NỀN TẢNG MÔ PHỎNG TẤN CÔNG MẠNG MANET DỰA TRÊN OMNET++5.X

Mai Cường Thọ
Khoa Công nghệ thông tin
Đại học Nha Trang
Nha Trang, Việt Nam
thomc@ntu.edu.vn

***Tóm tắt nội dung*—Bài báo này trình bày kết quả nghiên cứu kiến trúc phần mềm mô phỏng mạng OMNeT++ 5.6, thư viện INET4.x và NETA framework, từ đó thiết kế xây dựng nền tảng ứng dụng (NTU-Attack) để mô phỏng một số hình thực tấn công và chống tấn công mạng MANeT trên giao thức AODV. Nền tảng này được thiết kế để tiếp tục phát triển để để nghiên cứu và thực nghiệm các tấn công, và các giải pháp phát hiện và phòng chống tấn công mạng MANET khác.**

***Index Terms*—MANET, AODV, OMNEeT++, INET, NETA**

I. Đặt vấn đề

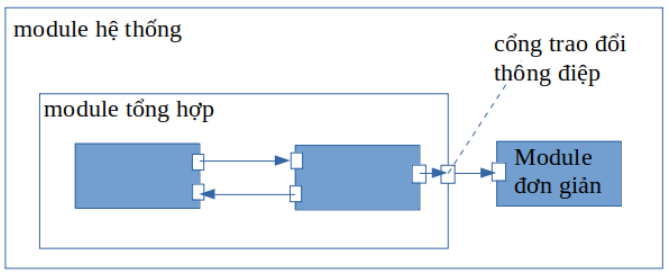
An ninh mạng nói chung, và an ninh mạng không dây nói riêng trong đó có mạng tùy biến không dây di động (MANeT), mạng cảm biến không dây (Wireless Sensor Network), mạng tùy biến di động trong xe hơi (VANeT), mạng cảm biến không dây dưới nước (UnderWater Wireless Sensor Network) luôn là vấn để quan tâm của các nhà nghiên cứu. Việc sử dụng phần mềm mô phỏng trong nghiên cứu là cần thiết trong bối cảnh sử dụng phần cứng thực là rất đắt đỏ và không theo kịp thay đổi của công nghệ. Các công cụ mô phỏng thường được sử dụng để kiểm tra đánh giá các giao thức mạng mới và các hệ thống phức tạp, cung cấp cho cộng đồng nghiên cứu triển khai các nghiên cứu mà không phụ thuộc quá nhiều vào thiết bị vật lý do chi phí lớn. Tuy nhiên việc lựa chọn một trình mô phỏng không phải là nhiệm vụ dễ dàng. Nó đòi hỏi một nghiên cứu trước đó đánh giá những ưu điểm và hạn chế của các phần mềm này, tuy vậy tựu chung, mỗi trình mô phỏng có những ưu, nhược điểm trên các miền ứng dụng riêng.
Nhiều phần mềm mô phỏng đã ra đời [1](NS2, NS3, OMNeT, OPNeT,..), có cả bản mất phí lẫn miễn phí dành cho học thuật. Đối với các nhà nghiên cứu đa số dùng các bản dành cho học thuật, và do đó họ sẽ phải tự nghiên cứu để khai thác phần mềm mà họ sử dụng, điều này tiêu tốn nhiều thời gian. Ngày nay OMNeT++ là một trong các công cụ được sử dụng rộng rãi nhất nhờ vào một lượng lớn các nền tảng (INET, MIXIM,..) nó cung cấp, cùng với đó là tính linh hoạt cao và giao diện người dùng thân thiện và một số ưu điểm khác. Nó là một phần mềm mô phỏng mã nguồn mở, thiết kế hướng đối tượng, mã lệnh C++ hoàn toàn, miễn phí, được phát triển cho cả mạng không dây và có dây, hệ sinh thái lớn dựa vào nó. Do đó rất thích hợp cho nghiên cứu các vấn đề về mạng không dây. Trên cơ sở đó, phát triển một nền tảng dành riêng cho an ninh mạng không dây trên nền tảng OMNeT là rất cần thiết. Việc có được nền tảng giả lập các tấn công mạng là tiền đề cho việc phát triển các nghiên cứu bảo mật mạng MANeT, WSN, VANeT, UWSN.

II. Vật liệu và phương pháp nghiên cứu

Phần sau đây trình bày căn bản các đối tượng nghiên cứu trong bài báo, bao gồm: OMNeT++, INET, và NETA và giao thức định tuyến AODV trong MANET.

A. OMNeT++

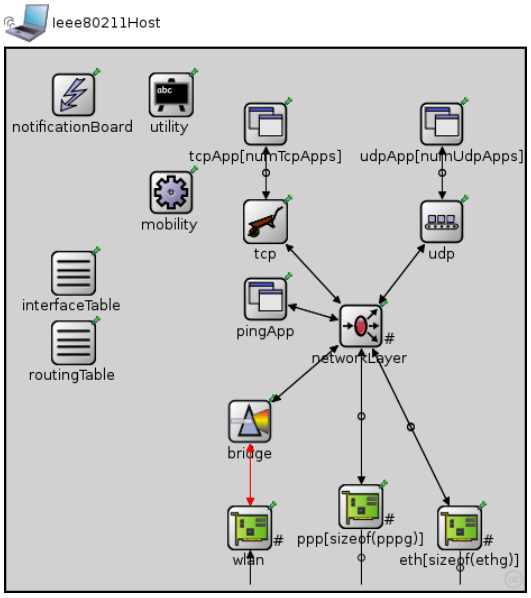
OMNET++ [2] [3] (Objective Modular Network Testbed in C++) là một ứng dụng cung cấp cho người sử dụng môi trường để tiến hành mô phỏng hoạt động của mạng. Mục đích chính của ứng dụng là mô phỏng hoạt động mạng thông tin, tuy nhiên do tính phổ cập và linh hoạt của nó, OMNeT++ còn được sử dụng trong nhiều lĩnh vực khác như mô phỏng các hệ thống thông tin phức tạp, các mạng kiểu hàng đợi (queueing networks) hay các kiến trúc phần cứng... OMNeT++ cung cấp sẵn các thành phần tương ứng với các mô hình thực tế. Các thành phần (module) được lập trình theo ngôn ngữ C++, các module này sau đó được tái sử dụng và tập hợp lại thành những thành phần hay những mô hình lớn hơn bằng một ngôn ngữ bậc cao (NED).



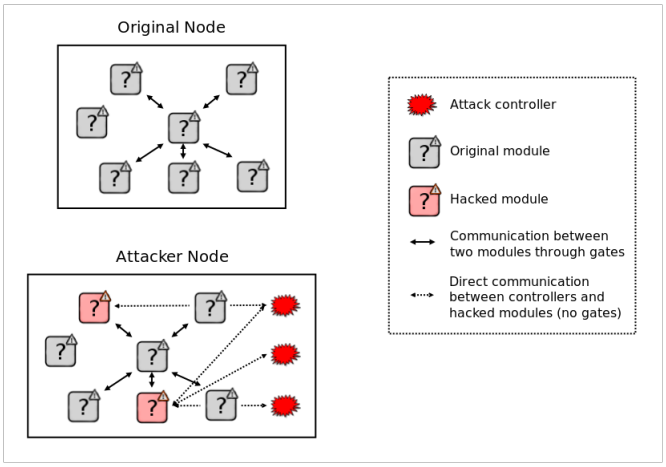
Hình 1. Các thành phần đơn, tổ hợp và giao tiếp giữa các thành phần

OMNeT++ hỗ trợ giao diện đồ hoạ, tương ứng với các mô hình cấu trúc của nó đồng thời phần nhân mô phỏng (simulation kernel) và các module của OMNeT++ cũng rất dễ dàng nhúng vào trong các ứng dụng khác. Các nền tảng mô phỏng chính trên nền tảng OMNeT++ 4.x có thể kể đến như

INET framework, OverSim, Veins, INETMANET, MIXIM và Castalla. Hiện nay, OMNet++ đã phát triển đến phiên bản 5.6 với cấu trúc và tổ chức dự án có nhiều thay đổi, do vậy, nhiều nền tảng trên chạy trên bản 4.x sẽ không tương thích và không hoạt động được nếu không được phát triển và thay đổi theo.



Hình 2. Ví dụ: Node mạng không dây được xây dựng từ các module đơn



Hình 3. Node mạng thông thường và Node mạng tấn công

B. INET framework

INET [8] cung cấp các giao thức, các tác tử, các mô hình khác cho các nhà nghiên cứu và sinh viên làm việc với mạng truyền thông, INET đặc biệt hữu dụng trong thiết kế và kiểm tra các giao thức mới, hoặc khám phá các kịch bản mới và kỳ lạ. Nó là nền tảng được sử dùng nhiều nhất và lớn nhất trong hệ sinh thái OMNeT++ [9][10]. Nó bao gồm các mô hình mô phỏng cho ngăn xếp giao thức Internet (TCP, UDP, IPv4, IPv6, OSPF, BGP), các giao thức lớp liên kết không dây và có dây khác nhau (Ethernet, PPP, IEEE 802.11..), hỗ trợ tính di động, các giao thức mạng MANET, một số mô hình ứng dụng và nhiều giao thức và các thành phần khác. INET được duy trì và phát triển bởi đội ngũ OMNeT++, do vậy sự phát triển, thay đổi của OMNeT++ kéo theo sự thay đổi và phát triển của INET. Phiên bản INET hiện tại tương thích với OMNeT++ 5.x là INET4.x

C. NETA framework

NETA (NETwork Attacks Framework for OMNeT++) [10][11] là một nền tảng mô phỏng tấn công mạng truyền thông, được phát triển trên nền INET và OMNeT++ dựa vào việc xây dựng các node mạng tấn công, hành động tấn công trên các node này sẽ được kích hoạt trong lúc hoạt động bởi các thông điệp điều khiển. NETA được dự định trở thành nền tảng hữu dụng cho các nhà nghiên cứu tập trung chủ yếu trên lĩnh vực an ninh mạng. Thiết kế linh hoạt của nó phù hợp để thực hiện và đánh giá nhiều loại tấn công mạng khác nhau, làm cơ sở đánh giá chuẩn cho các giải pháp phòng thủ hiện tại trong cùng điều kiện thử nghiệm hoặc để phát triển các kỹ thuật phòng thủ mới. Ba loại tấn công được thực hiện trong NETA, bao gồm sinkhole attack, delay attack và IP dropping attack.

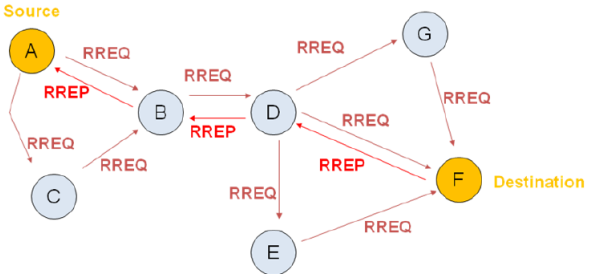
Ý tưởng chính của NETA thực thi các node mạng mới có thể thực hiện tấn công (*attacker node*), việc này được quản trị bởi bộ điều khiển tấn công (*attack controller*), các bộ điều khiển này sẽ quản trị một hoặc nhiều module của một *attacker node* bởi việc gửi các thông điệp điều khiển (*control message*).

Do được thiết kế dựa trên OMNeT++/INET, nên khi kiến trúc nền và tổ chức mã lệnh của nó thay đổi thì NETA không hoạt động được do không tương thích. Đây chính là động lực để chúng tôi phát triển một nền tảng mới dựa trên ý tưởng của NETA và chạy trên INET4.

D. Giao thức định tuyến AODV

AODV là một giao thức định tuyến động, hoạt động theo yêu cầu, đa chặng và tự khởi động giữa các nút di động trong mạng MANET. Nó cho phép tìm đường nhanh và không yêu cầu các nút duy trì các con đường tới đích khi không truyền thông. Giao thức này cũng cho phép các nút hoạt động bình thường ngay cả khi cấu trúc mạng thay đổi hoặc liên kết bị đứt. Quá trình tìm đường được khởi tạo bất cứ khi nào có một nút cần truyền tin với một nút khác trong mạng mà không tìm thấy tuyến đường liên kết tới đích trong bảng định tuyến. Nó phát quảng bá một gói yêu cầu tìm đường (RREQ) đến các nút lân cận. Các nút lân cận này sau đó sẽ chuyển tiếp gói yêu cầu đến nút lân cận khác của chúng. Quá trình cứ tiếp tục như thế cho đến khi có một nút trung gian nào đó xác định được một tuyến "đủ tươi" để đạt đến đích hoặc gói tin tìm đường được tìm đến đích. AODV sử dụng số thứ tự đích để đảm bảo rằng tất cả các tuyến không bị lặp và chứa hầu hết thông tin tuyến hiện tại. Trong quá trình chuyển tiếp RREQ, các nút trung gian ghi vào bảng định tuyến của chúng địa chỉ của các nút lân cận khi nhận được bản sao đầu tiên của gói quảng bá, từ đó thiết lập được một đường dẫn theo thời gian. Nếu các bản sao của cùng một RREQ được nhận sau đó tại một nút, các gói tin này sẽ bị huỷ. Khi RREQ đã đạt đến đích

hay một nút trung gian với tuyến "đủ tươi", nút đích (hoặc nút trung gian) đáp ứng lại yêu cầu RREQ bằng cách phát unicast một gói tin trả lời (RREP) ngược trở về nút lân cận mà từ đó nó nhận được RREQ. Khi RREP được định tuyến ngược theo đường dẫn, các nút trên đường dẫn đó thiết lập các thực thể tuyến chuyển tiếp trong Bảng định tuyến của chỉ nút mà nó nhận được RREP.

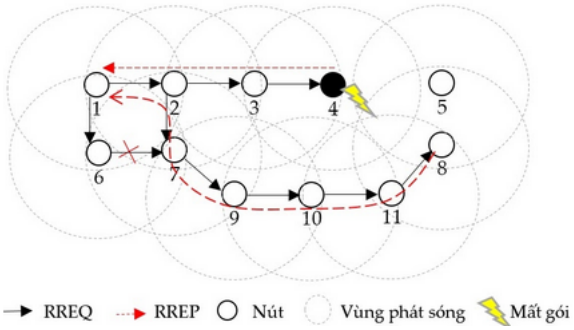


Hình 4. Quá trình khám phá tuyến khi có nhu cầu

Giao thức AODV [13] không hỗ trợ bất kỳ cơ chế an ninh nào để chống lại các cuộc tấn công. Một số điểm yếu chính của giao thức AODV có thể kể đến bao gồm: *(1)- Kẻ tấn công có thể đóng giả một nút nguồn S bằng cách phát quảng bá gói RREQ với địa chỉ IP như là địa chỉ của nút nguồn S. (2)- Kẻ tấn công có thể giả làm nút đích D bằng cách phát quảng bá gói RREP với địa chỉ như là địa chỉ của nút đích D. (3)- Kẻ tấn công có thể giảm giá trị trường hop count trong RREQ và RREP để các nút nguồn cho rằng nó có tuyến đường đi ngắn nhất tới đích. (4)- Kẻ tấn công có thể tăng giá trị trường sequence number trong RREQ và RREP làm các nút nguồn cho rằng nó có tuyến đường đi mới nhất đi tới đích.*

E. Tấn công lỗ đen- Blackhole Attack

Giao thức AODV chưa có cơ chế an ninh trong quá trình khám phá tuyến, nút nguồn NS chấp nhận tất cả các gói RREP nhận được để cập nhật đường đi mới nếu thỏa điều kiện là tuyến đường vừa khám phá đủ "tươi" và có chi phí tốt nhất. Lỗ hổng bảo mật này bị tin tặc khai thác để thực hiện nhiều hình thức tấn công mạng trong đó có tấn công lỗ đen [14]. Tấn công lỗ đen có thể thực hiện với một hoặc nhiều nút độc hại riêng lẻ, trong trường hợp sử dụng hai nút độc hại kết nối với nhau thì hình thức này được gọi là cộng tác tấn công. Để thực hiện tấn công lỗ đen, nút độc hại thực hiện qua hai giai đoạn: Giai đoạn 1, nút độc hại tự quảng cáo cho nút nguồn rằng bản thân nó có tuyến đường đến đích với chi phí tốt nhất, nhờ vậy mà nút độc hại có thể đánh lừa nút nguồn chuyển hướng đến đích thông qua nó. Giai đoạn 2, nút độc hại đón nhận tất cả gói tin từ nguồn chuyển đến và huỷ (drop) tất cả nên đây được gọi là hình thức tấn công phá hoại. Trong cộng tác tấn công lỗ đen thì gói tin dữ liệu được chuyển tiếp đến nút thứ hai, và bị huỷ tại nút này nhằm tránh bị phát hiện. Hình 5 mô tả mô hình mạng có node độc hại (node 4) thực hiện tấn công blackhole, node nguồn (node 1), node đích (node 8)



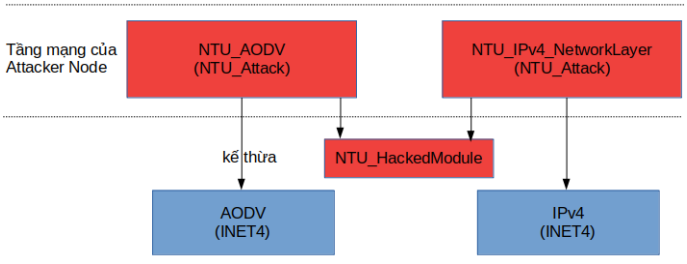
Hình 5. Mạng có node độc hại (node 4) thực hiện tấn công blackhole, node nguồn (node 1), node đích (node 8)

III. Kết quả nghiên cứu và Thảo luận

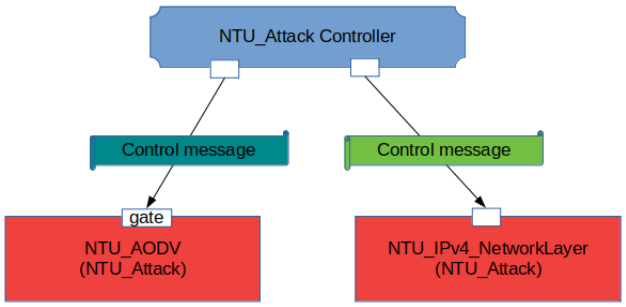
A. Phân tích thiết kế nền tảng NTU-Attack

Hệ thống NTU-Attack được chúng tôi thiết kế dựa trên việc thừa kế và phát triển từ ý tưởng và kiến trúc của NETA framework, tuân thủ nguyên tắc thiết kế của NETA (không hiệu chỉnh các framework nền: INET/OMNeT++; sửa đổi ít nhất có thể các module bị hack), để hoạt động tương thích trên nền tảng OMNeT++ 5.x và INET 4.x, giúp mô phỏng một số tấn công mạng MANET, thử nghiệm các giao thức mới và các kỹ thuật phòng chống và phát hiện tấn công.

Tấn công mạng nói chung, mạng MANET nói riêng có thể được thực hiện ở tầng khác nhau trên mô hình TCP/IP (tấn công tầng ứng dụng, tầng vận chuyển, tầng mạng, và liên kết dữ liệu. NTU-Attack nhằm mục tiêu trước hết là mô phỏng tấn công vào tầng mạng, mà cụ thể là vào giao thức định tuyến AODV. Do vậy, để xây dựng node mạng mạo danh là node đích (gửi ngay gói RREP giả mạo ngay khi nhận được gói RREQ với Destination Number rất lớn) và thực hiện các hành động như hủy gói hoàn toàn, hủy gói theo tần suất nào đó, hủy gói theo giao thức tầng ứng dụng khi nhận được gói, cần xây dựng phiên bản override các module IPv4 và AODV của INET (được đặt tên là NTU-AODV và NTU-IPv4networkLayer) (Hình 6), để thực hiện các hành vi độc hại trên. Các tham số cho các hành vi độc hại được gói trong thông điệp điều khiển gởi bởi bộ điều khiển *NTUattack Controller* (Hình 7).



Hình 6. Mô hình thiết kế các module độc hại dựa trên việc kế thừa, và overide các hàm chức năng của lớp cha, cùng với kế thừa NTU hacked module và ghi đè thực hiện xử lý thông điệp điều khiển nhận được từ controller



Hình 7. Các module độc hại được kích hoạt và nhận thông số qua thông điệp điều khiển gửi bởi module điều khiển tấn công NTU Attack contrller

B. Các module đã xây dựng

Module NTU-AODV - là module độc hại, thực hiện hành vi tấn công lỗ đen, lỗ xám và lỗ chìm. Như thiết kế, các tham số điều khiển và kích hoạt tấn công sẽ được đóng gói trong thông điệp điều khiển cùng tên, được tạo ra và gửi tới từ bộ điều khiển.

Module NTU-IPv4networkLayer - là module độc hại, thực hiện hành vi loại bỏ gói tin với tần suất nào đó.

Module NTU-Attack-Controller - là module điều khiển việc thực hiện các hành vi độc hại, các tham số cho thực hiện hành vi gồm có: thời gian kích hoạt, thời gian ngừng kích hoạt, lỉ lệ hủy gói, giao thức hủy gói. Để thực hiện các kiểu tấn công cụ thể, từng module điều khiển cụ thể sẽ được xây dựng dựa trên việc kế thừa module NTU-Attack-Controller và xây dựng cấu trúc thông điệp điều khiển riêng, tạo thông điệp và gửi cho các module độc hại NTU-AODV và NTU-IPv4 (Các controller cụ thể đã xây dựng bao gồm: sinkhole, blackhole, grayhole, dropping)

Các node độc hại thực hiện các hành vi tương ứng sẽ được xây dựng bằng tạo bản sao của node bình thường rồi thay thế các module con bình thường bằng các module độc hại, đồng thời gắn thêm các bộ điều khiển thực hiện hành vi độc hai. Hình 8 mô tả mã nguồn node mạng sử dụng AODV bình thường. Hình 9 mô tả node độc hại được tạo ra bằng cách thay thế module AODV bằng module độc hại NTU-AODV, thêm 2 bộ điều khiển thực hiện các hành vi điều khiển tấn công lỗ đen *NTU-BlackholeAttack* và điều khiển hành vi hủy gói *NTU-DroppingAttack*

```
module NTU_AodvRouter extends NTU_AdhocHost
{
    submodules:
        aodv: Aodv {
            @display("p=825,226");
        }
    connections:
        aodv.socketOut --> at.in++;
        aodv.socketIn <-- at.out++;
}
```

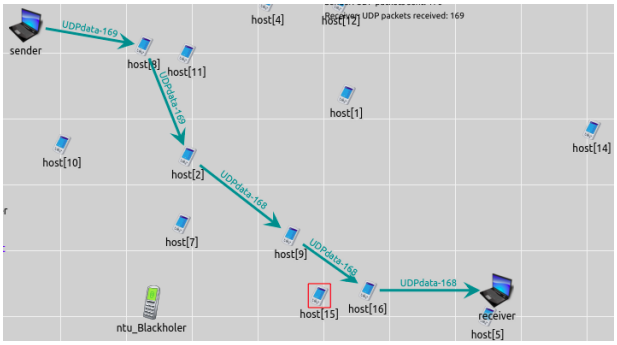
Hình 8. Mã nguồn tạo node mạng sử dụng giao thức AODV bình thường

```
module NTU_Attacker_Blackhole extends NTU_AdhocHost
{
    submodules:
        aodv: NTU_Aodv {
            @display("p=1000.12494,349.65");
        }
        blackholer: NTU_BlackholeAttack {
            @display("p=987.52496,72.45");
            active = true;      startTime = 0s;  endTime = 20s;
            seqnoAdded = 999;  numHops = 1;
        }
        dropper: NTU_DroppingAttack {
            @display("p=837.89996,77.174995");
            active = true;      startTime = 0s;   endTime = 20s;
            droppingAttackProbability = 1;
            droppingUDPdata = true;
            droppingTCPdata = true;
            droppingPINGdata = true;
        }
    connections:
        aodv.socketOut --> at.in++;
        aodv.socketIn <-- at.out++;
}
```

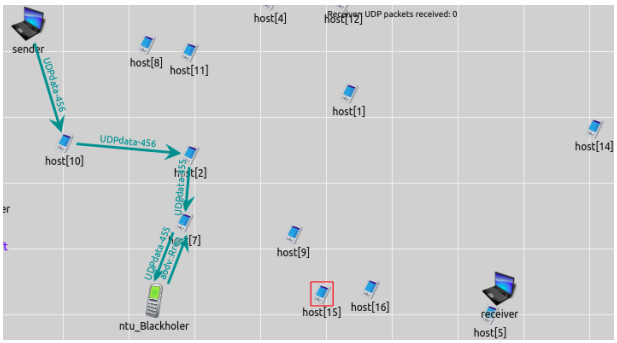
Hình 9. Mã nguồn tạo node mạng thực hiện tấn công lỗ đen giao thức AODV, với tỉ lệ hủy gói bằng 1 trên các loại dữ liệu TCP, UDP, và PING

C. Kết quả thực nghiệm và đánh giá

Dưới đây là 2 hình kết quả mô phỏng trong việc có và không có tấn công blackhole giao thức AODV.



Hình 10. Demo khi node tấn công ntu-Blackholer chưa kích hoạt - mạng hoạt động bình thường



Hình 11. Demo khi node tấn công ntu-Blackholer được kích hoạt - lưu lượng mạng gửi từ sender đến receiver bị ntu-Blackholer bắt và hủy bỏ.

IV. Kết luận

Nghiên cứu cho thấy, có nhiều nền tảng mô phỏng mạng khác nhau, mỗi nền tảng có những ưu điểm riêng. Cùng với sự phát triển của công nghệ mạng, các nền tảng mô phỏng mạng cũng thay đổi theo, dẫn tới có những nền tảng đã không còn phù hợp để phát triển và cũng như là độ tương thích với nền tảng dưới. OMNeT++ và hệ sinh thái to lớn của nó, khiến OMNeT++ là một trong các công cụ ưa thích của các nhà nghiên cứu và sinh viên. INET4, OMNeT++5, và NETA, giao thức AODV, tấn công mạng Blackhole là các vật liệu chính chúng tôi nghiên cứu, từ đó xây dựng nền tảng mới để mô phỏng các tấn công mạng tùy biến không dây di động MANET, cũng như là công cụ để từ đây chúng ta có thể sử dụng đề phát triển các giải pháp phòng chống tấn công mạng MANET và các giao thức mới hữu hiệu. Bài báo đã trình bày kết quả thử nghiệm với tấn công blackhole. Tuy vậy, chúng tôi cũng đã thử nghiệm các tấn công khác như sinkhole, grayhole, dropping và một vài cải tiến giao thức AODV để chống lại tấn công blackhole. Tiếp theo, trong tương lai chúng tôi sẽ cố gắng cài đặt thêm nhiều hình thức tấn công mạng khác trên MANET, cũng như thực hiện các nghiên cứu phòng chống và phát hiện tấn công.

V. Tham khảo

Tài liệu

[1] Manpreet and J. Malhotra, (2014), "A survey on MANET simulation tools," Innovative Applications of Computational Intelligence on Power, Energy and Controls with their impact on Humanity, Ghaziabad, pp. 495-498.

[2] OMNET++,http://www.omnetpp.org

[3] András Varga. "OMNET++ Discrete Event Simulation System Version 3.2 User Manual".

[4] Xian, X., Shi, W., and Huang, H. (2008). Comparison of OMNET++ and other simulator for WSN simulation. 2008 3rd IEEE Conference on Industrial Electronics and Applications, ICIEA 2008. https://doi.org/10.1109/ICIEA.2008.4582757

[5] Lessmann, J., Janacik, P., Lachev, L., Orfanus, D.: Comparative study of wireless network simulators. In: 7th International Conference on Networking. ICN, IEEE Computer Society (April 2008) 517–523

[6] Ur Rehman Khan, A., Bilal, S.M., Othman, M.: A performance comparison of open source network simulators for wireless networks. In: IEEE International Conference on Control System, Computing and Engineering. ICCSCE, IEEE Computer Society (November 2012) 34–38

[7] https://omnetpp.org/download/models-and-tools

[8] https://inet.omnetpp.org/TechnicalPresentations.html

[9] Mészáros L., Varga A., Kirsche M. (2019), "INET Framework. In: Recent Advances in Network Simulation", EAI Springer Innovations in Communication and Computing. Springer, Cham.

[10] https://omnetpp.org/download-items/NETA.html.

[11] Sánchez-Casado L., Rodríguez-Gómez R.A., Magán-Carrión R., Maciá-Fernández G. (2013) NETA: Evaluating the Effects of NETwork Attacks. MANETs as a Case Study. In: Awad A.I., Hassanien A.E., Baba K. (eds) Advances in Security of Information and Communication Networks. SecNet 2013. Communications in Computer and Information Science, vol 381. Springer, Berlin, Heidelberg

[12] Till Steinbach, Hermand Dieumo Kenfack, Franz Korf, and Thomas C. Schmidt, (2011). An extension of the OMNeT++ INET framework for simulating real-time ethernet with high accuracy. In Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques (SIMUTools '11). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, 375–382.

[13] Singh, Meeta and Kumar, Sudeep. (2017). A Survey: Ad-hoc on Demand Distance Vector (AODV) Protocol. International Journal of Computer Applications. 161. 38-44. 10.5120/ijca2017913109.

[14] Swetha, Mr Thunagamani, and M S, Swetha. (2018). A Survey on Different Types of MANET Attacks in OSI Model.