LIST OF PUBLICATIONS.


Role of the Penetration Testing and By Pass Testing in Web Applications

Naga Srinivasu, S, V¹* and Dr. I. Ramesh Babu²

1. Research Scholar, Dept. of CSE, Acharya Nagarjuna University, Guntur.
2. Professor, Head, Dept. of CSE, Acharya Nagarjuna University, Guntur, AP

KEYWORDS
Software Testing; Bypass Testing; Penetration Testing; Web Applications

Abstract: Role of the penetration testing and bypass testing in web applications has presented in this paper. Testing is the process of exercising a Web Application with the intent of finding errors. This paper clearly explains the procedure of the software testing process, bypass testing and its classification as Value Level Bypass Testing, Parameter Level Bypass Testing and Control Flow Level Bypass Testing. It also explained the role and classification of the Penetration Testing methodology as External Penetration Testing, Internal Penetration Testing and Social Engineering Testing.

1. INTRODUCTION

The World Wide Web has a great impact on how software is being developed and deployed. Web applications introduced new priorities for developers, driving the industry to value reliability, usability and security. The software quality poses a new need in developing new methods to design, implement and test software that is characterized by a distributed environment, implemented on a diverse collection of hardware and software platforms and often requires interaction of heterogeneous components. User interfaces are dynamically created and the flow of control depends on the inputs provided to the system.

Web applications are used to transmit, accept and store data that is personal, company confidential and sensitive. Web software components are distributed across multiple computers and organizations, are often created and integrated dynamically, are written in diverse languages and run on diverse hardware platforms and must satisfy very high requirements for reliability, availability and usability. Analyzing, evaluating, maintaining and testing these applications present many new challenges for software developers and researchers. A common activity of Web applications is to validate the users’ data. This is necessary to ensure that the software receives data that will not cause the software to do bad things such as crash, corrupt the program state, corrupt the data store on the server, or allow access to unauthorized users. This type of input validation and testing is crucial for Web applications.

Testing is the process of exercising a Web Application with the intent of finding errors. Tests must be designed to uncover errors in Web Applications that are implemented in different operating systems, browsers, hardware platforms, communications protocols, backroom applications. The testing framework for Web applications is shown in Figure 1.

* NAGA SRINIVASU, S, V
Research Scholar,
Dept. of Computer Science & Engineering
Acharya Nagarjuna University,
Guntur, AP, INDIA
Ph. No: 91-9849866902
E-mail: dr.svnssrinivasu@gmail.com
sarma_6@yahoo.com

Figure 1 Testing Framework for Web applications

Testing a web-based application includes three pivotal quality attributes performance, reliability and robustness. A
common technique in Web applications is to perform input validation on the client with scripting languages such as JavaScript. A dangerous problem with client-side input validation is that end users can bypass this validation. Bypassing validation can reveal faults in the software and can also break the security on Web applications, leading to unauthorized access to data, system failures, invalid purchases and entry of bogus data. To do this client side input validation testing, a strategy called bypass testing has been developed.

Bypass testing is a testing technique for Web applications that skips client-side validation mechanism and automatically generates input data to verify the reliability, security, performance and other quality aspects of Web applications. Bypass testing has several advantages; it can be easily automated, it does not require access to the source code of the Web application, and it can reveal faults that can be easily overlooked by many other techniques.

A penetration test is a method of evaluating the security of a computer system or network by simulating an attack from outsiders and insiders. Effective penetration tests will couple this information with an accurate assessment of the potential impacts to the organization and outline a range of technical and procedural counter measures to reduce risks.

2. RELATED WORKS


3. SOFTWARE TESTING PROCESS

Software testing can be stated as the process of validating and verifying that a software program/application/product that
meets the requirements that guided its design and development. The testing process is shown in figure 2. The figure reveals that the testing examines one or more of the following quality dimensions:

- Content is evaluated at both a syntactic and semantic level.
- Function is tested to uncover errors that indicate lack of conformance to stakeholder requirements. Each WebApp function is assessed for correctness, instability, and general conformance to appropriate implementation standards (e.g., Java or XML language standards).
- Structure is assessed to ensure that it properly delivers WebApp content and function, is extensible, and can be supported as new content or functionality is added.
- Usability is tested to ensure that each category of user is supported by the interface and can learn and apply all required navigation syntax and semantics.
- Navigability is tested to ensure that all navigation syntax and semantics are exercised to uncover any navigation errors (e.g., dead links, improper links, erroneous links).
- Performance is tested under a variety of operating conditions, configurations, and loading to ensure that the system is responsive to user interaction and handles extreme loading without unacceptable operational degradation.
- Compatibility is tested by executing the WebApp in a variety of different host configurations on both the client and server sides. The intent is to find errors that are specific to a unique host configuration.
- Interoperability is tested to ensure that the WebApp properly interfaces with other applications and/or databases.
- Security is tested by assessing potential vulnerabilities and attempting to exploit each. Any successful penetration attempt is deemed a security failure.

The figure 2 also reveals that the content model for the WebApp is reviewed to uncover errors. The interface model is reviewed to ensure that all use cases have been accommodated. The design model for the WebApp is reviewed to uncover navigation errors. The user interface is tested to uncover errors in presentation and/or navigation mechanics. Selected functional components are unit tested. Navigation throughout the architecture is tested. The WebApp is implemented in a variety of different environmental configurations and is tested for compatibility with each configuration. Security tests are conducted in an attempt to exploit vulnerabilities in the WebApp or within its environment. Performance tests are conducted. The WebApp is tested by a controlled and monitored population of end users. The results of their interaction with the system are evaluated for content and navigation errors, usability concerns, compatibility concerns, and WebApp reliability and performance.

Figure 2: Software Testing Process

Content testing combines both reviews and the generation of executable test cases. Reviews are applied to uncover semantic errors in content. Executable testing is used to uncover content errors that can be traced to dynamically derived content that is driven by data acquired from one or more databases. Content testing has three important objectives: to uncover syntactic errors (e.g., typos, grammar mistakes) in text-based documents, graphical representations, and other media. to uncover semantic errors (i.e., errors in the accuracy or completeness of information) in any content object presented as navigation occurs. to find errors in the organization or structure of content that is presented to the end user.

4. BYPASS TESTING

Bypass testing was introduced to submit values to Web applications that are not validated by client-side checking. Bypass testing requires a detailed model for how to
introduce inputs to server-side software components. Bypass testing is a unique and novel way to create test cases that is available only because of the unusual mix of client-server, HTML GUI, and JavaScript technologies that are used in Web applications. A major advantage of bypass testing is that it does not require access to the source of the back-end software. This greatly simplifies the generation of tests and automated tools and expects bypass tests can be generated automatically to build tools that parse HTML, discover and analyze the form field elements, parse the client-side checking encoded in the JavaScript and automatically generate bypass tests to evaluate the server-side software. Most input validation focuses on individual parameters, where the patterns of interaction between users and software are fixed and cannot be altered by the users. According to the classification of input validation types bypassing testing will be conducted at three levels, as discussed in the following subsections.

### 4.1 Value Level Bypass Testing

Value level bypass testing tries to verify whether a Web application adequately evaluates invalid inputs. It addresses data type conversion, data value validation and built-in input value restriction. Data type and value modification, HTML inputs are initially strings, but they are often converted to other data types on the server. Data type conversion testing uses values of different types to evaluate the server-side processing, including general strings, integers, real numbers and dates. The HTML tag input has an attribute maxlengh, Pre-defined input restrictions from HTML select, check and radio boxes are violated by modifying the submission to submit values that are not in the pre-defined set.

Special input value, when the data is stored into a database or XML document and under certain kinds of processing. This data is often validated with client-side checking, but sometimes with server side checking, values for text fields are generated with special characters such as commas, directory paths, slashes and ampersands.

### 4.2 Parameter Level Bypass Testing

Parameter Level bypass testing tries to address issues related to built in input parameter selection, built-in data access and inter-value constraints. It is relatively easy to enumerate possible invalid inputs for an input parameter. The restrictive relationships among different parameters are hard to identify, hard to validate and are thus often ignored during testing. There are many kinds of relationships. One type is invalid pair, where two parameters cannot both have values at the same time. For example, it is not reasonable to have a checking account number and a credit card expiration date in the same transaction. Another type is required pair, where if one parameter has a value, the other must also have a value. For example, if we have a credit card number, we must also have an expiration date. Parameter level bypass testing tries to test Web application by executing test cases that violate restrictive relationships among multiple parameters. These relationships are very often difficult to obtain statically and must be identified dynamically. Parameter level bypass testing focuses on relationships among different parameters, therefore, all values of input parameters are selected from a set of valid values.

### 4.3 Control Flow Level Bypass Testing

The previous two types of bypass testing assume users follow the control flow that is defined by the software. This is a safe assumption for traditional software applications. The users of Web applications can alter the control flow by pressing the back button, pressing the refresh button or by directly entering a URL into a browser. This ability adds uncertainty and threatens the reliability of Web applications. Control flow level bypass testing tries to verify Web applications by executing test cases that break the normal execution sequence. The input units that were identified can be used to define all normal control flows from that unit. All the input units that are returned from that submission are considered to be candidates for the next step in the control flow.

The three levels of value level testing; parameter level testing and control flow level testing can be used individually or combined together. Parameter level and control flow level bypass testing focus on interactions among different parameters and different...
server components, thus can be run independently of value level bypass testing.

5. PENETRATION TESTING

The penetration testing offering consists of several major components, with each component having smaller sub-components. This hierarchy allows the client to pick and choose only those services needed at the time, thereby reducing the complexity and cost of the solution. The major components of the offering include the following: External Penetration Testing, Internal Penetration Testing and Social Engineering Testing.

5.1 External Penetration Testing

These are all publicly available network applications such as Email, DNS, FTP, database, Firewalls, Routers, VPN concentrators 802.11(abg) wireless access points Bluetooth devices Dial-In, Specific modems attached to network devices, Blocks of phone numbers (1 to 1000's). Web sites/applications such as SQL Injection, Cross Site Scripting (XSS), Cookie tampering, Incorrect directory permissions. Directory traversal such as Privilege escalation, Missing patches, Authentication credentials, Backend database connections, Operating system components, Middleware Network infrastructure devices.

5.2 Internal Penetration Testing

These are all internal networks, infrastructure devices and applications such as Servers, Desktops, Application servers, Network management devices, Routers, Switches, PBX, VoIP devices Extranet/Intranet networks/servers.

5.3 Social Engineering Testing

Social engineering testing is designed to test the human components of a network. Often the best security technologies in the world can be circumvented by a single employee not following the proper procedures. This testing is designed to test anything from a single employee to a whole department. The testing is carefully designed in cooperation with the client to ensure specific components of existing policies are tested.

These are external phishing emails: attempt to elicit sensitive information, including network login using external email addresses Internal phishing emails; attempt to elicit sensitive information, including network login using spoofed internal email addresses Dumpster diving External calls to help desks, support personnel, etc., attempt to elicit sensitive information, including network login Attempts to physically access computer rooms, wiring closets, etc. These are also pretending to be various support personnel Building walk-through's, sensitive information laying on desks, PC's with no screen saver/passwords, Accounts/passwords written on white boards, monitor, etc., Unlocked cabinets, Other tests as determined by corporate policy and Check of trash can for sensitive information.

6. PENETRATION TESTING METHODOLOGY

When performing external or internal penetration tests, Syrinx Technologies employs a standard 5-step methodology. This methodology allows for a systematic testing process that ensures all appropriate tests have been applied to the proper devices. The testing process is cyclical by nature and often involves discovering and re-testing new networks and devices as they are uncovered during the testing process. Typical Penetration Testing Methodology is shown in figure 3.

![Penetration Testing Methodology Diagram](image)

**Figure 3: Penetration Testing Methodology**

6.1 Reconnaissance

This step attempts to discover as much information about the client as possible using publicly available resources. Various web search engines are used along with information from the client's web site(s). DNS queries also provide useful information along with queries to the various domain registries. Other sources of information include local, state and Federal regulatory agencies.

6.2 Scanning

During this phase various scanning tools are used to determine the operating systems, protocols, ports and applications in use. Depending on the operating systems and
applications discovered, various other port, vulnerability and application scanners are then used to further define the exact environment. The goal at the end of this phase is to understand in detail the exact applications, versions and configurations for all network devices.

6.3 Gaining and Maintaining Access

This phase is where the ethical hacker will attempt to actually gain access to the target systems or network. The exploit could occur over a LAN, the internet, offline or as deception or theft. After the scanning phase where the ethical hacker has established all necessary information about target network, he will try to exploit possible system vulnerabilities to get into the actual network. Additional vulnerabilities could also be created using backdoor Trojans or dialers. The penetration tester might need to use ‘sniffer’ techniques in order to capture data packets from the target network. This is the most important stage of penetration testing in terms of establishing the potential damage to the target systems. During a real security breach it would be this stage where the hacker can utilize simple techniques to cause irreparable damage to the target system.

6.4 Clearing Tracks

The final stage of penetration test or ethical hacking is to check whether the ethical hacker can erase or cover the mark that has been created in earlier stages of the test. Security breaches are made but never detected. This includes cases where firewalls and vigilant log checking were in place. From this stage we can establish what attacks and exploits a hacker is able to cover up and which we can easily detect. In order that the target company’s security engineer or network administrator cannot detect the evidence of attack, the hacker needs to delete logs files and replace system binaries with Trojans. The attacker can use automated scripts and automated tools for hiding attack evidence and also to create backdoors for further attack.

7. CONCLUSIONS

Testing is the process of exercising a Web Application with the intent of finding errors. Testing a web-based application includes three pivotal quality attributes performance, reliability and robustness. Bypass testing is a testing technique for Web applications that skips client-side validation mechanism and automatically generates input data to verify the reliability, security, performance and other quality aspects of Web applications. A penetration test is a method of evaluating the security of a computer system or network by simulating an attack from outsiders and insiders. This paper clearly explains the functioning of the software testing process. Bypass testing and its classification as Value Level Bypass Testing, Parameter Level Bypass Testing and Control Flow Level Bypass Testing. Penetration Testing methodology role and its classification as External Penetration Testing, Internal Penetration Testing and Social Engineering Testing.

8. REFERENCES


7. Anastasis, A., S. and Andreas, S. A.
   “Automatic, evolutionary test data
generation for dynamic software testing”
8. Luciano Baresi “An Introduction to
   Software Testing” Electronic Notes in
   Theoretical Computer Science, Vol. 148,
   Issue 1, February 2006, pp 89–111.
9. Chien-Hung Liu “Data flow analysis and
testing of JSP-based Web applications”
10. Jeff Tian and Li Ma “Web Testing for
    Reliability Improvement” Advances in
11. John, G., Guoliang, D & John, H
    “Deployed software component testing
    using dynamic validation agents”
12. Yao-Wen Huang., Chung-Hung Tsai.,
    Tsung-Po Lin., Shih-Kun Huang., D.T.
    Lee., Sy-Yen Kuo “A testing framework
    for Web application security
    48, Issue 5, August 2005, pp 739–761
13. Atif M Memon “Advances in GUI Testing”
    Advances in Computers, Vol. 58, 2003,
    pp 149–201
14. Morrison, K.W and Purves, R.S
    “Customizable landscape visualizations
    Implementation, application and testing
15. James Miller., Marc Roper., Murray
    Wood and Andrew Brooks “Towards a
    benchmark for the evaluation of
    software testing techniques” Information
    and Software Technology, Vol. 37, Issue
    1, 1995, pp 5–13
PERFORMANCE EVALUATION OF AODV UNDER THE BLACK HOLE ATTACKS USING THE OPNET

Naga Srinivasu S.V 1*, Dr.I.Ramesh Babu 2*
1. Research Scholar, Dept. of CSE, Acharya Nagarjuna University, Guntur
2. Professor, Head, Dept. of CSE, Acharya Nagarjuna University, Guntur

Abstract: Ad-Hoc networks are the self structured, controlled and configured networks that are capable of operating themselves without using any specific communications and infrastructures. Basically, here exist three different types of Ad-hoc networks such as MANET- mobile ad-hoc networks, wireless mesh networks as well as wireless sensor networks. In general, the black hole attacks that take place in the networks are been categorized into two different types such as RREQ based black hole attack and RREP based black hole attack. This paper deals with the performance evaluation of AODV under the black hole attacks by using the OPNET simulator.

1. INTRODUCTION

Ad-hoc On Demand Distance Vector (AODV) Protocol was initially introduced by MANET (Mobile Ad-hoc Networks) working Group in year 2001 for routing. This protocol was one among DV (Distance Vector Routing Protocols) class. In DV, each of the node be familiar with its neighbor node and even know the cost to reach that particular node. Here, the function of node is to uphold its own routing table, store the entire nodes in network and the distance and next hop to this node. In case, whenever the node is not reachable then the distance for that particular node is kept to infinity. All nodes in the network sends the whole routing table periodically to its neighbors so that the node can verify whether there is a good route to another node by making use of its neighbor as next hop. In this process if a link breaks then automatically count-to infinity enables

According to the views of David A. Maltz (2001) nowadays, ad-hoc networks are playing significant role in the networking processes which includes different types of routing protocols that are used for managing the routing process within the mobile Adhoc networks. The Uni-cast as well as multicast are the two different routing processes that take place in Adhoc networks and this particular process is performed by the Adhoc networks with use of the AODV- Adhoc on demand distance vector protocol. This AODV protocol involves a basic algorithm that performs different tasks such as on demand process through which the main routing processes will be managed by using different nodes that are present in the networks.

Parikshit Machwe (2006) stated that in order to add or change the predefined rotes associated with the different types of network parameters such as bandwidth as well as congestion the different rules are passed to the Adhoc networks. In order to connect the nodes within the networks, the AODV protocol will make use of the tree based structure that contains different groups related to multicast routing process. In general, the AODV protocol is a type of reactive protocol which is having similar characteristics of the proactive routing protocols. The proposed analysis is to perform the performance evaluation process of AODV under the black hole attacks by making use of the OPNET

Clifton Lin (2008) opined that routing protocols will make use of the routing tables for the purpose of managing the routing process because they appear in static nature which has fixed routing process through which there are not capable of managing the configuration changes that take place in the network topology. Reactive routing protocols are known as the on demand routing protocols which appear in dynamic nature and are highly capable of managing the entire configuration changes that take place in the network topology. Whereas, hybrid routing protocols are integrated with the proactive as well as reactive routing protocol characteristics and will work on multiple conditions and performs the both tasks done by proactive and reactive protocols.

II.PROPOSED ANALYSIS

The proposed analysis includes the following steps

- To analyze the information on ad-hoc networks in association with routing protocols
- To evaluate the information on different MANET attacks within ad-hoc networks
To design the scenarios of OPNET for evaluating the AODV performance under black hole attacks
To use the OPNET simulation tool for developing the design scenarios

Mobile ad hoc network is used for evaluating the performance of AODV routing protocol under black hole attacks and thus in this context a simple MANET is simulated using OPNET. Wireless LAN mobile workstations are used as the mobile clients and in this scenario a total of 26 mobile nodes are used. They are dragged from the object palette towards the workspace. Application configuration is used to set the applications used across the network and in this scenario FTP and Web applications are used.

**Application Configuration settings**

Application configuration object is used to define the required applications that generate the traffic over the network. Application configuration object has an option to create any number of applications and in this simulation two applications are used like FTP (File Transfer Protocol) and Web. Required configurations are used to set the applications and the corresponding screenshot is as given in figure 1. It can be understood that there are two applications used like FTP where medium load is imposed over the network and for the Web application a simple HTTP is used with heavy browsing. Once the required applications are create now the corresponding profiles should be created to support the application traffic and the corresponding procedure is given in the next section.

**Profile configuration**

There are two applications across the network and to generate the required application traffic corresponding profiles need to be created. Profile configuration object is used to create the profile and as there are two application two profiles are created and the corresponding screenshot is as given in figure 2.

**Mobility configuration**

Mobility configuration is required to set the mobility patterns for all the 26 mobile nodes used across the simulation. There are number of mobility models available for simulation and in this scenario default random way point mobility is used and the corresponding screen is as given in figure 3.
Default random way point is used as the required mobility model for the mobile nodes. Speed of the mobile nodes is set to a constant value of 50 seconds, pause time to 0 seconds, start time is set to a constant value of 15 seconds and the stop time is till end of the simulation. Once the mobility model is set for the mobile nodes across the network now the nodes should support the mobility.

Mobile nodes and wireless LAN server configurations

As discussed in the previous sections there are 26 mobile nodes and a single wireless LAN server and these mobile nodes acts as application clients and WLAN server acts as application server. For communication among these clients and server always a protocol is required and in this simulation AODV is used as the required routing protocol.

Set mobility profile is used to set the mobility defined across the mobile configuration and as the mobility used is default mobility all the mobile nodes now follow the corresponding mobility patterns.

III. RESULTS

Number of hopes per route indicates the overall hopes traversed during the communication process and it plays an important role in evaluating the performance of AODV routing protocol. The actual number of hopes traversed across the route by the three scenarios is given in figure 7.

Figure 4: Mobile nodes and wireless LAN server configurations

Figure 5: Mobile configuration object

Figure 7: Number of hopes per route
It is clear that across the first scenario the number of hopes is constant at the beginning of simulation and later it was decreased against its normal working conditions. When the case with attacks scenario is considered the number of hopes traversed is always increasing and also more when compared to rest of the scenarios.

Route discovery time indicates the overall time taken to discover the route across the communication process. Route discovery time depends on several factors like the overall traffic over the network and the nature of the applications and the actual comparison of the scenarios.

Across the first scenario the route discovery time is constant at the beginning of simulation and later it was decreased against its normal working conditions. When the case with attacks scenario is considered the discovery time is high initially and later on decreased due to increase number of hopes when compared to rest of the scenarios. In the third scenario a standard route discovery time is recorded and this indicates that the performance of AODV has increased even in case of black hole attacks.

**IV. CONCLUSION**

Main aim of this simulation is to evaluate and improve performance of AODV routing protocol under black hole attacks. Mobile ad hoc network (MANET) is simulated using OPNET modeler and three scenarios are created. First scenario has normal MANET working conditions with AODV routing protocol, second scenario has black hole attacks and third scenario has improved AODV working conditions. AODV routing protocol parameters and wireless LAN parameters are configured to simulate the black hole attacks and same parameters are used to improve the AODV performance and overall network and application performance.

**REFERENCES**


[10]R.Lippmann and H.M Heggestad,” Lincoln Laboratory intrusion detection research ”Proc.4th computer mis use and anomaly detection workshop, Maryland.
Mobile ADHOC Network for Performance Assessment of AODV Protocol under Black Hole Attacks

Naga Srinivasu, S.V 1 and Dr.I.Ramesh Babu 2
1. Research Scholar, Dept. of CSE, Acharya Nagarjuna University, Guntur.
2. Professor, Dept. of CSE, Acharya Nagarjuna University, Guntur

Abstract: - Total numbers of active attacks over ad-hoc networks, black hole attacks are happened to be serious attacks that make the networks more vulnerable and it will not be able to perform the routing tasks. In this black hole attacks process, the entire malicious nodes will request for the new routing processes without transferring the data and information present in the networks. In this paper a metric assessment for AODV protocol under black hole attacks in mobile Adhoc networks.

Keyword: AODV, MANET, Black hole, Adhoc networks

I.INTRODUCTION

Ad-hoc On Demand Distance Vector (AODV) Protocol was initially introduced by MANET (Mobile Ad-hoc Networks) working Group in year 2001 for routing. This protocol was one among DV (Distance Vector Routing Protocols) class. In DV, each of the node be familiar with its neighbor node and even know the cost to reach that particular node. Here, the function of node is to uphold its own routing table, store the entire nodes in network and the distance and next hop to this node. In case, whenever the node is not reachable then the distance for that particular node is kept to infinity. All nodes in the network sends the whole routing table periodically to its neighbours so that the node can verify whether there is a good route to another node by making use of its neighbor as next hop. In this process if a link breaks then automatically count-to infinity enables.

Basically, AODV protocol is defined as an on demand routing protocol by means of diminutive delay. Here, the routes are set up only when required in order to diminish the traffic overhead. This protocol further supports Unicast, Broadcast and Multicast exclusive of other protocols. With the help of registration of costs and sequence numbers, the problem occurred with count-to-infinity and loop issues are solved. Each and every hop in AODV contains constant cost of one. To put up the mobile nodes movement the routes age very rapidly and link breakages can be efficiently repaired locally. Moreover to typify AODV, the five norms used by Keshav is AODV distributed, deterministic, state dependent, hop-by-hop and single path.

AODV protocol makes use of IP (Internet Protocol) and refers IP address as unique identifier. It can be easily carried out by setting the subnet mask to 255.255.255.255. However the aggregated networks are supported and these are implemented as subnets. In the complete subnet, barely one router in each has an ability to function AODV and further it serve as default gateway. It needs to uphold a sequence number for complete subnet and also to forward each and every package. Routing table in AODV is extended by sequence number to each of its target and time to live for all its entry. Further even it is prolonged by interface, routing flags, outdated routes the last hop count is stored and for list of precursors.

II.AODV ROUTING PROTOCOLS

a) Uni-cast Routing of AODV Protocol

Uni-cast routing makes use of three control messages, which are:

- RREQ (Route REQuest)
- RREP (Route REPly)
- RERR (Route ERRor)

Whenever a source node wish to send a packet to other node where it doesn't have any route then the source node sends RREQ message. If the node receives RREQ message which it doesn't have seen before it set up reverse route to source node. Here, if end node fails to route its detonation node (that is source node) then it just rebroadcasts the updated RREQ message by incrementing the hop count and in case if end node identify route to source node then it sends RREP message.

RREP consists of unique identifier, destination IP address and sequence number, source IP address and sequence number along with hop count initialized with zero and flags. RREP is uni-casted to source node by taking benefit from its reverse routes. Now when the source node receives RREP message then it verifies whether the hop count in RREP is lesser than one in its own routing table or sequence number of destination is higher than one in its own routing table. If in the above case, none of the condition is true then it just throw the packet. In case if it is true then it just updates its routing table and if that is not its end node then it re-uni-casts the RREP.
Link breakage is one of the common problems in mobile networks. In a network, if a node comprehends those routes to other nodes are not reachable then this node broadcasts RERR message which includes list of unreachable nodes along with their IP addresses and sequence number and some flags. The node that receives RERR message will again iterate over the list of unreachable destination nodes and verifies whether the next hop in its routing table includes one of these nodes or not. If it has the node then it automatically updates its routing table. In case if the receiving node still maintain routes to unreachable nodes it again sends RERR message on its own including this data.

The life time of both routes and links are prolonged by sending the packet through it and by hello messages. Here, a hello message is a special RRER which is valid only for its neighbours. In a network, the node can periodically send a hello message so that no link breaks are assumed whenever they don't get any message from other node for a long time. In an active route if the link breaks then the node can easily repair the route locally. To attain this, the node sends a RREQ message to locate a new route to its detonation on out of order link side by not disturbing other routes. Here, even there exists other special package RREP-ACK for unidirectional or unreliable links. Apart from these, even there are mechanisms like precursors which is used to track the list of active routes for using RERR emission.

b) Multicast Routing of AODV Protocol

One of the main benefits by making use of AODV is that it is an integrated multicast routing. Here, in multicast routing table the sequence number of group and IP address are stored. Apart from these, even it stores leaders IP address and its hop count along with the next hop in multicasting tree and its lifetime. If a node wants to join a multicast group then that particular node should initially send an RREQ message to group address along with join flag set. Any of the nodes in multicast tree that receives RREQ message can give answer through RREP message. In this way a requester can receive numerous RREP from various nodes. Among these entire RREP message the node can select any one of the message having the shortest distance to its group. A Multicast Activation (MACT) message is send to selected tree node in order to activate this branch. In case, if a requester fails to receive any one RREP message from any node then this particular node is referred as a group leader since there doesn't exist any multicast tree for this group in that particular segment. The multicast RREP message comprises hop count to next group member and additional IP of a group leader. Periodically, the group leader sends group RREP (hello message) and further increments sequence number each time in a group. Whenever two network segments are connected then two separate group trees needs to be connected. Each group member that receives two group hello messages from different leaders will identify a tree connection and simultaneously emit RREQ message with repair flag set to group. In case, in a group tree if the node fails to receive any group hello or other group then that particular tree must be repaired with RREQ message and must make sure that not a RREP message from node in its own tree is selected. In a group tree if a group member (which is a leaf) desire to go away from the group then with the help of MACT and flag prune the member an easily prune from branch in case if the member is not a leaf then that particular group member should carry on it process to serve as a tree member.

III ASSESSMENT OF AODV

A detailed evaluation of the simulation procedure followed to create the Mobile ad hoc network and the estimating the performance of AODV routing protocol under Black hole attacks is discussed in this section

a) Number of hopes per route

Number of hopes per route indicates the overall hopes traversed during the communication process and it plays an important role in evaluating the performance of AODV routing protocol. The actual number of hopes traversed across the route by the three scenarios is given below figure , it is clear that across the first scenario the number of hopes is constant at the beginning of simulation and later it was decreased against its normal working conditions.

![Figure 1: Number of hopes per route for AODV](image_url)

b) Route discovery time

Route discovery time indicates the overall time taken to discover the route across the
communication process. Route discovery time depends on several factors like the overall traffic over the network and the nature of the applications and the actual comparison of the scenarios are as given below figure, it is clear that across the first scenario the route discovery time is constant at the beginning of simulation and later it was decreased against its normal working conditions. When the case with attacks scenario is considered the discovery time is high initially and later on decreased due to increase number of hopes when compared to rest of the scenarios.

c) Routing traffic received in bits per sec
Routing traffic received across the network indicates the overall performance of the mobile nodes and the wireless LAN server. The actual performance of AODV is estimated based on the routing traffic received in bits per sec and the actual bit rate for the three scenarios is as given below figure, it is clear that the overall traffic received in bits per sec is more across the first scenario as there are normal working conditions. When the case with second scenario is considered the overall traffic sent is very less due to the Black hole attacks and in the third scenario the overall traffic received has increased and the performance of AODV has improved a lot when compared to second scenario.

d) Routing traffic sent in bits per sec
Routing traffic sent across the network indicates the overall performance of the mobile nodes and the wireless LAN server. The actual performance of AODV is estimated based on the routing traffic sent in bits per sec and the actual bit rate for the three scenarios is as given below figure.

It is clear that the overall traffic sent in bits per sec is more across the first scenario as there are normal working conditions. When the case with second scenario is considered the overall traffic sent is very less due to the Black hole attacks and in the third scenario the overall traffic received has increased and the performance of AODV has improved a lot when compared to second scenario.

e) Total cache replies sent
Total cache replies sent indicates the overall response sent from the wireless LAN server and the actual performance of AODV routing protocol depends on the cache replies sent. Cache replies should be less for the idle working conditions of AODV and the actual comparison graph is as shown below figure.
When the case with second scenario is considered the overall cache replies sent is more due to the Black hole attacks and thus the value has increased a lot. When the case with third scenario is considered the number of cache replies are reduced when compared to the second scenario and thus the performance of the AODV has increased.

Table 1: Parametric evaluation for AODV

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Normal scenario</th>
<th>Attacks scenario</th>
<th>AODV performance scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data dropped bits per sec</td>
<td>221.97</td>
<td>504.48</td>
<td>53.44</td>
</tr>
<tr>
<td>Load</td>
<td>46568.5</td>
<td>49264.2</td>
<td>47907.1</td>
</tr>
<tr>
<td>Network Load</td>
<td>46411.3</td>
<td>49023.4</td>
<td>47959.68</td>
</tr>
<tr>
<td>Retransmission attempts</td>
<td>0.1163</td>
<td>0.1404</td>
<td>0.1408</td>
</tr>
<tr>
<td>Throughput bits per sec</td>
<td>302598.8</td>
<td>288330.2</td>
<td>278858.26</td>
</tr>
</tbody>
</table>

IV CONCLUSION

Wireless communication has gained lots of popularity these days due to the nature of flexibility and usage standards. The main aim of this research is to evaluate the impact of Black hole attacks over AODV routing protocol and thus mobile ad hoc network (MANET) is considered as the required wireless network. OPNET modeler simulation is done to create the MANET and three scenarios are used to evaluate the performance of AODV under Black hole attacks. All the mobile nodes and wireless LAN server are set to support the AODV routing protocol and two applications are created like FTP and Web applications. All the scenarios are compared against the performance metrics and from the overall analysis of the results it is clear that performance of AODV has reduced a lot due to Black hole attacks and can be observed from the graphs of second scenario.

V. REFERENCES