Network Forensics: Analysis of Bank of Money's Regional Office Operations

Giorgio Daneri[1], Nicola De March[2], Bice Marzagora[3]

University of Luxembourg.

Contributing authors: giorgio.daneri.001@student.uni.lu; nicola.demarch.001@student.uni.lu; bice.marzagora.001@student.uni.lu;

1 Introduction

This project focuses on analyzing firewall and IDS logs as part of Mini-Challenge 2 of the VAST 2012 Challenge. We focused on analyzing network activities captured in firewall and IDS logs to uncover patterns indicative of malicious activities, ranging from unauthorized connections to data exfiltration attempts. To achieve this, we utilized Python for data pre-processing and D3.js for data visualization. Following an initial phase of data filtering and analysis, we developed an interactive visualization that provides clear insights into activity within the network. The structure of the report is as follows: the first section describes how we manipulated the datasets prior to the visualization, in order to make them more manageable on a local machine. The second addresses the questions posed by the VAST Challenge, especially the first and second one. It analyses all the major events, the security flaws and the suspect activities within the network. Then we proceed in briefly describing the visualization tools we used to convey important information and how the user can interact with them. The last section addresses the third question of the challenge in more depth, regarding the mitigation of security risks and the bad practices that were used by the network administrators.

2 Data Filtering and Sampling

The first key step was to pre-process the data in order to remove irrelevant information and map to numerical values the attributes that have a limited number of unique values in the form of strings. The use of a mapping from a set of strings to a set of numerical values is written to a file so that it can be used later on to reinstate the semantics of what would otherwise be meaningless numerical values. These are used during the processing steps to increase efficiency, but are reconstructed right before rendering the charts, so that the process is hidden from the end user of the visualization. This is useful to reduce the size of the datasets, especially those related to the firewall logs, which is quite cumbersome. All the entries corresponding to the `Source Hostname` and `Destination hostname` are emtpy, thus we dropped these attributes. We added a `key` attribute as a univocal identifier for each entry of the firewall dataset, since this is needed when the server applied pagination to the data prior to sending it to the React component. After transforming the data, we also applied sampling to the entries where the destination service is one of the two most common ones, namely `http` and `6667_tcp`. These logs amount to more than 95% of the total amount. We sampled 1 every 20 values for the former and 1 every 10 for the latter. We made sure that the distribution of the data was not altered significantly during the process, so that the visualization conveys the information correctly. All of these operations are applied before the server is launched, making sure they are executed only once, not every time the server is started. This choice was made to ensure that the computational load of the server is limited to loading the sampled datasets, eventually applying pagination and transmitting them to the javascript application. Also, the server uses caching to improve the overall efficiency, so that it reads the datasets from the memory only once.

3 Data Analysis and Findings

We performed several investigations of the datasets prior to conceiving and building the visualization. This was an important step to assess the information to be conveyed, as well as the best visualization tools to display it. The visualization must be driven by the findings, not the other way around. Let us pinpoint and describe the five main events happened in the time frame covered by the logs, according to our analysis. All these are included in the `analysis_results` notebook, which also offers a some raw visualizations of the information of our interest.

- **Unauthorized activity and potential security risks on ports 6667 and 22**:
  There is an increasingly large presence of network connections that use 6667 as destination port, commonly used for IRC (Internet Relay Chat), which is an activity that violates the Bank of Money policy. The first occurrence of such event is on April 5th at 20:25:35 by source IP 172.23.234.254, which corresponds to an individual workstation inside of the network. Actually, the connections using port 6667_tcp are the second most common ones, preceded only by the port 80 accessed with http protocol. The large use of the former is a fact to be investigated and is a symptom of the upcoming events, while the latter is not problematic per se. This service also supports file transfer capabilities, which could pose a security risk for sensitive data.

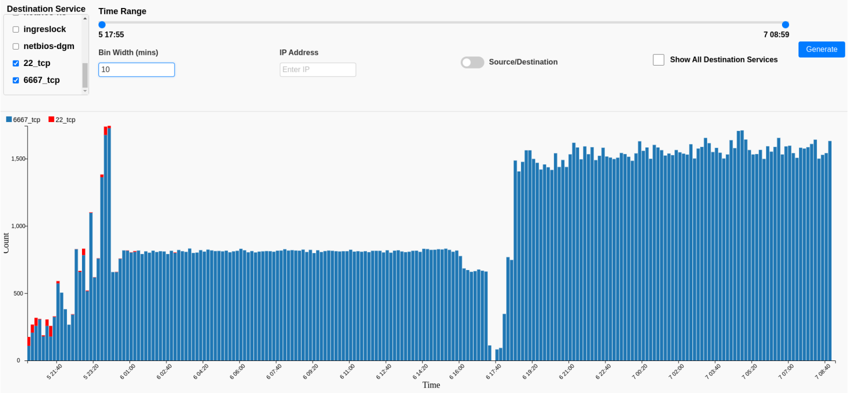
Table 1 Most Common Services

Destination Service	Count
http	21,331,506
6667_tcp	2,329,914
ftp	1,825
domain	752
22_tcp	538
1026_udp	194
netbios-ns	130
auth	62
1025_udp	50
https	42

Table 2 Least Common Services

Destination Service	Count
syslog	8
knetd	6
ms-sql-s	4
wins	3
pptp	3
netbios-dgm	2
kpop	2
ms-sql-m	1
ingreslock	1

  Note that in the second table, we excluded all the services that are classified based on the port, since they were not of our interest. Moreover, there is a consistent number of connections over port 22 TCP, which is commonly used for `SSH` and `SCP` protocols. Hackers often target port 22 because they could potentially do unlimited damage if they can log onto a remote device. Because of the risk, many organizations will turn off port 22, while the bank left it open. This could be a sign of external actors transferring data through the `SCP protocol` from compromised workstations to external servers.



Destination Service
☐ ingreslock
☐ netbios-dgm
☑ 22_tcp
☑ 6667_tcp

Time Range: 5 17:55 – 7 08:59

Bin Width (mins): 10

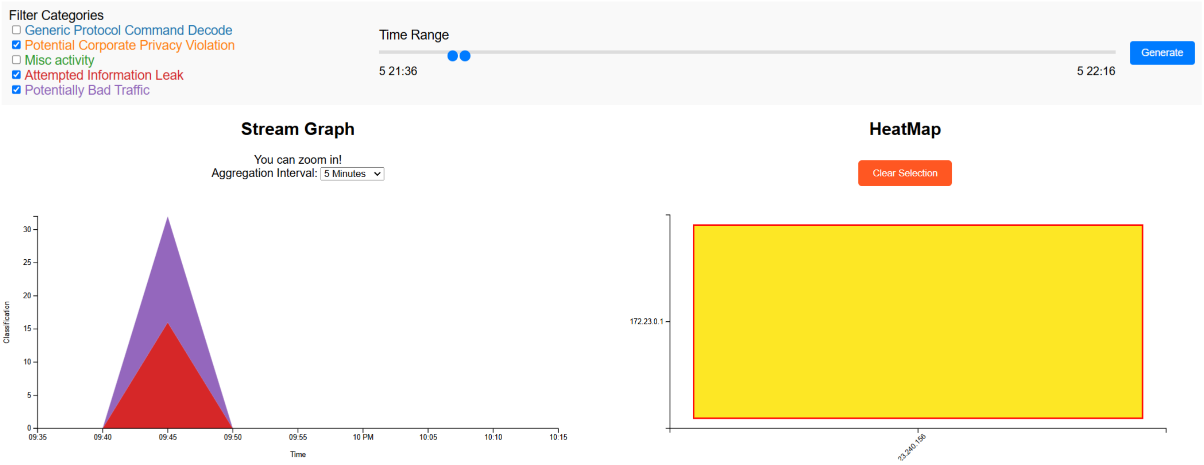
IP Address: Enter IP

Source/Destination

☐ Show All Destination Services

Fig. 1 Connections over port 22

- **Initial cyberattack attempts blocked by firewall**:
  We observe the first attempts to hack into the bank servers between 21:47 and 21:48 on April 5th. There are several logs labeled as suspicious inbound connections to the MSSQL, Oracle SQL, mySQL, and PostgreSQL servers. The source IP is 172.23.240.156, likely an infected workstation that is also part of a botnet set up by the attackers. The destination is 172.23.0.1, the firewall internal to the bank network, meaning that it managed to intercept the attack and stop it.


Filter Categories
☐ Generic Protocol Command Decode
☑ Potential Corporate Privacy Violation
☐ Misc activity
☑ Attempted Information Leak
☑ Potentially Bad Traffic

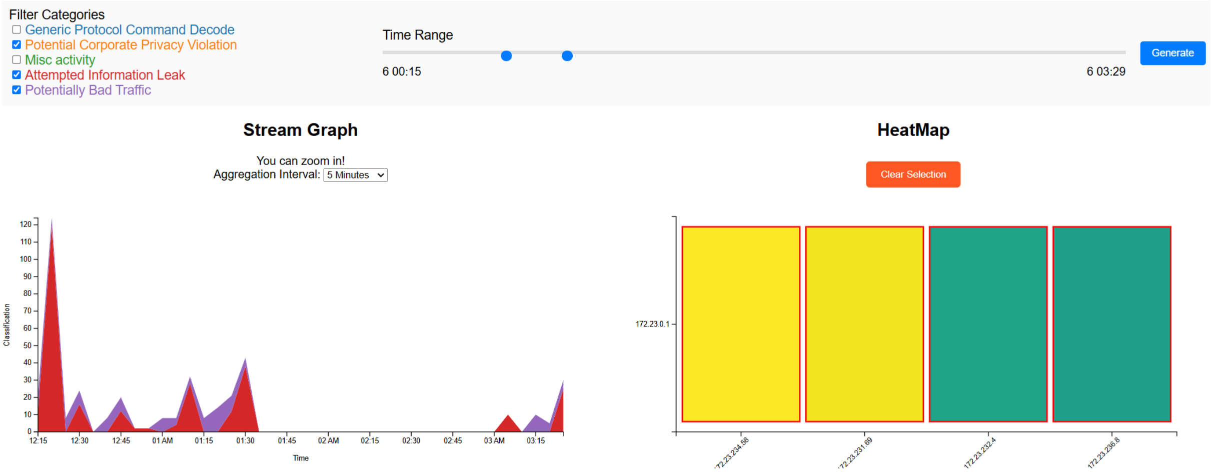
Time Range
5 21:36 5 22:16

Stream Graph
You can zoom in!
Aggregation Interval: 5 Minutes

HeatMap

Fig. 2 First attempts to hack into the bank from IP 172.23.240.156

Later on, other IPs such as 172.23.236.8, 172.23.231.69, 172.23.234.58 and 172.23.232.4 also target the bank. The attacks occur between 23:30 of the 5th and 03:20 of the 6th, all during the closing hours of the bank, leaving it less capable of reacting to the danger. So far, we have identified five compromised workstations. While several future connections will still be classified as suspect and dangerous, the above ones are actually the last ones labeled as Potential Corporate Privacy Violation, Attempted Information Leak, or Potentially Bad Traffic.


Filter Categories
☐ Generic Protocol Command Decode
☑ Potential Corporate Privacy Violation
☐ Misc activity
☑ Attempted Information Leak
☑ Potentially Bad Traffic

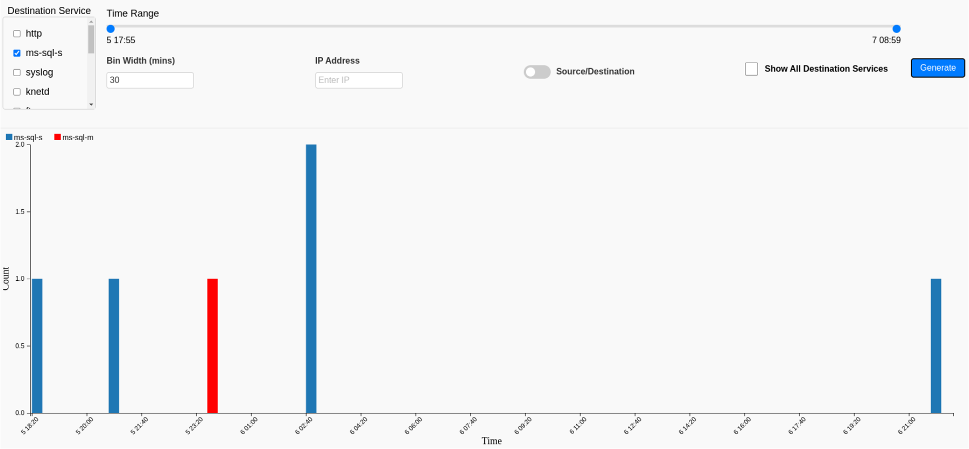
Time Range
6 00:15 6 03:29

Stream Graph
You can zoom in!
Aggregation Interval: 5 Minutes

HeatMap

Fig. 3 Attack fron 00:30 to 03:20 of 6th April

Moreover, at 23:45:29 of the 5th, the firewall logs show that an hack attempt to the MSSQL server by the means of the `ms-sql-m` exploit that targets Microsoft SQL database servers. This is was supposed to be the beginning of a privilege escalation attack, but the firewall manages to deny the connection and block the attack. The attackers would have been able to access the database and exfiltrate data at their will. Clearly, some external agents are targeting the Bank of Money with a series of cyberattacks, of which this is just the start. In facts, a similar exploit called `ms-sql-s` is used four times by external servers to attack the company, between 18:31 of the 5th and 21:51 of the 6th. Both vulnerabilities could be exploited due to the fact that the network administrators left port 1443 open. The most notable one happens at 2:48 of the 6th over port 6667 and is intercepted by the firewall interface to the internet. None of these attempts are successful, as they are all denied by the network defenses.



Destination Service
□ http
☑ ms-sql-s
□ syslog
□ knetd

Time Range
5 17:55
7 08:59

Bin Width (mins)
30

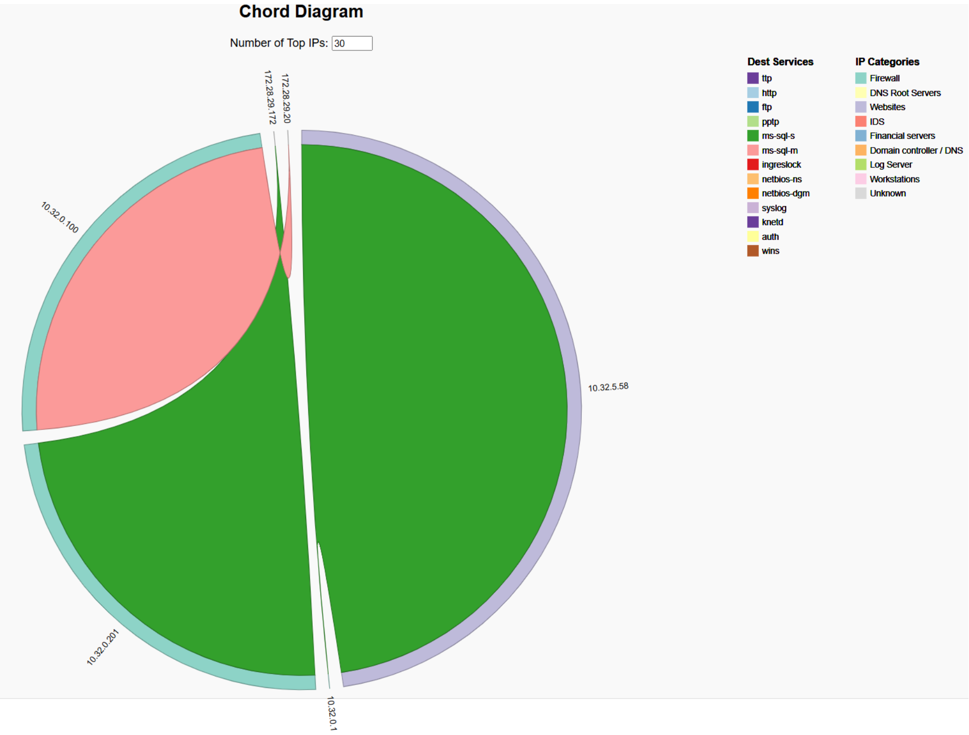
IP Address
Enter IP

Source/Destination

□ Show All Destination Services

Generate

Fig. 4 Attacks to the DBs using backdoor vulnerabilities



Chord Diagram

Number of Top IPs: 30

Fig. 5 destination IPs of the ms-sql-s exploit

- **Firewall outage, data exfiltration and Ingreslock vulnerability**:
  At 17:21:50 on April 6th, the firewall is down due to unclear reasons. In fact, we can see a sudden drop to zero in the number of network packets analyzed by it. Shortly after this, the IDS logs show a spike in the connections labeled as `NET POLICY DNS Update From External net` between 17:26 and 17:29. All the source IPs are internal from the net and target the DNS using port 53, which is strange given

the classification by the IDS. This is likely an attempt from the attackers to exfiltrate data now that the network defenses are not operational. In the following visualization, we focus on the time period during which the attack occurred. The IP involved are highlighted in green, and it shows that are connected to the DNS. This pattern highlights the attackers' likely strategy to exploit the downtime of the firewall and exfiltrate data through DNS requests, particularly targeting port 53.

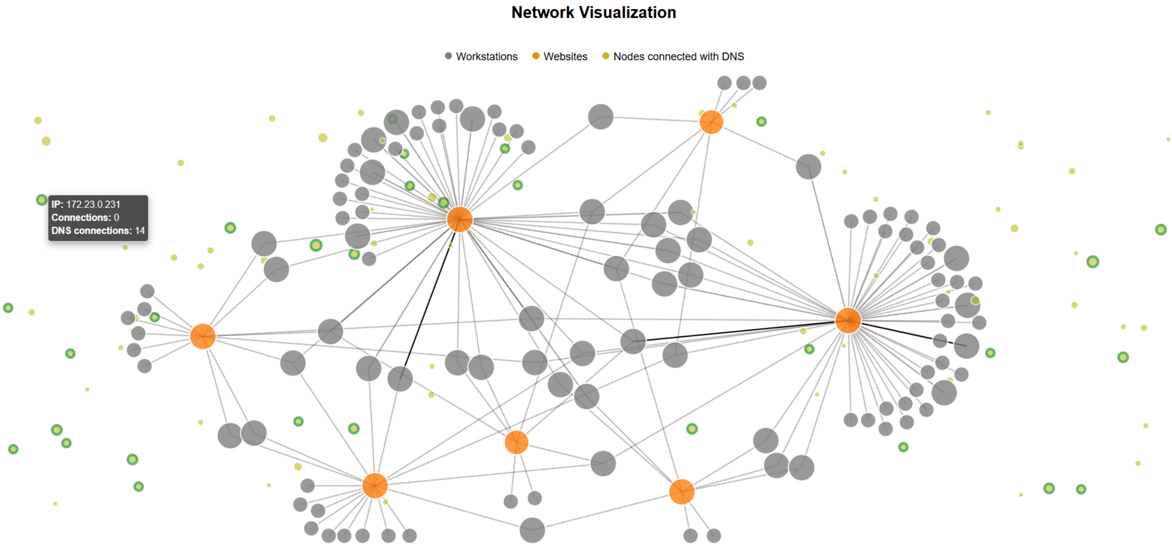
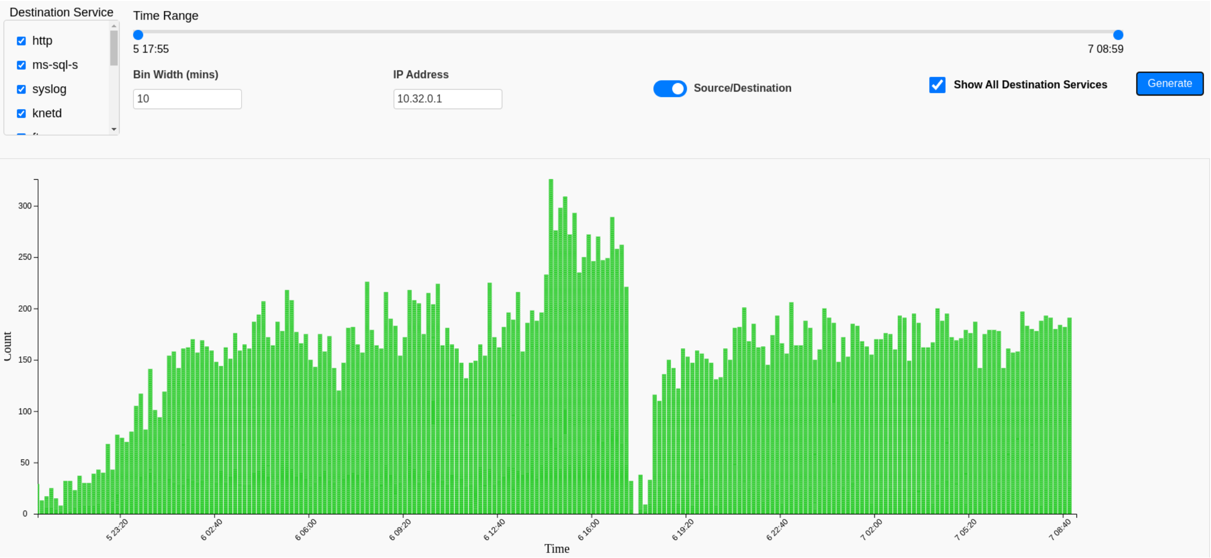


Fig. 6 Visualization of IP Connections During the Firewall Downtime

The company manages to bring the firewall back up in a few minutes, precisely at 17:40, but the damage is already done, since the attackers were free to operate and apparently very active during this short time frame. Moreover, at 18:16:59 of the 6th the attackers proceed in using an `ingreslock` vulnerability to gain root access to the internal servers. This vulnerability stems from the fact that the TCP port 1524 was commonly used as a backdoor listening port by attackers after successfully compromising a system, which has happened following the previous attack. The simplicity of exploiting this vulnerability lies in the ability to gain root-level access to the target machine by simply connecting to the open port. While this has been known for a long time, legacy system like the one of the bank still have it enabled. Even though the malicious connection is denied once again, this fact highlights the lack of basic security measures in place.


Destination Service

☑ http

☑ ms-sql-s

☑ syslog

☑ knetd

Time Range: 5 17:55 – 7 08:59

Bin Width (mins): 10

IP Address: 10.32.0.1

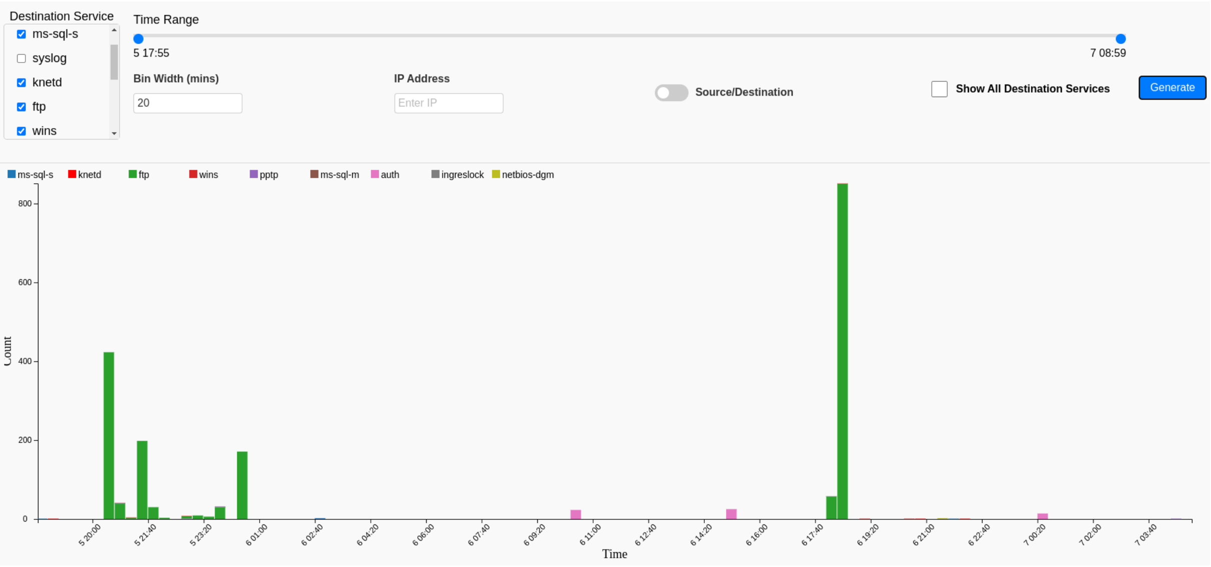
Source/Destination

☑ Show All Destination Services

Generate

Fig. 7 Firewall activity and downtime

- **Spike in FTP connections and botnet expansion following firewall restoration**: After some minutes, the firewall comes back up, contextually to a huge spike in the FTP connections, at around 18:30. This activity clearly violates the bank policy and can be interpreted as several attempts of information leakage by the botnet, which has expanded noticeably. All the connections are blocked by the firewall, which is doing a good job at limiting the potential damage suffered by the bank. The 286 IPs involved in this activity are again all internal to the network, once again confirming the hypothesis above. The seven destination IPs correspond to websites, which are accessed through the port 21. Although the attackers did not manage to get access directly to the databases, but only to data stored on individual workstations, they still gained access to potentially critical data.


Destination Service

☑ ms-sql-s

□ syslog

☑ knetd

☑ ftp

☑ wins

Time Range: 5 17:55 – 7 08:59

Bin Width (mins): 20

IP Address: Enter IP

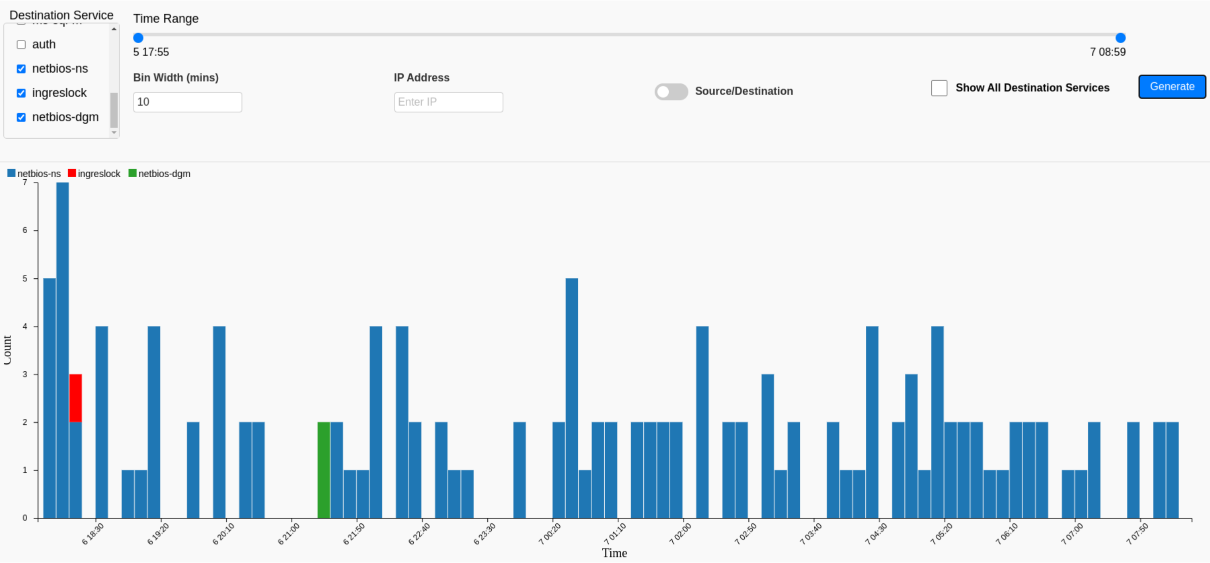
Source/Destination

□ Show All Destination Services

Generate

Fig. 8 Attempted FTP connections

- **Increased malicious activity, potential DDoS threats, NBT-NS poisoning**: The last noticeable events we observed are the following. First, a sharp increase in the activity of IP 172.23.252.10, which was dormant before the attack to the firewall. After this event, it becomes extremely active and takes the spot as the IP which attempted the most connections. A large portion of these, about 40%, are TCP connection that target 6667 as destination port, which is again in violation of the rules. All of these connections are to seven websites and half of them are successfully built, while the other half is denied by the ACL (Access Control List). It is known that a common use of botnets is DDoS (Distributed Denial of Service) attacks, which may be our case. As previously said, this service has built-in transfer capabilities, so the attackers may be once again transferring data to external servers. Secondly, right after the firewall malfunction, connections using the `netbios-ns` protocol start appearing. They all use UDP packets and build short-lasting connections, which are torn down after less than two minutes. This could hint to NBT-NS poisoning, a technique to obtain the credentials of the users within the network, which can be used to further escalate the attack. On the other hand, it could be yet another way the attackers are using to build connections to the outside and transfer the obtained data, since the FTP connections have all been blocked.



Destination Service

□ auth

☑ netbios-ns

☑ ingreslock

☑ netbios-dgm

Time Range: 5 17:55 – 7 08:59

Bin Width (mins): 10

IP Address: Enter IP

Source/Destination

□ Show All Destination Services

Fig. 9 Usage of NetBIOS-NS service after firewall downtime

3.1 Visualization in D3.js

To visualize the insights derived from the data analysis, we implemented three interactive visualizations using D3.js. Each visualization has a different purpose.

- **Streamgraph for Connection Classifications**: We created a streamgraph to show how network connection types change over time, such as normal, suspicious, or critical connections. Users can pick specific time ranges, and this updates the other visualizations to focus on the selected time period.
- **Heatmap for Source and Destination IPs** A heatmap was made to show the communication patterns between source and destination IP addresses. It helps users spot frequent connections or clusters of activity. Hovering on a cell displays the corresponding number of connections and the relative attributes by the means of an html tooltip. Clicking on a connection in the heatmap highlights the related nodes and edges in the network graph.
- **Graph Network for Visualizing Connections** We used a graph to represent the network. Nodes are IP addresses, and edges are connections between them. The graph automatically updates based on the time range brushed on the streamgraph, and highligth commections based on the cells selected on the heatmap, so they only see the relevant nodes and connections for their focus.
- **Histogram for Firewall Dataset** The histogram is a simple and widely used concept, but still very effective in the way it conveys information. It is especially suited for this datasets, as it is capable of clearly displaying trends, sudden changes, isolated and suspicious events through proper filtering. We offer a highly configurable histogram with a dedicated control bar, where the user can select several options. First of all, the time range to be covered by the means of a slider, as well as the destination services of interest that can be selected via a check list. These are a subset of all the existing ones, which

includes only those that are significant for the analysis. Then, the user has the possibility of selecting a single IP address as either source or destination, in order to study the behavior of a single connection point. We also give the user the possibility to select all the possible destination services, including those that are not present in the dedicated check list, in order to have a global overview. Lastly, the user can select the bin width in terms of minutes, so that the visualization aggregates the data according to their needs. We opted for bin width over the number of bins since it is a better indicator of the chart granularity. The histogram can display all the major events concerning the firewall, effectively showing its downtime, as well as specific protocol usage over time.

4 Answers to Mini-Challenge 2

Questions 2.1 and 2.2 were addressed in the Section 3. We can briefly summarize the latter, then proceed in answering the questions 2.3.

4.1 Question 2.2

Since the beginning of the logs, it appeared quite clear that some workstations internal to the network were already compromised. The attackers used the access to these machines to exfiltrate information that was present in the local file system. The large use of the `6667_tcp service` is quite symptomatic of this fact. There are a handful of attempts to gain root access to the databases of the bank, which are all blocked by the cyber defenses. At some point, the firewall is brought down due to unknown reasons, we did not find any clear indication that the attackers caused this. It is unlikely that the security administrators chose to do this on purpose, since the firewall and IDS already showed malicious activities in the previous 24 hours. During the downtime, the IDS detects a peak of suspect connections, which probably correspond to sensitive data transfer to external servers. After this, the hackers attempy yet another attack using an `ingreslock` vulnerability, which is blocked by the firewall. We also see a spike in the FTP connections, which are all denied, as well as a sudden increase in connections using the `netbios-ns` protocol, possibly indicating a NBT-NS poisoning attack.

4.2 Question 2.3

The security standards of the bank network are very low, since they exposes flaws and use bad practices that have long been known. First of all, allowing traffic over port 6667 is a terrible and dangerous choice, especially due to its high usage and widely known security risks. Moreover, it violates the bank policy, so the firewall should block all those connections, or at least notify the administrators to take action against them. From the logs, it is clear that malicious intruders have control over several workstations and activate them during the night. It is frankly embarrassing that such activity goes unnoticed and the infected machines are not disconnected from the network and neutralized. The attempted use of known exploits and backdoors, for a total of at least six times, should raise a general alarm, which it clearly does not. The most notable and yet unexplained event is the firewall being offline for approximately 20 minutes. We failed in determining whether or not the attackers caused this. It might still be a possibility that the administrators scheduled a maintenance or an update that required rebooting it. This would be the culmination of the lack of knowledge and preparation of the security admins, which seem to be oblivious of what is happening. Also, there is a number of bad practices: the increasing usage of the UPD protocol, especially for the `netbios-ns` service, the usage of old protocols such as `PPTP`, known for their weak encryption, and leaving ports such as 22, 1443, and 1524 open. The bank could adopt a whitelist for all the ports that are considered safe and block all the others, thus greatly reducing the potential attacks and security breaches. Finally, it would be a good idea to make sure that all the individual workstations do not have access to sensitive data and work in user-mode, thereby limiting the potential damage caused by employees' mistakes and single machine infection. To summarize, the overall network security of the Bank of Money is not sufficient and proves incapable of withstanding cyber attacks, at least partially. The network administrators are not responding adequately to the intrusion attempts and are lacking basic knowledge on how to counter such events.