Predicting Multi-Stage Attacks Based on IP Information

Abdulrazaq Almutairi
School of Electronic, Electrical and Systems Engineering
Loughborough University
Leicestershire, UK LE11 3TU
a.almutairi@lboro.ac.uk

David Parish, James Flint
School of Electronic, Electrical and Systems Engineering
Loughborough University
Leicestershire, UK LE11 3TU
(d.j.parish, j.a.flint)@lboro.ac.uk

Abstract—Multi-stage attacks can evolve dramatically, causing much loss and damage to organisations. These attacks are frequently instigated by exploiting actions, which in isolation are legal, and are therefore particularly challenging to detect. Much research has been conducted in the multi-stage detection area, in order to build a framework based on an events correlation approach. This paper proposes a framework that predicts multi-stage attacks based on a different approach, which is an IP information evaluation. This approach was chosen after analysing three different multi-stage attack scenarios. This paper shows the analysis of those scenarios, detailing their steps and information hitherto unexploited in current intrusion detection systems. The paper also details the results obtained in the evaluation process, including detection and false positive rates.

Keywords-Intrusion detection; Multi-Stage attacks; IP Check; Data mining; Fuzzy Logic

I. Introduction

Multi-stage attacks have a significant impact on organisations. They have been described as the most challenging set of attacks to investigate and detect [1].These attacks occur through multiple phases to get access to an organisation. Most of those attacks involve three phases. In the first phase, attackers try to analyse available information about the target, to find vulnerabilities and weaknesses that can be exploited. In the second phase, attackers exploit the weaknesses found in the first phase to inject malware into, or to gain access to, the system. In addition, they try to get more details and conduct a deep analysis about the system to find data or resources in which they have an interest. In the final phase, after gaining access, attackers are in a position to destroy the system or steal valuable information [2]. Different solutions have been introduced to detect multi-stage attacks, some of those being event correlation-based. Event-correlation based solutions try to match network events with certain attack patterns. When a stream of network events matches a certain pattern, attacks can be stopped before progressing to the next stages. Many researchers claim the effectiveness of that approach in detecting multi-stage attacks. However, this approach requires having a-priori knowledge of the multi-stage attack pattern (sequences), which is not always feasible since discovering new complex attacks normally takes some time. The Shady Rat Operation attack is a good example of that; it started in 2006 and was only discovered in 2011[3].Thus, it has been decided to follow a different approach in this research, rather than network events correlation when proposing a solution for predicting multi-stage attacks. The approach is based on evaluating the reputation of IP addresses participating in network traffic using the fuzzy logic. The fuzzy logic works based on define rules to produce an output. Based on specified rules, the fuzzy logic decides whether we need to stop the traffic with evaluated IP addresses to block potential attacks.

Section 2 provides a brief background of the fuzzy logic, social engineering, and CRLF (carriage return line feed) injection. Section 3 provides an analysis of three different multi-stage attack scenarios that help in understanding the behaviour of multi-stage attacks. The first scenario is about communication with a bad DNS server and how that has been employed by an attacker to register machines to its bot army. The second scenario discusses the Shady Rat attack, which is a good example showing how social engineering can be employed to target an organisation. The third scenario shows how header splitting can be employed by an attacker to target a network connected to a web host running a web application. Section 4 goes through the proposed solution. Section 5 discusses the evaluation process detailing the evaluation approach and results obtained in different evaluation phases. Section 6 goes through some related work. Section 7 provides the conclusion and future work of this paper.

II. Brief Background

A. Fuzzy Logic

Fuzzy logic is a computational approach based on human language rules. The fuzzy systems translate the defined rules to mathematical equivalents [4]. Those systems, as shown in Fig. 1, consist of a fuzzifier, inference engine, rules base, and defuzzifier. Fuzzy systems work as follows [4]:

The fuzzifier converts crisp inputs to a fuzzy set by using specified membership functions for each input.

Based on the defined rules, the inference engine produces a fuzzy output.

The fuzzy output is converted to a crisp value using the membership functions defined for defuzzification.

Fuzzy logic is suitable for ambiguous scenarios [5] where there is no certainty about making decisions. When comparing fuzzy logic with machine learning algorithms, it has been found that constructing the fuzzy rules for a system does not take much effort and time, compared to machine learning algorithms. Machine learning algorithms require large data sets for training to obtain accurate results. In addition, the training time with a large data set is a very time consuming process [6]. However, fuzzy logic may not be suitable in scenarios where it is difficult to deduce the reasoning logic.

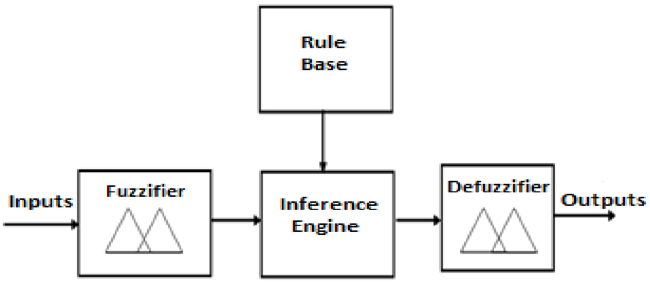


Figure 1. Fuzzy logic Components

It will be shown in section 4 that the logic of detecting multi-stage attacks can be simply modelled using 'if then' rules. Based on the nature of the problem and mentioned advantages of fuzzy logic, it will be a suitable choice for building the reasoning module in the proposed solution.

B. *Social Engineering*

Social engineering can play a role in constructing a multi-stage attack. It is the art of abusing human behaviour in order to violate security without victims realising that they have been manipulated [7].

C. *CRLF Injection*

The CRLF injection, which is also known as HTTP Response splitting, is an attack that can be easily constructed. However, it is an extremely destructive web attack. Attackers construct this kind of attack by exploiting vulnerable web applications that may also allow other types of vulnerabilities, such as cross site scripting and cross site forgery. The CRLF injection is carried out by injecting a very significant sequence of characters into web requests. This sequence contains two special characters representing EOL (end of line), which is used as a marker for many protocols, including such as HTTP and NNTP. In web applications, headers are split based on the position of CRLF in requests. Malicious users inject their own CRLF sequence into an HTTP request. In the absence of filtering malicious inputs, malicious users will be able to control the functionalities of a web application function. In the next section, an example of CRLF injections will be discussed, showing how CRLF injections can be employed by attackers to construct multi-stage attacks [8].

III. Multi-Stage Attack Scenarios

A. *Scenario A*

This scenario has been analysed by using a trace file that contains a capture of real network traffic [9]. The scenario gives an example of how attackers can register machines to their bot army. Fig. 2 gives an overview of the sequence of the attack. The figure shows that the attacker used the compromised host to contact a bad DNS server. The DNS server returned an unusual DNS response containing 11 IP addresses, while a normal response normally does not return more than five IP addresses. The attacker used the compromised host to scan IP addresses returned in the DNS query response and tried to establish communication with them. After a successful 3-handshake with one of the IP addresses returned in the response, the attacker sent packets that contain commands used by the botnet.

Some steps in this scenario could be considered to predict the occurrence of the attack. Detecting a DNS query with a bad DNS server can trigger an alert of malicious traffic. In addition, an irregular DNS response can indicate unusual behaviour. Moreover, sending packets containing commands used by botnet gives a strong indication that the traffic is malicious.

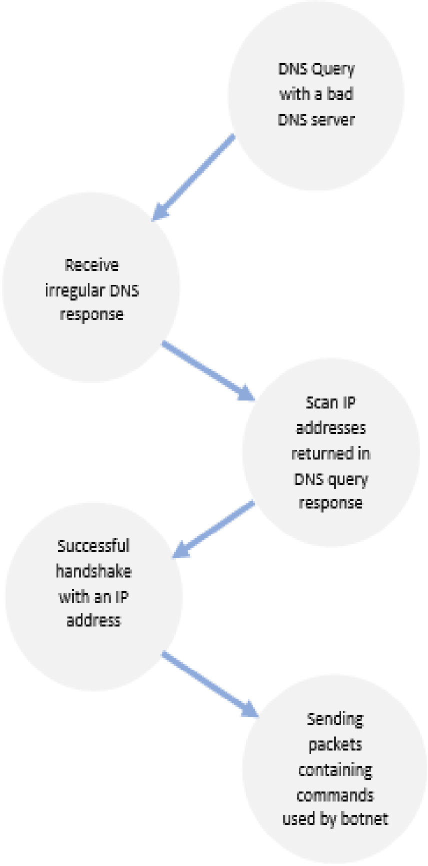

DNS Query with a bad DNS server
Receive irregular DNS response
Scan IP addresses returned in DNS query response
Successful handshake with an IP address
Sending packets containing commands used by botnet

Figure 2. The Stages of Scenario A

B. *Scenario B*

One of the multi-stage attacks, that is social engineering-based, is Operation Shady Rat. This attack was categorised by MacAfee [10] as an advanced persistent threat. An Operation Shady Rat attack involves five steps. In the first step, attackers select one or more organisations, then email individuals who work at those organisations. The emails sent contain information that attracts those individuals. Those emails also contain attached files that are relevant to the email body. Those files appear to recipients as normal files such as Word, Excel, or pdf files, but they are loaded with malicious code. For example, employees in a marketing company have a high interest in getting new contacts. Therefore, attackers may target this group by sending an email attached with an Excel file containing a contacts list. In the second stage, recipients download the attached files, then open them. At the point of opening the file, the malware is installed on the victim’s computer, thus compromising their computer. In the third stage, the installed malicious program tries to establish a connection with a remote site specified in the code. The remote site URL does not look suspicious and it looks like a link to an image or normal html file, but the returned contents from that URL contains some information used by the malicious code. That information cannot be seen as being suspicious content, as it appears as a part of the html content. In addition, that information may be encoded or encrypted, so it will be difficult to analyse. For example, html comments can be used to embed the information that malware uses inside the html content. The comments are visible to end users, look absolutely legitimate, and cannot be seen as any kind of threat. The html comments may contain an IP address of a remote server or a command in an encrypted or encoded format. In the fourth stage, the installed malicious code establishes a connection with the IP address obtained in the third stage. In the fifth stage, attackers at the remote site establish a remote shell and run shell commands targeting the compromised machine. Attackers at this point can upload or download from the compromised side. Fig. 3 shows the sequence of this scenario.

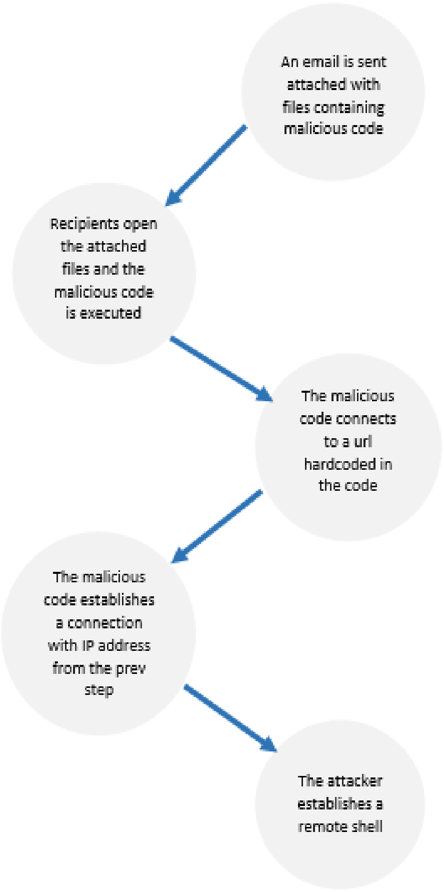
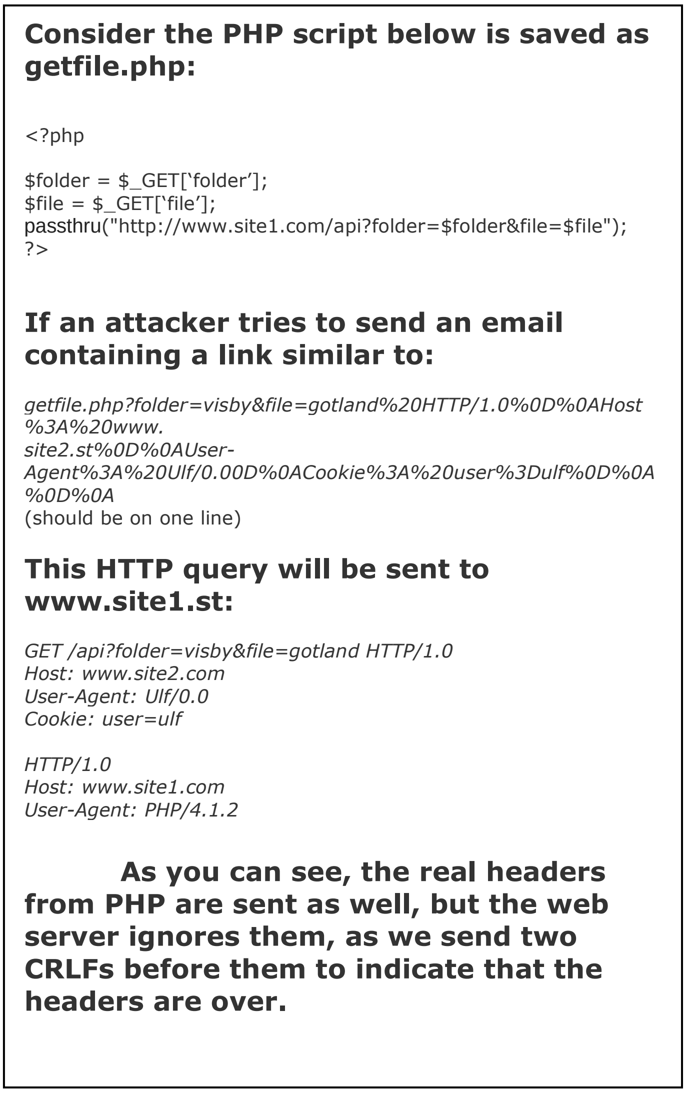

An email is sent attached with files containing malicious code
Recipients open the attached files and the malicious code is executed
The malicious code connects to a url hardcoded in the code
The malicious code establishes a connection with IP address from the prev step
The attacker establishes a remote shell

Figure 3. The Stages of Scenario B

All steps of this scenario look legitimate and not suspicious. However, checking the reputation of the IP addresses involved in the communication traffic between the malware code and other servers may give an indication of suspicious traffic.

C. Scenario C

This scenario is based on exploiting an insecure web application. An insecure web application can give a chance for attackers to get access to machines. The scenario shows how attackers exploit a vulnerable PHP web application to make a CRLF injection. The first step in this attack is carrying out a web vulnerability scan on a web server. This scan gives an attacker information about PHP configurations and different URLs, including POST and GET parameters sent with them. The attacker then uses that information to send an email to a victim containing a CRLF-manipulated link. This link looks legitimate, but it contains parameters set to values that makes a vulnerable web application open a different URL, rather than the specified URL in the code, as shown in Fig. 4 [8].



Consider the PHP script below is saved as getfile.php:

```
<?php

$folder = $_GET[`folder’];
$file = $_GET[`file’];
passthru("http://www.site1.com/api?folder=$folder&file=$file");
?>
```

If an attacker tries to send an email containing a link similar to:

getfile.php?folder=visby&file=gotland%20HTTP/1.0%0D%0AHost%3A%20www.site2.st%0D%0AUser-Agent%3A%20Ulf/0.00D%0ACookie%3A%20user%3Dulf%0D%0A%0D%0A
(should be on one line)

This HTTP query will be sent to www.site1.st:

```
GET /api?folder=visby&file=gotland HTTP/1.0
Host: www.site2.com
User-Agent: Ulf/0.0
Cookie: user=ulf

HTTP/1.0
Host: www.site1.com
User-Agent: PHP/4.1.2
```

As you can see, the real headers from PHP are sent as well, but the web server ignores them, as we send two CRLFs before them to indicate that the headers are over.

Figure 4. CRLF Injection on a PHP script

The injected URL may point to a file that runs on the victim's machine to push a remote shell for the attacker. The attacker proceeds by getting access to the web server, then downloads files or scans the network to find information they are interested in, or find targets they want to destroy. Fig. 5 shows the steps that occur during this attack.

This type of attack can be predicted or stopped at different points. The first point is checking parameters sent with web requests coming to the web server, whether it can cause CLFR injections or not. In addition to that, outgoing requests from the web server can be checked to see whether they go to trusted destinations or not.

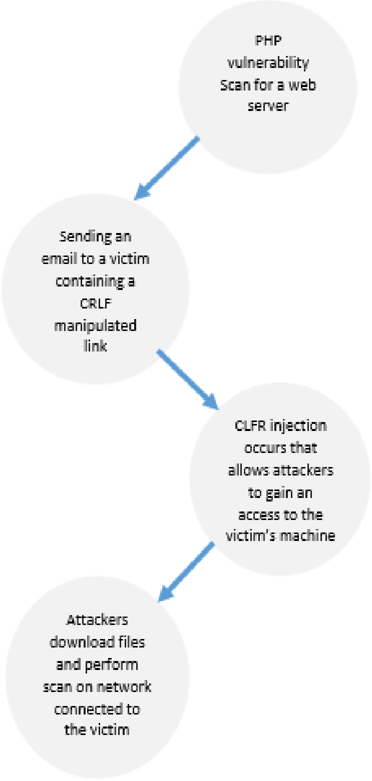

PHP vulnerability Scan for a web server
Sending an email to a victim containing a CRLF manipulated link
CLFR injection occurs that allows attackers to gain an access to the victim's machine
Attackers download files and perform scan on network connected to the victim

Figure 5. The Stages of Scenario C

IV. The Proposed Solution

A. *An overview on the proposed solution*

The proposed solution is based on evaluating the reputation of IP addresses participating in the captured network traffic. The solution consists mainly of three modules as shown in Fig. 6. The first module (Network Sniffer) is responsible for monitoring network traffic by reading incoming and outgoing traffic (It is implemented using tcpdump). This module extracts IP addresses found in network packets. The IP information finder is responsible for finding information related to the IP addresses. The information obtained by the second module includes IP geographic information and other information that shows whether the IP addresses to be checked are malicious. The last one is the reasoning module which is fuzzy logic-based. This module receives IP information from the previous module then analyses the information based on predefined rules to decide whether the checked IP is a potential source of malicious traffic or not.

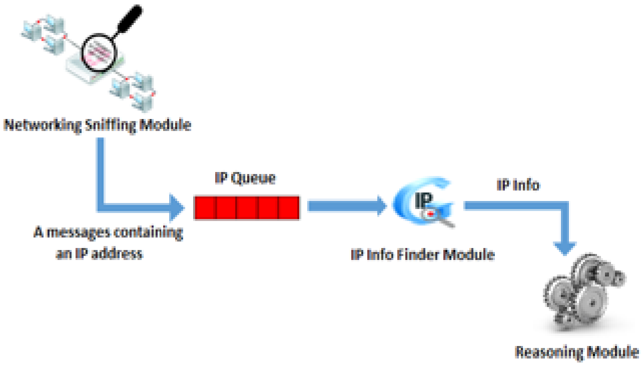

Networking Sniffing Module
IP Queue
A messages containing an IP address
IP
IP Info
IP Info Finder Module
Reasoning Module

Figure 6. The proposed solution block diagram

B. *Network Sniffing Module*

Network traffic is monitored using TcpDump tool. It has been decided to choose this tool as it is has distributions over many operating systems. In addition, it is a command line which simplifies the integration process with other modules. Moreover, it can be used with a software such as wireshark to obtain a graphical representation. TCP dump reads network packets then parsed to extracts IP addresses, it then push messages in a queue that will be consumed by the next module (IP info finder).

C. *IP Information Finder Module*

This module gets information about IP participating in the traffic using web services (Neutrinoapi and fraud lab) [11, 12]. The information obtained includes IP geographic location, whether the IP in a block list, if the IP is an anonymous proxy, if the IP is an exit tor node, and the average IP rating which has a value between one and three (one is a the lowest rate and three is the highest). The geographic location will be checked against a predefined list of countries known with high volume of malicious traffic (the list will be referenced later as the malicious geographic list).

D. *Reasoning Module*

As mentioned earlier, the reasoning module is fuzzy logic based. The reasoning module receives its inputs from the previous module (IP info finder) and analyse them based on defined rules in order to decide whether the IP is malicious or not. The four elements of fuzzy logic have been implemented as follow:

1- *The Fuzzifier:* The membership function selected (is IP in the malicious geographic list, is IP an anonymous proxy, is IP block listed) is a singleton function, as those inputs are Boolean values. The selected membership function for IP rating is specified using triangle functions, as shown in Fig. 7.

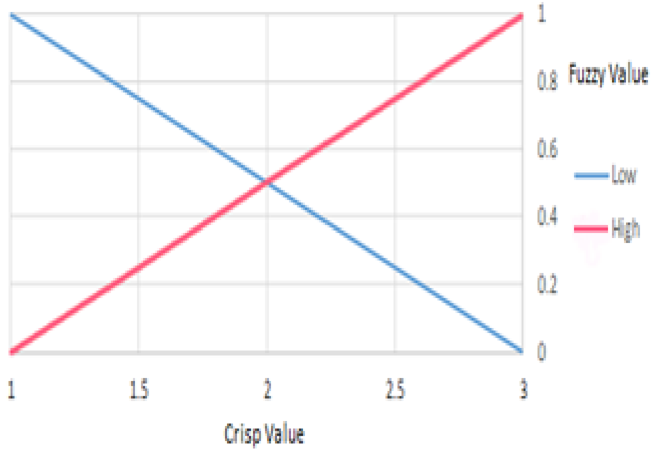


Figure 7: The selected membership function for IP rating

2- *Rule Base:* This is the part that contains the logic of producing the output. The rule base in this module contains four if-then rules as shown in Table1.

Table I If-then rules used in the reasoning module

If condition	Then statement
(IP in a block list)	Possible malicious traffic
(IP country in the malicious geographic list) AND (IP is an anonymous proxy)	Possible malicious traffic
(IP country in the malicious geographic list) AND (IP is a TOR exit node)	Possible malicious traffic
(IP Rating is low)	Possible malicious traffic

The first rule is straightforward, the IP will be considered as a malicious one if the IP address is found in a block list. Finding an IP in a block list means that the IP address has been reported to be used in malicious activities. The second and third rules check two parameters. One of them is whether an IP is on the malicious geographic list or not. It is not practical to consider an IP as a malicious one if it is only located in one of the countries found in the malicious geographic list, as there may be legal traffic from these countries. Anonymous proxies and tor are used in a way that enables users to protect access to the web anonymously. Attackers normally do not need to be in the listed countries, they direct their traffic through a proxy or tor located in one of those countries. Therefore, getting traffic from anonymous proxies or tor-exit nodes located in those countries raises an alert of potential malicious traffic. The last rule checks the average IP rating. The IP address will be considered malicious if the average rating is low.

3- *Defuzzifier:* The selected membership function for the output is as shown in Fig. 8. The output represents the probability of having malicious traffic from the checked IP address. If the probability is higher than 0.5, the IP will be considered as malicious. Otherwise, it will be considered as normal.

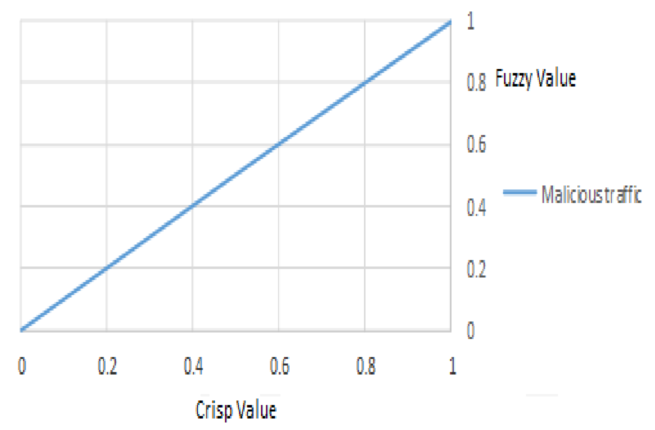


Figure 8: The membership function for the ouput

4- *Inference Engine:* The inference engine can be considered as the heart of reasoning, as it is responsible for mapping given inputs to a fuzzy output, using the specified rules. The inference engine used in this module is Mamdani, which is commonly used in fuzzy logic systems [13].

V. Evaluation

A. *Evaluation Approach*

The system was evaluated using a metrics based approach [14]. The approach looks at intrusion detection systems from different angles, and it includes logistics, architectural, and performance metrics. The logistic metric evaluates the system in terms of maintainability, manageability, and dependency. The design metric is used to find how well the system performs in terms of resources consumption, integration, and speed. The last metric used in this approach is the confusion metric (performance metric), which finds how well the system does its job (detecting multi-stage attacks) in the form of true positive, true negative, false positive, and false negative. Each category in the logistic and design metrics will have a score between one and three (one is the lowest and three is the highest) based on number advantages and disadvantages For example, consider evaluating the system throughput. The system will score one If it has a low throughput while it will score two if it has a high throughput but with consuming a lot of hardware resources. On the other hand, the system will score three if it has a high throughput without consuming a lot of hardware resources.

B. *Logistics Metrics*

Table 2 shows the score for each item in the logistic assessment. The score for the distributed management item is two, as the system supports it but with some potential issues in the buffering area. The score for ease of configuration is two as many of its components can be easily installed but the configurations is not centralized in one user interface and is scattered over different areas. In addition, some components require prior knowledge to get installed. The score for ease of policy management is also two, as detection rules can be easily changed by using the same inputs. The score for outsource solution is poor (one), as the system is found to be massively dependent on using web services. The score for platform requirements is three, as the system supports running on different platforms, and its hardware requirements are dependent on network volume traffic.

Table II Logistics metrics

Item	Score
Distributed Management	2
Ease of configuration	2
Ease of policy management	2
Outsource Solutions	1
Platform Requirements	3

C. *Design Metrics*

Table 3 shows the score for each item in the design assessment. The score for adjustable sensitivity is two, as it supports adjusting sensitivity through modifying the fuzzy rules but is associated with some difficulties in some scenarios. The score of data storage is three, as it does not require less than one Megabyte to store fuzzy rules and blacklisted countries in a database. The score for multi-sensor support is three, as it has the ability to communicate with different sensors, other than the one proposed with the system. The score for both firewall interaction and incident logging/notification is also three, as the system is an open source PHP code that can be easily modified. The score for packet loss is two, as tcpdump cannot perform well in high speed networks. The system throughput on the testing environment has not achieved a high rate abut it is acceptable (around 10 packets/second), so the score will be two for this item.

Table III Design Metrics

Item	Score
Adjustable sensitivity	2
Data Storage	3
Multi sensor support	3
Firewall Interaction	3
Incident Logging and notification	3
Packet loss	2
System Throughput	2

D. *Performance Metrics*

The performance was first tested using a list of 91,744 IP addresses (10.99% Normal, 0.57% anonymous proxy in a black listed country, 88.53% block listed IP addresses) [15] to ensure that the solution was capable of distinguishing between malicious and normal IP addresses. The results obtained were as shown in Table 4.

Table IV Confusion Metrics

Class	True Positive	False Negative
Normal	1.00	0.00
Malicious	0.9984	0.0016

The solution was then tested with four different multi-stage attack scenarios (SQL attack, Cross site scripting, Dictionary attack, and UDP scan) [9] using their trace files. The solution was able to predict three of them (SQL attack, Dictionary attack, and UDP scan) from the first packet, while it failed to detect the cross site scripting scenario as none of the IP addresses participating in the traffic was categorised as malicious.

VI. Related Work

A number of research studies have been conducted in the multi-stage attacks detection area. One of the studies [16] proposes a correlation framework that combines two engines, online and offline, and uses two mechanisms, high quality knowledge-based and statistical-based correlation. The proposed framework achieved a 92% multi-stage detection rate and 21.8% false positive rate during their lab experiments. This approach reduces the computation expenses by analysing only alerts received by IDS. However, the massive dependence on alerts received by IDS may lead to missing capturing attacks if alerts are not received.

Another study [17] proposed a system that follows the attack scenario construction approach. This approach is based on associating two security incidents, and it tries to find consequences of one incident and prerequisites for the incident that may occur later. The strong point of this approach is the ability to construct new attacks created by a mixture of known attacks that can be detected. On the other hand, attacks cannot be tracked without finding cause and effect of these attacks. Moreover, it requires a large consumption of computer resources.

Another study was based on using Hidden Markov Models (HMM) [18]. This study found that the HMM approach achieved greater classification accuracy, compared to other approaches. However, they reported that the accuracy obtained was at the expense of additional computations.

The proposed solution has an advantage over the above-mentioned solutions by not being dependent on receiving alerts from IDS. Nor does it require a complex computation, or memory resources, compared to them. In addition, the previous solutions require an update with sequences of new attacks, while the proposed solution focuses on the identity. However, this may represent an issue, if an attack comes from an IP address not classified yet as suspicious. Moreover, the throughput of the proposed system is relatively low compared to other solutions, due to using web services that take some time to get IP information.

VII. Conclusion and Future Work

The proposed approach in this paper to detecting multi-stage attacks is based on evaluating IP addresses participating in monitored network traffic using fuzzy logic. The solution that used this approach was evaluated using a metrics-based approach. It has a medium score from the logistics perspective. On the other hand, it has a high score when looking from the design perspective. The last part of the evaluation looks at the system performance, and it was found that the system achieved a good performance with zero false positive and a high detection rate. However, the solution fails to detect multi-stage attacks if IP addresses participating in the traffic are not classified as malicious IP addresses.

It is planned to combine the proposed approach in this research with an event correlation-based approach to gain advantage of both approaches by looking at both identities and traffic content.

VIII. References

[1] *D. Clark, "The Problem isn't Attribution; It's Multi-Stage Attacks" the Re-Architecting the Internet Workshop, 2010, Article No.11*

[2] *J. Muila, A Novel Intrusion Detection System (IDS) Architecture, 2010*

[3] *Tal Global. (2011) Operation Shady Rat – What It Really Means, and What You Can Learn From It? [Online] Available from: http://talglobal.com/operation-shady-rat-what-it-really-means-and-what-you-can-learn-from-it/ . [Accessed: 19th Feb 2015]*

[4] *S. Rajasekaran, G.A. Pai, "Neural Networks, Fuzzy |Logic and Genetic Algorithm: Synthesis and Applications" 2003, PHI Learning Pvt. Ltd.*

[5] *P. Alberto, A. Sala, and M. Olivares, "Fuzzy Logic Controllers. Methodology. Advantages and Drawbacks" [Online] Available from: http://www.softcomputing.es/ [Accessed: 19th April 2015]*

[6] *K. Pulo, "Fuzzy Logic vs Machine Learning" , [Online] Available from: http://www.kev.pulo.com.au/ai/fuzzyml_report/ [Accessed: 20th April 2015]*

[7] *A. Chitrey, "prehensive Study of Social Engineering Based Attacks in India to Develop a Conceptual Model" IJINS ,2012. vol.1, n.4*

[8] *B. Hall, 2011 Countering Web Injection Attacks: A Proof of Concept, MSc thesis, University of Manchester UK.*

[9] *TP Group, Network Trace Files [Online] Available from: http://www.tp.org/jay/nwanalysis/traces/General%20Trace%20Files/ [Accessed: 24th Feb 2015]*

[10] *Mcafee. Operation Shady Rat – What It Really Means, and What You Can Learn From It? [Online] Available from: http://www.symantec.com/ [Accessed: 25th Feb 2015]*

[11] *Neutrino (2015) Web Services [IP Check Web Services] Available from: https://www.neutrinoapi.com/. [Accessed: 25th Feb 2015]*

[12] *Fraudd Lab (2015) Web Services [IP Check Web Services] Available from: https://www.fraudlabs.com . [Accessed: 25th Feb 2015]*

[13] *Mathworks. "What Is Mamdani-Type Fuzzy Inference?" [Online]*

Available from: http://uk.mathworks.com/help/fuzzy [Accessed: 26th April 2015]

[14] *G. A. Fink, "A Metrics-Based Approach to Intrusion Detection System Evaluation for Distributed Real-Time Systems" Information TransferTechnology Group, 2002, Code B35, Naval Surface Warfare Center, Dahlgren Division*

[15] *Sans (2015) Block Listted IP Addresses [List of IP addresses] Available from: https://isc.sans.edu//block.txt. [Accessed: 26th April 2015]*

[16] *F. Alserhani, M. Akhlaq, I. Awan, A. Cullen, and A. Mellor (2009): "Multi-Tier Evaluation of Network Intrusion Detection Systems" Journal for Information Assurance and Security (JIAS), 5 (4): 301-310.*

[17] *S. Templeton and K. Levit. A requires/provides model for computer attacks. In Proc. of New Security Paradigms Workshop, pages 31 – 38. September 2000.*

[18] *D. Ourston, S. Matzner,W. Stump, and B. Hopkins. Applications of hidden markov models to detecting multistage network attacks. In Proceedings of the 36th Hawaii International Conference on Systems Sciences, Los Alamitos, CA, USA, 2003 2003. IEEE Comput. Soc. 36th Hawaii International Conference on Systems Sciences, 6-9 January 2003, Big Island, HI, USA.*