Project Proposal

Project Title: ARP Spoofing Detection Approach for Network Intrusion Detection System

Group member's details:

Serial No.	Name	Enrollment Number
1	Abhishek Kanhar	16114005
2	Anil Kumar	16114012
3	Anoop Singh	16114016
4	Hasanwala Faizal	16114030
5	Rahul Singh	16114052

IP scheme over Ethernet is one of the world's most widely used network structure. However, ARP Spoofing attacks still remain as one of serious security threats on the LAN(Local Area Network). Despite the seriousness, there is no protective mechanism that can effectively protect against ARP Spoofing attacks available yet.

The project proposed plans to build a system that would help customer to detect and act against the intruders trying to get access to the system. In the case of our customer which is a bank having multiple employees on a single network. Network attacks like ARP spoofing becomes common through an infected PC handled by any employee. ARP spoofing can be launched easily through any system which may have been infected by a malware. And for the case of banks which usually have a large network of PCs, it gives attacker a large choice to initiate his attack.

Our project proposes an ARP query process mechanism that corresponds with the current IP/MAC mapping correlations based upon the existing ARP protocol and the "Direct Communication" characteristic of the LAN. It can effectively protect against ARP Spoofing attacks without change of network structures or an increase of investments in personnel and equipments. Our proposed mechanism would work on top of the kernel’s default IP/MAC mapping whilst providing enhanced security and reliability.

Basic intention of an attacker behind ARP spoofing is to either get access to the critical information being transferred between systems or to launch a DoS(Denial of Service) attack against a master server. Both the variety of attacks can result in loss of million of rupees. Even the unavailability of master server for a minute can result in a large loss. Our project proposes to mitigate such losses by reporting such instances to system administration, and also to filter such attack and stop them.

Our proposed project plans to accomplish following tasks

1. Detect any ARP spoofing.
2. Detect any network intrusion in the system.
3. Take actions against the attacker without needing any human interaction.
4. Report the system admins about the attack.
5. Report the handler of PC through which attack was initiated.
6. Provide weekly/monthly analysis of attack on system.

The project is basically about working on networking especially low level networking which itself makes it a not-so-easy task to dive into. (A low level design means that there is not a very solid base to start with) Networking is a very vast topic and though we are just scratching a part of it, but that doesn’t mean one can let the connected topics stay anonymous (One thing leads to another).

Of course, 5 developers will break down things and further we will be taking a slow but solid approach towards the project. Instead of realizing the full final goal at once, we will be working towards building small encapsulated modules at a time, so the goal is surely realizable by our team.

Feasibility Report

User of the software

ARP spoofing detection software which uses active detection, and includes reporting to system admins will be developed for a bank consisting of a very large number of PCs(or systems) connected to LAN(Local Area Network). Bank has hosted number of important websites and softwares on LAN which needs protection from attacks like ARP spoofing.

Main actors interacting with the software:

- System Administrator
    - Oversees the statistics collected from various PCs
- Employee
    - First person to get notified about intrusion(if initiated from his/her PC)
    - To inform system administrator about attack.

Feasibility Study

1) Technical Feasibility

The project delivers two components. First, web based central monitoring system. Second, software which detects ARP spoofing.

Technologies used

1) Python based libraries
2) MySQL
3) Django framework
4) HTML
5) CSS
6) JavaScript
7) Graph plotting tools

Considering above technologies, they are easy to obtain(most of them are free). Skills required to operate this technologies is manageable and easy to learn. So, project is technically feasible.

2) Resource Feasibility

Resources required for the project are

1) Hosting space (which can be availed freely)
2) Database (also available for free)

Cleary project is resource feasible.

3) Economic Feasibility

Project costs are the cost of development and maintenance. Mentioned cost occurs because of assigning software engineers to the project.

Project is financially feasible.

Visibility Plan

Communication between client and development team

It is important to remember that each of your customers is organization with likes and dislikes. But finally customer is a which makes final decision. It is important to report every update/progress to the customer. The contact to the customer will be made in a formal way likes emails. After every week a formal meeting with the customer would be conducted to update customer with the progress or to get feedback from the customer on delivered product.

Communication among team

Communication will go same way as with client, instead it will be more frequent. Regular meeting will discuss the progress accomplished till last meeting. What they plan to accomplish till next meeting. What problems have they faced.

Also coding would be done using methodology of pair programming, which will help to ensure the quality of code, and also help to ensure error prevention in initial coding phase.

Task to be undertaken

Tasks to be accomplished during the the project are guided by the principles of software engineering in a proper manner. After the feasibility study and analysis requirement analysis will be done within the next 2-3 days. Specification of the requirements will be thoroughly mentioned and delivered to the customer for further checking.

The design scheme will consist of modularizing our project into set of smaller project to reduce the complexity of the original problem. The subproblems are easy to handle and can be done by each developers separately. Approximate code length and factors affecting the code will be determined in the designing part. Soon after completion of design the the coding part can be easily initiated with in next few days.

Scalability and durability of the code can be easily maintained by daily meetings of the developers which avoids lengthy and buggy code. After the implementation is members should take a step forward for system testing after being combining the modules developed by individual developers. System testing will consist of dummy attacker system trying to change the look up table and the action of the code will be verified thoroughly. Precaution will be taken at coding step to avoid errors. As the project is rather small if any error found during system testing the errors can be rechecked by the developers within couple of hours without disintegrating the code structure.

After completion of all the above steps the deliverables will be provided to the customer and further correction and adaptive maintenance will be provided as soon as possible.

Functional Objectives

Initial conversation with the customer lead to the conclusion that customer requires following functionality in the software.

- A software to prevent attacks on system.
- Analysis of attacks on system.
- Reporting to system admins.

Performance Objectives

After analysis of functional objectives. Following performance objective were crafted.

1) No requirement for extra hardware.
2) No degradation of PCs speed/performance.
3) Software should run with minimum user interaction.

Environment

Environment required for the software to run on a system.

1) Requires PC's system to run on default IP/TCP stack.
2) One time installation on clients PC.
3) Interaction required with software only in case when ARP spoofing is detected.

Deliverables

A intrusion detection software which runs on PC of every employee. This software detects spoofing and reports incident to a central web based monitoring system which gives statistical analysis to system administration. It also includes a GUI(Graphical User Interface) for system admins. All data will be stored in central database hosted on same system as central web application.

Process to be followed: Incremental Model

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle.

Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.

Since the product is built by decomposing it into number of components each component delivered to the client when it is complete. This allows partial utilization of the product and avoids a long development time. It also avoids a large initial capital outlay and subsequent long waiting period. This model of development also helps ease the traumatic effect of introducing a completely new system all at once.

Outline plan, showing principal activities and milestones

- Meeting within an interval of 2~3 days regarding the project at night
- Requirement analysis and specification ~ 4 days
- Diagram of the project and collecting information for developers to learn.
- Designing and distribution of modules ~ 3 days

- Coding and Unit testing of separate modules ~ 7 days
- Integrating all the modules and system testing ~ 4 days
- Further error analysis ~ 2 days
- Package Formation and transfer ~ 1 day

Risk analysis

1) As the customer is going to be big banks, this may mean that there maybe a sudden increase in requirements, the requirements turn out to be overwhelming or requirement conflicts may happen making it difficult to reach deadlines.
    a) Our basic priority will be to in the very beginning to conduct a requirement analysis taking the worst case scenario
2) Like most of the cases, being a small team, estimation and scheduling can be seriously hampered by the casual approach or illness of even a single person.
    a) To tackle this, the basic approach would be, for every single milestone/ increment decided, there would be a meeting 2-3 days before to collect the status of the developers. If someone is not complete, he would be given the 2-3 days extra(with some penalty) with others working on coalescing their parts together.
3) Procedural risks will be the last aspect to be taken care of. Improper process implementation, conflicting priorities and requirements might be a big step back.
    a) The first way would be not to rush through the designing part to coding part, instead give complete time to designing.