A RESEARCH HONEYPOT TO UNCOVER TOOLS AND TACTICS OF BLACKHAT COMMUNITY

*** M.K. Chandrasekaran , #Dr. Pardeep Goal**

**Singhania University*

#Dept. Of Mathematics, M.M. College, Fatehabad

ABSTRACT

Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done.

INTRODUCTION

Every network security implementation is based on some model, which could he either specified or assumed. Mostly perimeter security models based on Firewalls and/or IDS are in uses which are reactive in nature. This model obviously with above mentioned risks lacks the robustness and provides false sense of security infrastructure. With tremendous complexity and hacking ease looming around; challenge is to build security into the network itself. This will lead to self healing and self defending network infrastructure. To achieve this, security has to he proactive i.e. should he part of the switching fabric that carries all the traffic: begin and malicious. There is compelling need to combine reactive and proactive security measures in order to have an integrated approach to the security across the information value chain.

Keeping this in view, it is proposed to design and develop a proactive network surveillance framework. This Framework aims to provide learning vision to the network attacks. Objective of research work is to bring improved network security through:

- Exploring and analyzing various exploit and their detrimental effects on network security,
- exploring various honeypots and analyze their working,
- Configuring al workplace.
- development of a proactive network surveillance framework,
- creating a bootable enhanced Linux distribution with security scripts and tools (built during this work) to analyze and enhance security,
- deployment and testing of the framework,
- learning and monitoring network in real time.

The scope of the work is to enhance the security at various layers through proposed framework and specifically implement a research honeypot to uncover tools and tactics of black hat community.

HONEYNETS

Honeynets are high-interaction honeypots. No services are emulated, and no caged environments are created. Real systems are offered to the attacker behind some access control device. The system configuration can be heterogeneous i.e. the systems within a Honeynet are true production systems. Honeynets are very flexible tool. Honeynets deceive attackers, detect attacks and capture the unknown.

Honeynets require an extensive amount of time and resources to build, implement and maintain. This technology adds tremendous value as research honeypots. These are used mainly to address following security concerns:

- Who are the attackers?
- What tools they use?
- What tactics do they employ?
- What motivates then

Honeynets can collect in-depth information about the attackers, such as their keystrokes when they compromise the system, their chat sessions with their peers, the tools they used to probe and exploit, vulnerable systems.

As research honeypots, Honeynets also excel at trend analysis and statistical modeling. The information gathered can be used to predict attacks, acting as an early warning system.

WORKING OF HONEYNETS

A Honeynet is constituted as a network of multiple systems. It is a self-contained environment with three critical elements: data control, data capture and data collection. Data control is the controlling of the blackhat activity. Once blackhat takes control of a honeypot within the honeynet, activity needs to be controlled so that attacker can not harm any non honeynet systems. Data capturing is capturing of all the activity that occurs within the honeynet. Data collection is the aggregation of all the data captured by multiple honeynets.

IMPLEMENTATION DETAILS: FILESYSTEM LEVEL

The choice of file system is arm important one since it affects performance, recovery from errors, compatibility with other OS‟s, limitations on partition, file sizes and security features.

A filesystem is the „method" used to organize data on a disk. It. controls the allocation of disk space to files, and associates each file with a filename, directory, permissions and other information. Unlike most other operating systems, Linux supports a wide variety of filesystems.

This is possible because, in its infancy, Linux got a “virtual file system” layer that allowed any filesystem to be „plugged in”.

For the proposed framework, a Journaled Filesystem concept is chosen and necessary changes have been implemented as per the following details.

INITIALIZATION AND SUPERBLOCK SETUP

Whenever a computer is switched off without a proper shutdown there is a possibility that data on the disk becomes corrupted-that is, some of the data will have been written while some has not, leaving files or even internal filesystem data in a “half-finished” state. Whenever that happens the system goes through a routine to check the disk for errors-“fsck” in Linux and “scandisk” in Windows. This is time-consuming, especially on large capacity disk drives.

Journaling filesysterns instead of writing modified files directly onto their area on the disk maintains a “journa1” on the disk which describes all the changes which must be made to disk. Then, a background process takes each journal entry makes the change and marks it as completed. If the system is halted without a shutdown, any pending changes are performed when it is restarted and the system is ready to continue running in seconds. Incomplete entries in the journal are discarded. This guarantees consistency and removes the need for a long and complex file system check on boot up.

REFERENCES

- Ross J. Anderson: Security Engineering: A Guide to Building Dependable Distributed Systems, ISBN 0-471-38922-6
- Morrie Gasser: Building a secure computer system ISBN 0-442-23022-2 1988
- Stephen Haag, Maeve Cummings, Donald McCubbrey, Alain Pinsonneault, Richard Donovan: Management Information Systems for the information age, ISBN 0-07-091120-7
- E. Stewart Lee: Essays about Computer Security Cambridge, 1999
- Peter G. Neumann: Principled Assuredly Trustworthy Composable Architectures 2004

- Paul A. Karger, Roger R. Schell: Thirty Years Later: Lessons from the Multics Security Evaluation, IBM white paper.
- Bruce Schneier: Secrets & Lies: Digital Security in a Networked World, ISBN 0-471-25311-1
- Robert C. Seacord: Secure Coding in C and C++. Addison Wesley, September, 2005. ISBN 0-321-33572-4
- Clifford Stoll: Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage, Pocket Books, ISBN 0-7434-1146-3
- Network Infrastructure Security, Angus Wong and Alan Yeung, Springer, 2009.