Performance Evaluation of NPA-VM Using Attack Graph Hierarchical Visualization Approach

Abhishek Pipliya, Sachin Chirgaiya

PG Scholar, Asst. Professor

Computer Science & Engineering Department

Oriental University, Indore, India

Abstract**— Internet attacks are on the rise and pose serious security threats to enterprise networks, commercial websites and to the millions of home internet users. Internet attacks are becoming more potent and complex with time. Network traffic visualization tools have successfully enabled security analysts to understand the nature of traffic present in a network. Conversely, these tools rely mainly on human expertise to discover anomalies in traffic and attack patterns. Human capacity to comprehend massive amounts of time-varying data is limited and network visualization tools need further visual aid to extract interesting patters from such large and complex data sets. Our approach is to search and highlight user-specified graph patterns in network traffic logs[1]. By visualizing a set of simple graph patterns, analysts can put together visual pieces of information conveyed by these smaller patterns and can learn about larger and more complex patterns. Theatrical performance of network traffic pattern in graphic language is visually intuitive, powerful and flexible specification and overcomes the limitation of poor pattern specification formats existing in the current tools. Therefore, our approach gives way to an iterative visual investigation and enables rapid discovery of more sophisticated attack patterns and anomalous features which are otherwise undetectable by standard network traffic visualization tools. [2]**

Keywords**—NPA-VM (Network Pattern Analysis Based Vulnerability Measurement), Forecasting, Transformation, forecasting.**

I. INTRODUCTION

An Internet attacks are growing at an alarming rate, becoming more potent and complex with time. These attacks pose serious security threat to enterprise networks, commercial websites and to the millions of home internet users. According to the Worldwide Infrastructure Security Report, a survey of 70 of the biggest net operators in North America, South America, Europe and Asia found that malicious attacks were rising sharply and that the individual attacks were growing more powerful and sophisticated [3]. There are plenty of automatic intrusion detection technologies available, but they have some inherent weaknesses. Anomaly detection based intrusion detection systems often generate a huge number of false alarms which overwhelm security engineers. Known attack signatures based intrusion detection systems lack the ability to detect new or unknown attacks. Thus, automatic intrusion detection systems alone are not sufficient for network security and security engineers are needed to cope up with malicious attackers [4].

Discovery and Search are the two main tasks of a security Engineer. Discovery is the process of finding patterns in some data and Searching is the process of determining if a particular pattern exists in a data set. Patterns are basically spatial and temporal structures in the data or correlation in the fields of the data. In network traffic, presence of patterns in the data indicates some attack. Automated discovery of patterns is not sophisticated and robust enough in the network security realm, owing to the growing complexity of internet attacks. Thus, though the task of Searching is efficient and fast on machines, Discovery is still a very human oriented task. The human eye has been frequently advocated as the ultimate data-mining tool and can recognize and infer patterns from visual data intuitively. Human security engineers can look through the data to create signatures for new attacks and visualization has often helped security analysts in this task of theirs [5].

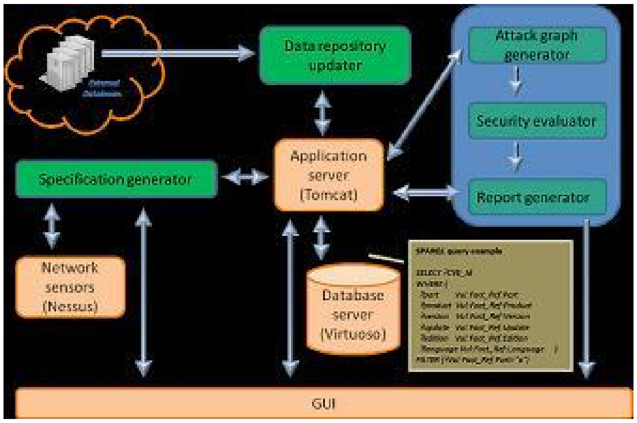


Fig. 1 Network infrastructure

Network data visualization has successfully enabled security analysts to understand the nature of traffic present in a network, to identify unusually high traffic and track down points in the network that create such a big traffic. Network traffic visualization also sometimes aids analysts to detect internet attacks like denial of service attacks and effects of worms. However, extracting complicated patterns from large amounts of time-varying data is tedious for humans as their capacity to comprehend large amounts of data is limited. Conventional network traffic visualization systems provide a rudimentary level of visual display of results and rely mainly on human expertise for finding anomalies in traffic. Thus, they can be used for detecting only large-scale internet attacks and cannot aid the discovery of subtle and more sophisticated attack patterns [6].

There has been little work on enhancing visualization to enable security analysts to discover non-trivial patterns in network traffic logs. Current network traffic visualization tools lack the key functionality of searching and visualizing spatial and temporal patterns in network traffic data. The patterns supported by most of the current visualization tools are restricted to represent constraints on the IP addresses, ports, protocols and other attributes of a single record/flow. The language/format that the pattern is represented in limits the types of patterns that can be found. One cannot specify, the more general class of patterns that can encode “who talks to whom in what order” information. Enhancing these tools with the generic pattern analysis capability makes them more practically used by network administrators.

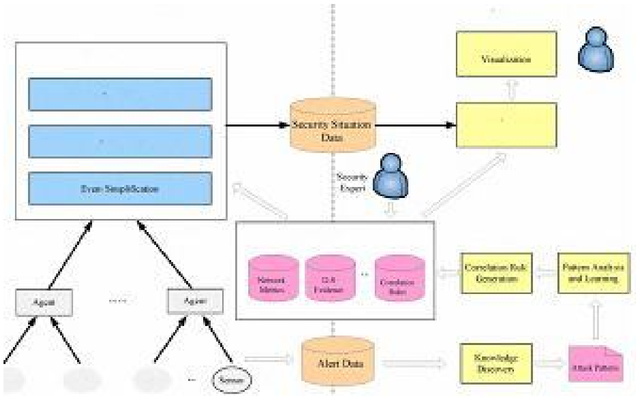


Fig. 2 User Attribute Detail

The main contribution of our work is the introduction of graph language as a simple, powerful, flexible and visually intuitive specification of spatial and temporal patterns. Our approach is to search and highlight user-specified graph patterns in network traffic data. By visualizing a set of simple graph patterns, analysts can put together visual pieces of information conveyed by these smaller patterns and can learn about larger and more complex patterns. Thus, the discovery of complex time-varying patterns will be an iterative investigation of attack patterns, facilitating detection of more sophisticated attacks than are allowed by current visualization tool [7].

Graph based network traffic visualization tools are very popular for network monitoring as they affords an intuitive visualization of information. Chart based network visualization tools depict data in a very aggregated form as opposed to log files that overwhelm a user with too much detail. Graph-based tools, on the other hand, provide a balanced level of detail. Network patterns also have an intuitive meaning in case of network traffic graphs. A network pattern is simply a sub graph of the network traffic graph and can thus be specified easily in the graphic language.

It describes the various parts of our implementation, including the pattern preserving layout and graph isomorphism algorithm. It visualizes various attack signatures as graph patterns which can be used as templates while searching for the presence of those attacks. In this section we have used our method to understand two real world attacks and we demonstrate how effective graph pattern visualization was in understanding those attacks. We conclude and talk about some of the future work in sections [8]

II. RELATED STUDY

There are several graphed-based visualization tools in the network traffic monitoring domain. There are both real time monitoring solutions and offline interactive tools that support different time slots comparison and analysis. These approaches, however, do not support graph pattern search and visualization feature and thus are not well suited for finding sophisticated time varying patterns.

NVisionIP [9] is a chart-based interactive network flow visualization tool. It does support searching for patterns in flow data, but the patterns are restricted to represent constraints on the IP addresses, ports and protocols of a single record/flow. Searching for a pattern involves finding all records/flaws in a set of Netflow that satisfy the constraints of the pattern. Reduces the time required for multiresolution data filtering and querying of network data using compressed bitmap indexing. The n-dimensional data is displayed using n-dimensional histograms. In this approach too, the queries represent constraints on attributes of flows like IP address, port and protocol. In these approaches, one cannot express the more general class of spatial and temporal patterns, i.e. Patterns specifying “who talks to whom in what order” information. That graph based approach has the advantage that one can specify arbitrary network traffic patterns spanning any number of nodes over time by constructing a sub graph representing it [10] It proposes a visual analysis with a declarative knowledge representation based on first-order logic. Though the declarative language used makes pattern specification more flexible as compared to that in and does capture some temporal relations, but since it involves fixed number of predicate overflow variables (individual flow records), it is not as powerful or intuitive as our sub graph pattern specification. The use of first-order declarative language also makes the pattern specification cumbersome and lengthy for analysts. For even simple patterns, the specification tends to involve many clauses and is thus quite complicated. [11]Visual data mining (VDM) focuses on using visualization to describe the data and patterns created by data mining algorithms. The visualization allows the user to discover more patterns in the data, evaluate the current patterns and then re-run the data mining algorithms with different inputs. The goals of VDM are similar to ours, but while they focus on modifying the working of a data mining algorithm, it focus on highlighting the user-defined patterns to facilitate the discovery of more complex pattern.

III. Problem Domain

The Network security situational awareness is a data analysis based approach requires a massive processing of information. It varies according to different devices, their transformations and integrations in the network. The aim is to increase the data availability by making the system more robust and reliable. In such scenarios, information processing is based on a fusion of network factors and parameters which is used to make the preventive assessment of the situation [12]. The aim is to detect the unusual patterns and from this predict the future effects of the attacks on mentioned devices. After studying the various existing approaches in the different areas of the network used for predictions and forecasting, this work had identified that analyst has to know the patterns in a restricted manner and the detection is totally based on logical capabilities of few of those. Thus, some automation is required for better understanding of vulnerabilities and effects of attacks. Here are the some identified issues in existing approaches for resolving the issues of vulnerability analysis.

Problem 1: All the existing systems will consider vulnerability in a qualitative aspect rather than some quantitative aspects which mislead the analyst's.

Problem 2: Real time measurement is not given by which losses are comparatively larger than others.

Problem 3: Massive data processing some time generates false alarm and incorrect predictions thus prediction accuracy needs to be considered as primary parameters for the work.

Problem 4: The assessment used to classify network state and the level of information required for optimal illustration is not complete always which misguide the calculation. Thus the transformation of such information with certain attributes is not provided by any of the existing mechanisms.

Pattern to be searched for in the network traffic graph can be specified as a subgraph in the DOT format [13]. For example, to search for a denial of service attack pattern, one can specify a graph where there are a number of nodes attacking (sending packets to) a single victim node around the same time. Note how one can specify various attributes in this specification. If a node or edge attribute matching is activated, the specified attributes are matched while finding patterns in the input graph. Enabling attribute matching provides a lot of flexibility in composing patterns. Attributes can be composed of the following types: string, position coordinates (pair of comma separated real numbers

IV. Proposed NPA-VM Approach & Algorithm

The pattern to be searched for in the network traffic graph can be specified as a subgraph in the DOT format [6]. For example, to search for a denial of service attack pattern, one can specify a graph where there are a number of nodes attacking (sending packets to) a single victim node around the same time. It shows such a DOT graph. Note how one can specify various attributes in this specification. If a node or edge attribute matching is activated, the specified attributes are matched while finding patterns in the input graph. Enabling attribute matching provides a lot of flexibility in composing patterns. Attributes can be composed of the following types: string, position coordinates (pair of comma separated real number.

Security is the means of achieving confidentiality and privacy with robust data transmission and availability. For effective communication over the network, it could be treated as critical factor and must be monitored continuously. The network is a big working environment made from a collection of various devices, protocols, servers and host parallel generating thousands of records per unit time. Processing of such huge amount of data is a complicated task and requires more efforts in terms of time and cost. Thus, this paper provides an alternative way of handling security by vulnerability assessment. According to the approach, network components are analyzed on their previous activities and changes accommodated. These factors should be permitted or rejected accordingly to their probability of attack vulnerable values called as assessment values.

Higher be the generated value larges be the attack occurrence probability and smaller be the value less probable to attack. Representation of a component of this network pattern analysis based vulnerability measurement is given by attack graph. There are some benefits of using the metrics in this work given here as:

1) Improved performance and protection level of the system

2) Monitoring model which compares the current values with the ideal values after which validation of operations and changes is measured.

3) Contribute to the enhancement of the existing security practices and to the integration of information security to its business process values.

4) There has been little work on enhancing visualization to enable security analysts to discover non-trivial patterns in network traffic logs.

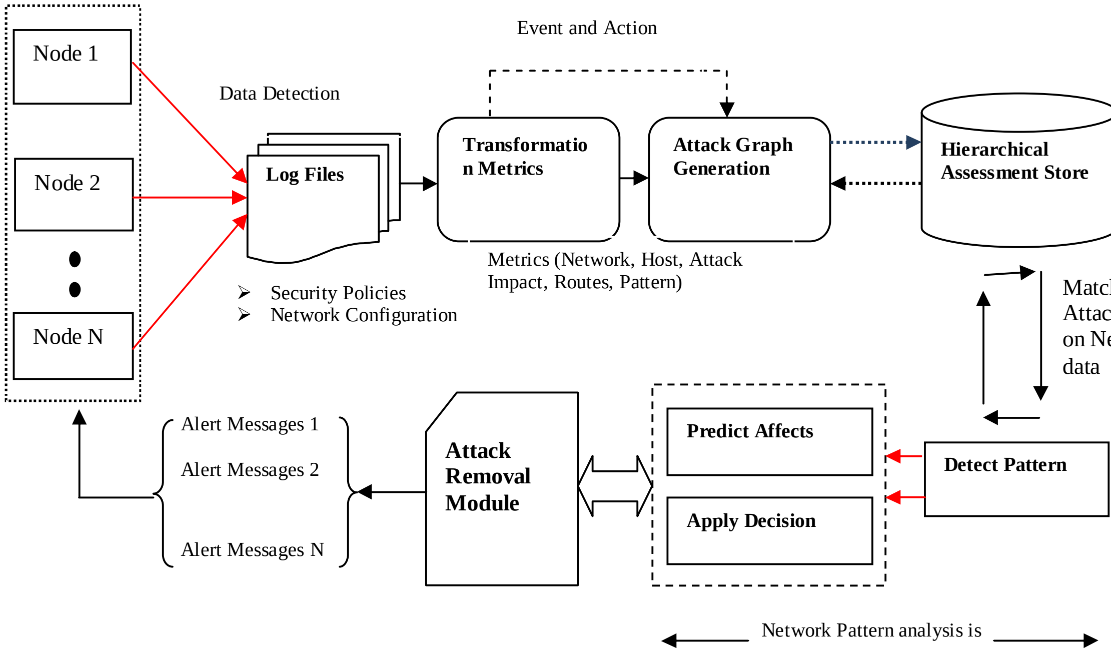


Fig. 3 AN Efficient Network Pattern Analysis Based Vulnerability Measurement (NPA-VM)

Algorithm:

1. Evaluate Packet dataset

Step 0: Initiate

Step 1: Get the outclass Dataset

Step 2: Evaluate all the rows and column and set in the array.

Step 3: After that assign all individual packet information in a line.

Step 4: Stop

2. Create an alert object

Step 0: Initiate

Step 1: Get the data packet in a line

Step 2: Examine srcip, dstip, srcport, dstport, type, time from the line for each packet information.

Step 3: Form an alert object

Step 4: Stop

3. Time and Space limit Analysis (TSLA)

Step 0: Initiate

Step 1: Get each and every alerts

Step 2: It is used for all alert which are critical and manic check the condition if(sip(AI)=sip(AJ) and dip(AI)=dip(AJ) and time(AJ)<time(AI) and time(AJ)-time(AI)<TW) if yes then goto step 6 otherwise go to step 2.

Step 5: Set (AI,AJ)in alert pair.

Step 6: Stop

4. Uniform attack Graph Generation

Step 0: Initiate

Step 1: Get the set of alert pair.

Step 2: Generate a node set N which consists of AI of each alert brace.

Step 3: Create a frame set E which consists of (AI, AJ) of all alert brace.

Step 4: For each edge (Ni, NJ) check the condition of an circumlocutory path Ni, NK, NJ then delete (Ni, NJ) from the frame set E and return graph G (N,E).

Step 5: Stop

5. Generation of Alert Device Evaluation Matrix

Step 0: Initiate

Step 1: Acquire Number of devices and alerts.

Step 2: For every alert in rows create a set E such that E (AI, DJ).

Step 3: Accumulate it in a hash set matrix.

Step 4: Stop.

6. Computation of Unit Risk Evaluation (URE)

Step 0: Initiate

Step 1: Find a exacting device and its alert set E from the matrix of module 5.

Step 2: Find alert level l(A) and device level l(D).

Step 3: Determine $EAD= 5l(A)-1 * 5l(D)-1$. Were *A refers* to the alert generate by IDS for corresponding attack; *D* represents the device which is attacked; *l* (*a*) and *l* (*d*) represent the levels of the alert and the particular device.

Step 4: Stop

7. Evaluation of Attack Risk Evaluation (ARE)

Step 1: Initiate

Step 2: Obtain all the URE of all the devices.

Step 3: For each URE Calculate, EA = EA + EADI.

Step 4: Stop

8. Calculation of Device Risk Evaluation (DRE)

```
Step 0: Initiate
Step 1: Obtain all the URE of all the devices.
Step 2 : For each URE calculate, ED = ED + EAJD
Step 3 : Stop.
```

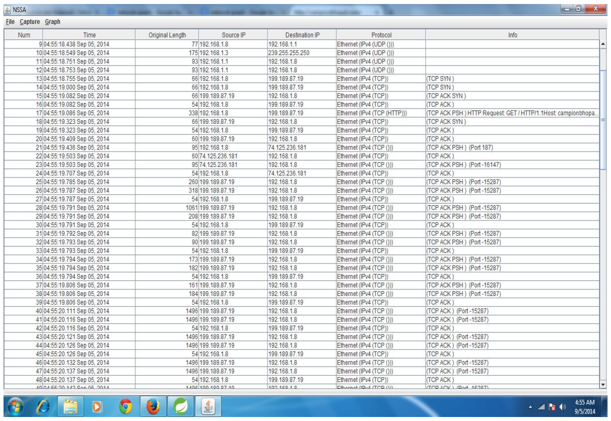
9. **Calculation of Network Risk Evaluation (NRE)**

```
Step 0: Initiate
Step 1: Obtain all the URE of all the devices.
Step 2: For each URE calculate, EN = EN + EAJ OR EN = EN +EDI
Step 3: Stop.
```

V. Result Evaluation

Logging table:

This table provides the uniform approach for data enter in the network with information about time, Source IP, Destination IP, Server information this entire field comes under this table approach.



Num	Time	Original Length	Source IP	Destination IP	Protocol	Info
9	04:55:18.438 Sep 05, 2014	77	192.168.1.8	192.168.1.1	Ethernet (IPv4 (UDP ()))	
10	04:55:18.549 Sep 05, 2014	175	192.168.1.3	239.255.255.250	Ethernet (IPv4 (UDP ()))	
11	04:55:18.751 Sep 05, 2014	93	192.168.1.1	192.168.1.8	Ethernet (IPv4 (UDP ()))	
12	04:55:18.753 Sep 05, 2014	93	192.168.1.1	192.168.1.8	Ethernet (IPv4 (UDP ()))	
13	04:55:18.755 Sep 05, 2014	66	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP SYN)
14	04:55:19.000 Sep 05, 2014	66	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP SYN)
15	04:55:19.082 Sep 05, 2014	66	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP))	(TCP ACK SYN)
16	04:55:19.082 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
17	04:55:19.086 Sep 05, 2014	338	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP (HTTP)))	(TCP ACK PSH) HTTP Request: GET / HTTP/1.1Host: campionbhopa...
18	04:55:19.323 Sep 05, 2014	66	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP))	(TCP ACK SYN)
19	04:55:19.323 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
20	04:55:19.409 Sep 05, 2014	60	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP))	(TCP ACK)
21	04:55:19.436 Sep 05, 2014	95	192.168.1.8	74.125.236.181	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port 187)
22	04:55:19.503 Sep 05, 2014	60	74.125.236.181	192.168.1.8	Ethernet (IPv4 (TCP))	(TCP ACK)
23	04:55:19.503 Sep 05, 2014	95	74.125.236.181	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -16147)
24	04:55:19.707 Sep 05, 2014	54	192.168.1.8	74.125.236.181	Ethernet (IPv4 (TCP))	(TCP ACK)
25	04:55:19.785 Sep 05, 2014	260	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
26	04:55:19.787 Sep 05, 2014	318	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
27	04:55:19.787 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
28	04:55:19.791 Sep 05, 2014	1061	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
29	04:55:19.791 Sep 05, 2014	208	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
30	04:55:19.791 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
31	04:55:19.792 Sep 05, 2014	82	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
32	04:55:19.793 Sep 05, 2014	90	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
33	04:55:19.793 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
34	04:55:19.794 Sep 05, 2014	173	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
35	04:55:19.794 Sep 05, 2014	182	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
36	04:55:19.794 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
37	04:55:19.806 Sep 05, 2014	161	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
38	04:55:19.806 Sep 05, 2014	184	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK PSH) (Port -15287)
39	04:55:19.806 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
40	04:55:20.111 Sep 05, 2014	1496	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK) (Port -15287)
41	04:55:20.116 Sep 05, 2014	1496	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK) (Port -15287)
42	04:55:20.116 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
43	04:55:20.121 Sep 05, 2014	1496	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK) (Port -15287)
44	04:55:20.126 Sep 05, 2014	1496	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK) (Port -15287)
45	04:55:20.126 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
46	04:55:20.132 Sep 05, 2014	1496	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK) (Port -15287)
47	04:55:20.137 Sep 05, 2014	1496	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK) (Port -15287)
48	04:55:20.137 Sep 05, 2014	54	192.168.1.8	199.189.87.19	Ethernet (IPv4 (TCP))	(TCP ACK)
49	04:55:20.142 Sep 05, 2014	1496	199.189.87.19	192.168.1.8	Ethernet (IPv4 (TCP ()))	(TCP ACK) (Port -15287)

Fig. 4 Logging network detail

Network graph Description:

In this graph it shows a central node pointed blue can be connected with different node in a uniform manner, different nodes represented by pink dots

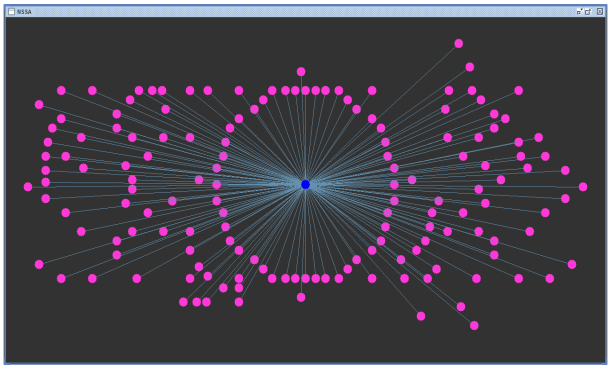


Fig. 5 Network graph

Node detail graph:

This graph shows details of different nodes in a loop, it provides information on IP Address from where it originated, it also provides information about the packet size and on which protocol it is based

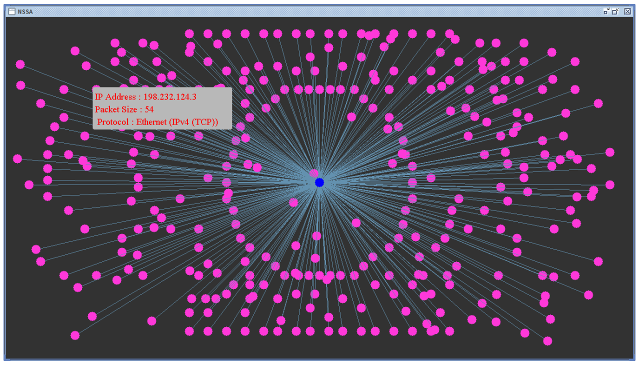


Fig. 5 Showing Node detail

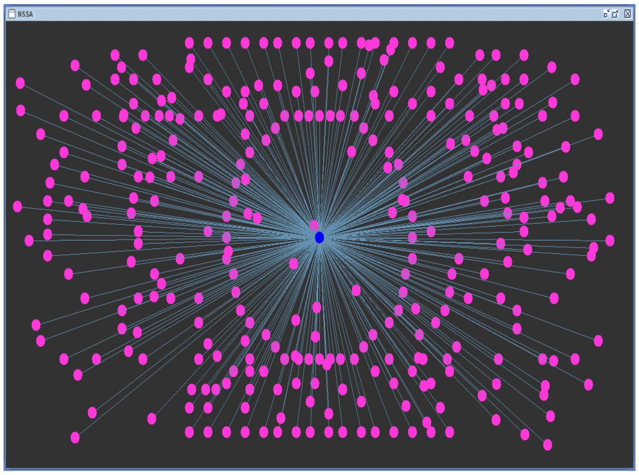


Fig. 6 Showing Node detail

VI. Conclusion

In this paper, we developed techniques to extract useful information about the network situation from alerting, while most of existing NSA systems just focuses on net flow data. We proposed an approach to automatically correlate the alerts to generate a simple attack graph based upon time and space restriction. The graph helps to administrate to understand the attack steps simply. This approach can discover new alert relations and it does not depend on background knowledge. At last. The simulations showed that with the proposed methods NPA-VM system can efficiently analyze large amount alerts and save administrators' time and energy also [14]. We have presented a network traffic analysis system that supports graph pattern matching and visualization. Graphical language is a highly intuitive, flexible and general pattern specification format that captures temporal and spatial events in network traffic. By putting together visual pieces of information conveyed by smaller patterns, security analysts can discover more complex and sophisticated attack patterns.

VII. Future Work

Some problems and concepts that remain unaddressed can be performed in the future. This system can further be extended to implement NPA-VM scheme in real-time networks where it has to deal with the unwanted attacks. It is judged by the approach which can be added to exact, timely analysis based on graph generation which can solve the problem easily. We are also working towards embedding the developing source code of our proposed scheme in the cloud based network. In our proposed scheme so as to use the benefits of an approach like open source.

References

[1] E. Bethel, S. Campbell, E. Dart, K. Stockinger, and null Kesheng Wu.Accelerating network traffic analytics using query-driven visualization.Symposium On Visual Analytics Science And Technology, 0:115–122, 2006.

[2] L. P. Cordella, P. Foggia, C. Samson, and M. Veneto. A (sub) graph isomorphism algorithm for matching large graphs. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(10):1367–1372,2004.

[3] J. Ellison, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull. Graphviz - open source graph drawing tools. Graph Drawing, pages483–484, 2001.

[4] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force directed placement. Software: Practice and Experience, 21(11):1129–1164, 1991.

[5] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force directed placement. Softw. Pract. Expert. 21 (11): 1129–1164, 1991.

[6] E. Gansner, E. Koutsofios, and S. North. Drawing graphs with dot.http://www.graphviz.org/Documentation/dotguide.pdf.

[7] M. Garey and D. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freema, 1979.

[8] G. GU, R. Perdisci, J. Zhang, and W. Lee. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnetdetection. In Proceedings of the 17th USENIX Security Symposium(Security'08), 2008.

[9] Fang Lan, Wang Chunlei, and MaGuoqing , "A Framework for Network Security Situation Awareness Based on Knowledge Discovery" 2010 2nd International Conference on Computer Engineering and Technology 2010 IEEE.

[10] Juan Wang,Feng-li Zhang,Jing Jin,Wei Chen, "Alert Analysis and Threat Evaluation in Network Situation Awareness" 2010 IEEE.

[11] Cyril Onwubiko, "Functional Requirements of Situational Awareness in Computer Network Security" 2009 IEEE.

[12] Liu Mixi, Yu Dongmei and Zhang Qiuyu et aI., "Network Security Situation Assessment Based on Data Fusion, " 2008 Workshop on Knowledge Discovery and Data Mining, 2008

[13] Wang Huiqiang, Lai Jibao, and Ying Liang, "Network Security Situation Awareness Based on Heterogeneous Multi-Sensor Data Fusion and Neural Network, " Second International Multisymposium on Computer and Computational Sciences, 2007 IEEE.

[14] Mr. Marc Grégoire, "Visualization for Network Situational Awareness in Computer Network Defense" (2005). In Visualization and the Common Operational Picture (pp. 20-1 - 20-6). Meeting Proceedings RTO MP-IST-043, Paper 20. Neuilly-sur-Seine.

[15] Mr. Abhishek Pipliya and Mr. Sachin Cirgaiya, "Network Pattern Analysis based Vulnerability Measurement using Attack Graph Hierarchical Visualization Approach" In International Journal of Computer Applications (0975–8887) Volume 99 – No.1 1, August 2014 (pp. 45-50)