additional firewall rules might be put in place on a load-balancing server, which impacts its performance and thus degrades the quality of the service to end-users. This change in performance ought to be captured in the calculation of contribution, with devices subjected to lower performance contributing less to the overall health of the network. Another example might be in disabling specific services from being run on devices throughout the network, which might make it difficult for malware to propagate, but at the same time reduce the functionality of the devices, and thus lower the overall health of the network.

From the point of view of the malware, the actions it takes can also have an impact on the contribution value of each node. For example, by opening new ports on infected devices, it can increase its ability to spread to new victims. In this sense, the contribution of the infected node increases, but this increase works in favour of the attacker, as it results in an overall greater negative impact for the network health.

In order to capture these aspects, we need to add additional steps to our calculation of contribution scores, past our initial computation of centrality. What these steps look like will depend on the nature of the network, the amount of data available, and which data is important to the security requirements of the network. We should also take into account what actions a security professional or the malware itself might enact, and make sure that the effects of these can be captured in some way in one or more of these steps.

For example, the number of services a device is running might be of importance to the contribution of a device, and these services might be started or stopped during the course of an outbreak. As such, one possible step would be to simply take the number of services running on a device and add that value to the contribution score. A more sophisticated approach could be, for example, to give higher value to some services compared to others. In this sense, the value of services like SSH or FTP being present on a device might be doubled compared to the presence of other, less important services. The granularity of each of these steps is solely dependent on the amount of data available for the network and the requirements of the security administrator. For example, different versions of SSH could be given different values, or take into account whether SSH is run using a password or a certificate.

The addition of more steps to the calculation of the contribution score of each device leads to an issue of mismatched units between the various components. For instance, while in our previous example in figures 2 and 3, the value of centrality was always lower than 1.0, the number of services running would be substantially higher. In order to account for multiple distinct calculation steps such as these, the resulting value of each individual calculation step should be multiplied by a weight variable, with the intent of creating parity between the various results. While in this paper we will present a variety of different example steps that can be taken based on the data available for a given network, the weights that should be given to each of these steps is left as an exercise to individual network administrators to tailor to their specific environment. This is because accurately capturing the elements that are impactful to devices' contribution values, is one that changes not only based on the network in question, but also the priorities of the security team and the nature of the malware being defended against. While it may be possible to arrive at a general consensus in regard to the use of common

properties, in this paper we make no claims on the subject and leave this as an open question for future work.

IV. Framework Implementation

Inflame is a network modelling and malware simulation tool currently in development at BT. Inflame models computer networks in much the same manner as shown in figure 1. On top of this model, it simulates the outbreak of a particular malware on the network, for example, a ransomware attack, and simulates how the malware might spread from device to device across the network. This malware propagation is done in an epidemiologically based approach, with infected devices being able to infect their direct neighbours according to a set of rules that dictate the necessary attributes each device must have to be susceptible, as well as an overall infection rate set for the simulation.

The simulation is based on the idea of steps, where each step represents an arbitrary unit of time (e.g., 5 minutes, an hour, a day, etc). At each new step the simulation calculates the probability that any compromised node infects one or more of its neighbours. We calculate the network health score at each step in the simulation, allowing for an analyst to understand how an outbreak will evolve over time, and which steps are more damaging to the health of the network. The network model in our tool is a generic one, where devices can have any number of data points, such as open ports, services running, user accounts, known vulnerabilities, etc. Using these additional data points, we extend the framework for the calculation of each node's "contribution value", to better capture the importance of each node. This extended framework is exemplified in figure 4.

When calculating the contribution value of a given node, we first initialise two variables - the contribution value itself, initially set to 0, and an "importance factor". The importance factor is a value assigned to each node on the network by an administrator, as an attempt to capture an expert's opinion on how impactful any given node is to the network as a whole. If not given, this importance factor is initialised to 1.

Following this, a series of steps modifies the contribution value:

1. The centrality of the node is multiplied by a weight variable and added to the contribution value.
2. The sum of the node's neighbour's importance factors is multiplied by a weight variable and added to the contribution value.
3. If the node can become infected - i.e., it has not been patched or otherwise made immune to the malware - the value of a weight variable is added to the contribution value (This step is a binary one - i.e. the node is either vulnerable or not, and as such the increase in contribution is a statically assigned value. This is shown in figure 4 as "1 * weight").
4. The contribution value is then set to the product of itself by the node's importance factor.
5. Finally, if the node is currently infected, the contribution value is negated.

The contribution value of each node, and consequently the health score of the network is calculated at each new step in the simulation. This has two direct benefits. Firstly, the progression of a network's health score can be analysed as the malware outbreak continues, allowing an administrator to understand how the security of the network evolves over time, and which areas become compromised when. Secondly, because the health score of a network is calculated based on the contribution value of each node within it, it becomes possible to identify regions of the network of particular importance. For example, regions most at risk of becoming targets of infection based on their contributions to the overall network, or regions which have already been affected the most by enacted security actions.

In figure 5 we show an example of network regions visualised within Inflame. The colours of the various nodes represent their total contribution value, with blue nodes contributing the most positively, followed by greens, oranges and finally reds, which contribute the most negatively. In this example, we can easily identify the nodes in the largest subnet at the top of the figure as contributing the most towards the health of the network, with 4 instances of infected nodes within said cluster. The region with the most infected nodes is found in the subnet left of the central router, with 6 total infected nodes (in orange). Individual network regions, such as specific subnets, can be treated as networks in and of themselves, and so health scores for individual regions can also be calculated.

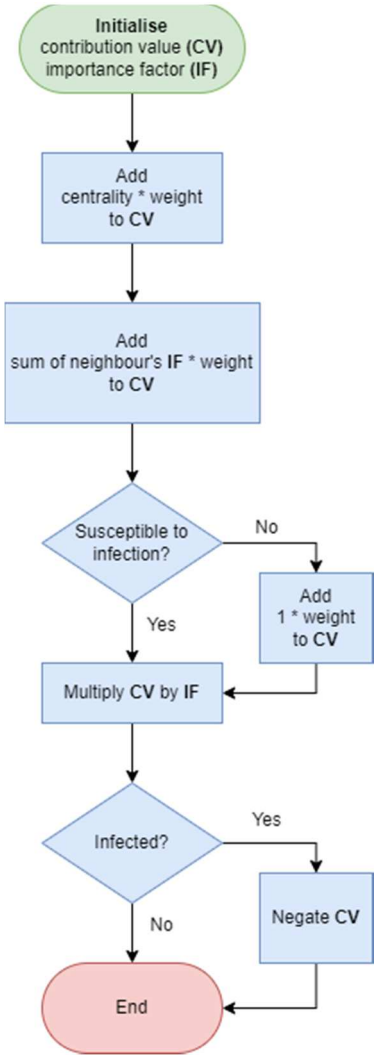

Figure 4: Inflame contribution value diagram flow

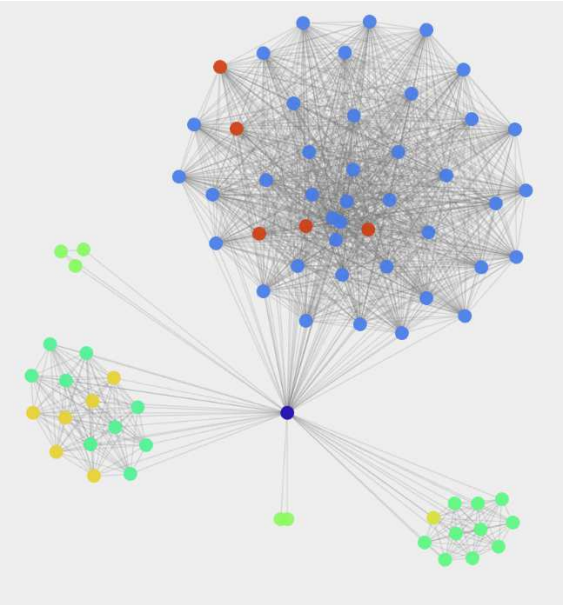

Figure 5: Example of identifying regions (health score = 85.42%)

One of the core functionalities of Inflame, is as an automated threat response tool, capable of determining the most appropriate actions in responding to malware outbreaks. We leverage the simulation and network health assessment to trial a large set of different actions and determine which of these result in the best possible end state for a network. The possible actions trialled mirror those that security professionals have access to, such as patching vulnerable nodes, isolating nodes from the network, disabling ports and applications, etc.

To illustrate this threat response process, consider the series of events depicted in figures 6 through 9, where we show an initial state of a network under attack (figure 6). This initial state depicts a network with a single infected node, itself connected to two other nodes, A and B. For the sake of simplicity in this example, we are considering the contribution value of each node to be equal to their centrality value. At this initial state, the network health score is 2.43, which after calculating the best and worst scores, can be converted to 92.33\%. We can simulate the next step and understand how the malware might spread, and in this example we see that both node A and node B become infected (figure 7), leading to a new network health score of 0.87 (65.16\%).

Knowing this, we can analyse two possible basic mitigating actions: (i) isolating node A from the network (figure 8) or (ii) isolating node B from the network (figure 9). Each of these actions would mean that the links involving each of these nodes would be removed[4]. We can simulate these actions and calculate the health score for the network in the following simulation step(s). In our example, removing node A, prevents it from being infected, resulting in a network health score of 1.27 (72.13\%), while removing node B results in a network health score of 1.26 (71.95\%). While subtle, this allows us to identify that either of our actions results in a better network state compared to doing nothing, and that removing node A is a slightly better action than removing node B.

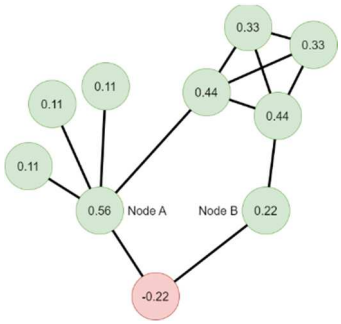

Figure 6: Action example, initial state (92.33%)

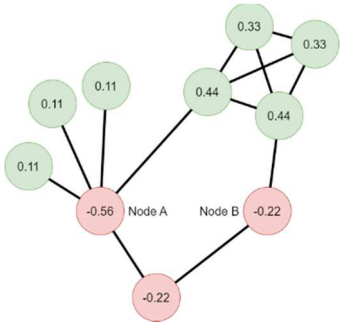

Figure 7: Action example, no action (65.16%)

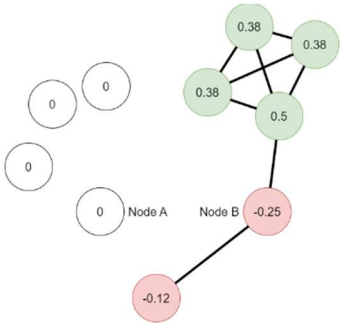

Figure 8: Action example, remove A (72.13%)

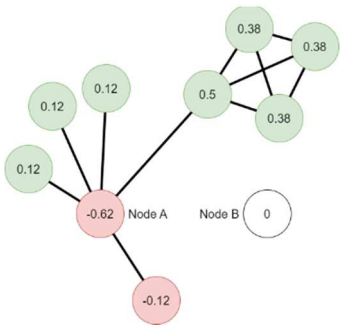

Figure 9: Action example, remove B (71.95%)

It is important to emphasise that the difference between the two actions is subtle primarily because we're only considering the centrality of nodes. In a more realistic scenario, the decision between removing either node will be dependent on a number of additional factors, such as their purpose within the network, the cost of removal/reintroduction, etc. - all aspects which should be accounted for in the calculation of contribution values for each node in the network.

[4] Note that by removing links from the graph, the centrality value of nodes changes.

Relying on the results of simulations may not be enough, and it is important that the effects of all possible actions are captured in the calculations of contribution values of nodes. Consider the case of an infected node “K”, connected to a susceptible node “V”. Because our infection model is a probabilistic one, it's not necessarily the case that node V will become infected in the very next step of our simulation. Intuitively however, the choice of patching node V over doing nothing is clearly the better option, as it prevents the node from being infected in the case that we “lose the dice roll”. Our network health calculations capture this benefit of patching a node in step 3, by increasing a node's contribution value if it's not susceptible to being infected, which will consequently improve the overall network's health score, and increase the perceived value of taking that action.

V. Extending the framework

As discussed previously, the method we are proposing allows for the addition of extra steps in the calculation of each node's contribution value, based on the amount of data available for the network in question. The attentive reader might at this point realise the fact that the definition of the “best” and “worst” network states is dependent on the steps we choose for determining the contribution value of each node in the network. In fact, the implementation described in our tool has one such caveat. When accounting for whether a node is susceptible to being infected or not, we have imposed an additional requirement on the definition of the best possible state. In allowing for this action to be captured in the contribution value calculation, the best possible state for the network is not only one where all of the nodes in the network are free from infection, but also where all of the nodes are immune to the malware in the first place.

The determination of the best and worst possible states for any network is thus a subjective one, dependent on the data available for the network as well as the use cases we are interested in. However, one caveat seems almost universal to any network, which is related to the network's original topology and additions to it. Take for example, the ability to deploy additional firewall devices, with the intent of segregating the various nodes in a network. If these additional firewalls are not accounted for in the original calculations of best and worst network health scores, this leads to the unintended consequence of recursively improving the network's health by continuously adding new firewalls to the network. Similarly, the health score could continuously decrease as these new devices become infected, leading to a network state with more infected devices than total devices present in the original network topology.

This idea is not only true for the network topology, but also attributes of the devices themselves. We run into a similar problem if, for example, the number of services running on a device increases their contribution value. There is a virtually infinite number of services that can be deployed on a device, and thus this would again lead to recursively increasing the health score of a network by continuously starting new services. In this case, a maximum limit to such a calculation step, or a fixed definition of which services count for the contribution value of a device becomes necessary.

One aspect we have not demonstrated is the definition of “negative” steps in the calculation of contribution values. Say that we want network devices to have a maximum of three connections, as any more connections might be considered a significant security risk. We can achieve this by adding a new contribution value calculation step that reduces the contribution value of a given node if it has more than three connections. Figure 10 is an example of such a network, where we have defined that the contribution value of a node is reduced by half, if it has more than 4 connections. In the case that the affected nodes are not compromised, this works correctly, as we are essentially penalising excessive connections. However, if this node were to be infected, this step no longer makes sense, as an infected node with additional connections would pose a larger risk to the network, and so its contribution value should not be penalised in the same manner. In such cases, the current state of the node (i.e., whether it is infected or not) should be considered to decide the contribution value of a node.

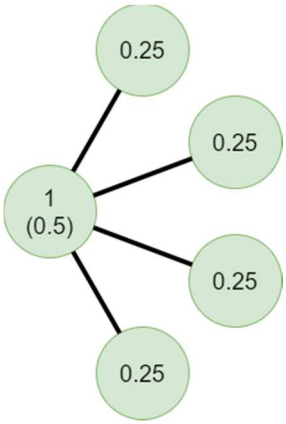

Figure 10: Example of negative calculation steps

VI. Discussion

In this paper we have proposed a method of assessing the health of a computer network during a security incident. The method works by calculating a contribution value for each device in the network, meant to encapsulate how important that device is for the overall network's function. This contribution value can be positive (if the device is benign) or negative (if the device is malicious), and the calculation is based on the data available for the network, as well as the intended use cases and possible action space for a defending agent. Elements such as the device's centrality in the network, or the services it runs affect the contribution value of the device, and we exemplify the use of our method with our internal malware modelling and threat response tool. The use of a network health score allows for a security professional to understand how a network's state evolved over the course of a malware outbreak and enables the selection of mitigating actions based on their effects on the network's health.

It is important to emphasise that the framework we have presented in this paper \textbf{must} be adapted to the individual needs of a network. The question behind what makes a device "important" to the network is dependent on the data available for said device, and will differ drastically between different network scenarios.

Throughout the paper, we have repeatedly used devices's centrality as part of the calculation of their contribution values. However, we must consider that a for 'raw' physical topology, like the ones we have presented in this paper this may not be the most suitable. For example, while a node may be 'physically' connected to a switch, which in turn is 'physically' connected to another device, both end devices could have a

'virtual' connection between them. In this sense, a malware looking to spread from device A to B might not even be aware of the switch in the middle, yet this does necessarily prevent it from spreading from one device to another. As such, the topology used for the calculation of health score should take this knowledge into account, again, if such data is available.

Currently we leave the determination of what makes devices 'important' to the network as a question for each network operator to decide, however future work is warranted for determining a general consensus of what elements are more often critical in assessing a healthy network state.

REFERENCES

[1] Paul Ammann, Duminda Wijesekera, and Saket Kaushik. 2002. Scalable, graph-based network vulnerability analysis. Proceedings of the ACM Conference on Computer and Communications Security, 217–224. https://doi.org/10.1145/586110.586140

[2] CrowdStrike. 2023. CrowdStrike 2023 Global Threat Report. (2023).

[3] Harshit Gujral, Abhinav Sharma, Pulkit Jain, Shriya Juneja, and Sangeeta Mittal. 2022. Design and Implementation of a Quantitative Network Health Monitoring and Recovery System. Wirel. Pers. Commun. 125, 1 (jul 2022), 367–397. https://doi.org/10.1007/s11277-022-09554-9

[4] Kyle Ingols, Richard Lippmann, and Keith Piwowarski. 2006. Practical Attack Graph Generation for Network Defense. In 2006 22nd Annual Computer Security Applications Conference (ACSAC'06). 121–130. https://doi.org/10.1109/ACSAC.2006.39

[5] Cawthra J., Ekstrom M., Lusty L., Sexton J., and Sweetnam J. 2020. Data Integrity: Identifying and Protecting Assets Against Ransomware and Other Destructive Events. (2020).

[6] S. Jha, O. Sheyner, and J. Wing. 2002. Two formal analyses of attack graphs. In Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15. 49–63. https://doi.org/10.1109/CSFW.2002.1021806

[7] Palo Alto Networks. 2020. The 2020 State of Security Operations. (2020).

[8] Walker R., Amrutur B., Mottishaw P., Joiner S., Chesler L., and Hardcastle I. 2001. Network monitoring system with built-in monitoring data gathering. Patent No. 6975617, Filed Mar. 23rd., 2001, Granted Dec. 13th., 2005.

[9] Gun-Yoon Shin, Sung-Sam Hong, Jung-Sik Lee, In-Sung Han, HwaKyung Kim, and Haeng-Rok Oh. 2022. Network Security NodeEdge Scoring System Using Attack Graph Based on Vulnerability Correlation. Applied Sciences 12, 14 (2022). https://doi.org/10.3390/app12146852

[10] Symantec. 2021. Ransomware Threat Landscape: What to Expect in 2022. (2021).

[11] Aki Y. and Saito H. 2001. Network monitoring system. Patent No. 7353269, Filed May 23rd., 2001, Granted Apr. 1st., 2008