Network Security - Fall 2018

Tathagata Mukherjee

Lecture 3 - Term Project

8/28/2018

Make up Classes

- Will have 3 missed classes in September (AFRL Reporting Talks)
- First one Sep 4, 2018 (Make up class: Friday Sep 7, 2018)
- Second one Sep 18, 2018 (Makeup class: Students decide)
- Third one Sep 20, 2018 (Makeup class: Students decide)

Term Project Discussion

- The Linux VM Image will be posted **TODAY** after class
- Instructor will share a DropBox link for download
- Distribution is based on **Ubuntu**
- Distribution has a lot of vulnerabilities
- Goal: To find the vulnerabilities (easy part) and fix them (hard part)
- Hack-a-thon at the end of semester **BEFORE** Thanksgiving

Term Project Discussion

- Root Access: msfadmin\msfadmin
- Use a **Virtual Machine** (Virtualbox or VMware preferred)
- **Do not get owned by unknown actors!** Please!
- **Networking:** Supports *only*:
  1. NAT
  2. Host Only
- Vulnerabilities: Many are documented
- However some vulnerabilities are not documented
- That is the fun part

Term Project Discussion: Attack System

- Use a virtual machine for this as well
- Best options: Kali Linux (Debian) or Backbox Linux
- Many other options: Parrot Security OS etc.
- Many are hard to install on Windows 10 Systems (UEFI)
- Should you use any one of these systems?
- Why or Why Not?
- HIDE, IMPERSONATE, FOOL TARGET

Term Project Discussion: Target System

- Active Security (take active measures)
  - Cover your back (look for basic things)
  - Have policies that **actively** protect system
  - AUDIT AUDIT AUDIT
  - Have a a designated sysadmin (to blame)
  - Port scans: Should you allow them?
  - Should Kali or Backbox distros raise a flag?
  - Protect root
- Make sure that you wear both hats as a team
- Good Luck! ENJOY!

Diffie-Hellman Key Exchange

- Client and server agree on a large prime (shared prime p)
- Both of them agree on an encryption generation method (AES)
- Client comes up prime number $p_1 \neq p$
- Server comes up with prime number $p_2 \neq p$
- p_1 and p_2 are not shared and used as private keys
- Private key (p_1 or p_2) and p used with AES
  - Client: Generate public key p_{k_1}
  - Server: Generate public key p_{k_2}
  - Exchange generated public keys
- Server uses p_{k_1}, p and p_2 to generate encryption key
- Client does the same but with p_{k_2}, p and p_1

What is the Magic

We will study in detail going forward

Diffie-Hellman Key Exchange

- Client and server agree on a large prime (shared prime p)
- Good to ensure $\frac{p-1}{2}$ is also prime
- Client and server *also* agree on another shared key g
- Client comes up prime number $p_1 \neq p$
- Client computes $p_{k_1} = g^{p_1} mod\ p$
- Server comes up with prime number $p_2 \neq p$
- Server computes $p_{k_2} = g^{p_2} mod\ p$
- Exchange p_{k_1} and p_{k_2}
- p_1 and p_2 are not shared and used as private keys
- Private key (p_1 or p_2) and p used with AES
- Client shared key: $p_{k_1}^{p_2} modp = g^{p_1 p_2} (mod\ p)$
- Server shared key: $p_{k_2}^{p_1} modp = g^{p_1 p_2} (mod\ p)$
- Choose g such that $g^{(p-1)} = 1\ (mod\ p)$

Diffie-Hellman Key Exchange

Question

Is it possible to launch a man-in-the-middle attack on the DH Key Exchange