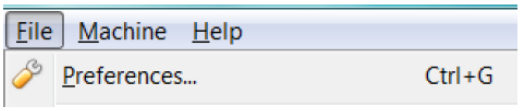
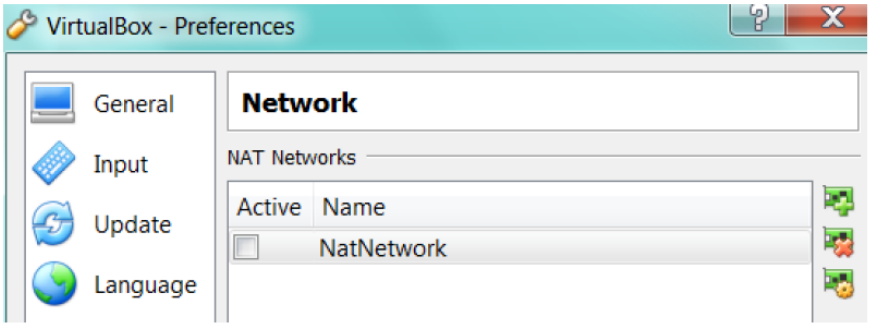
51.506 Security Tools Lab 1

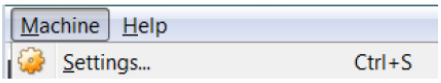
Assignment 3 – Web Application Attacks

Hand-out : 10-Feb-2021

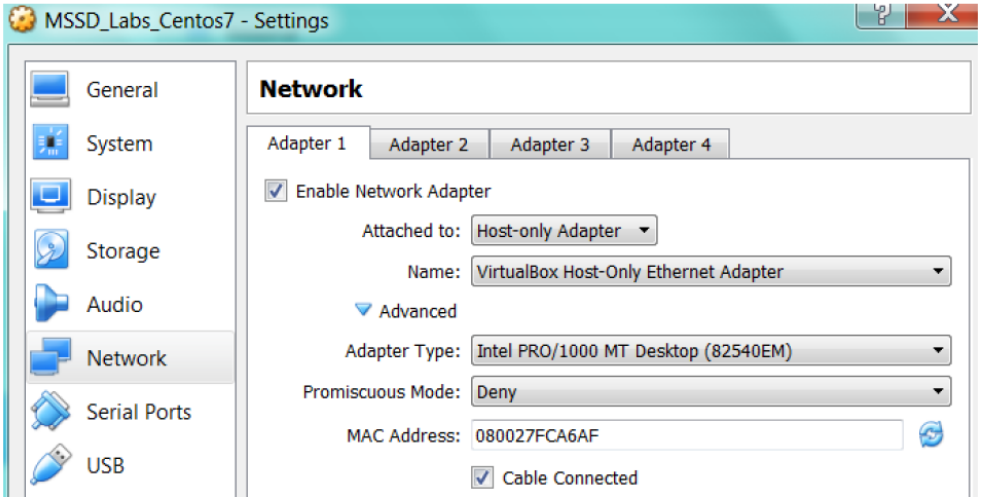
Hand-in : 21-Feb-2021 (2359hrs)

1.	**Objectives** • Perform SQL Injection and Cross-Site Scripting (XSS) on a vulnerable web site • Perform (Shell) Command Injection in order to execute arbitrary commands at server • Experiment with the vulnerability scanning tools

2.	**Setup**
	
	• There are 2 VMs for today :
	  • Kali VM (Needed for Nmap only)
	  • Victim VM (Webserver)
	Import the VMs provided in Virtual Box but don't start it yet.
	
	• Ensure you're **not using** NAT (uncheck if in use).
	 
	
	 
	
	• Ensure you're **using** Host-only adapter



(Adapter type & MAC address might be different)



3. SQL Injection

• The victim machine has HTTP server running at a port that you must find out by nmap tool. You can also discover the IP address of the server using the nmap tool. Recall the commands from today's lecture.

Do not login into the VM.

• To use nmap, ensure you've a Kali/Linux VM or download nmap for Windows.

• After starting the VM in Virtualbox, go to http://<URL_OF_VM>:<PORT> . You will be greeted by a login page. Your role is to login as Alice (who will represent the attacker). Her email is alice@alice.com, but you do not have a password.

• Based on the examples from the presentation, find a way to log in as Alice!

• Consider that the HTTP server is using sqlite3 database, and thus consider its syntax for performing the attack.

• Find at least **another 2 ways** of doing SQL injection (It shouldn't be the one that I demo in class and at least one should use set operator).

4. Cross-site Scripting

• Now that you can login as Alice, we recommend you use **Firefox browser for Alice**, and **Google Chrome or other for Administrator**.

• Restarting the VM will refresh the SQL database and it will be empty.

	Persistent XSS Attack • After compromising Alice's account, as an attacker, you're now targeting Admin as your victim. The objective of this attack is to steal the admin's cookie for malicious purposes. – Your goal is to inject some java script + HTML code on a page that the admin user will display at her/his page. – Your injected code (HTML/.js) should retrieve the admin's session cookie to you. – Therefore, find a way to inject a simple XSS string into the database • Log in as Alice using SQL injection and check the different options the website gives you. Can you use one of them to conduct a persistent XSS attack? Exploit this to obtain the session cookie of the victim. By using that session ID, the attacker can take over an existing session of the victim and impersonate herself/himself. – Explore how posting news works: http://<URL_VM>:<PORT_VM>/news?text=<some_text> – As Alice, store your XSS string into database, and then "wait" until admin's browser call an HTTP GET to an URL of the attacker's choice. – Now, change your role to admin for a while and consider you are benign user. Therefore, use other browser (e.g., Google Chrome) for login in as admin using credentials (admin@a.com / averysecureadminpassword). Then, XSS code injected into database by Alice should be executed either on load of the page or on clicking to some link (depending on how you implement it) • This simulates how admin is attacked, and thus "unintentionally" disclose her/his session ID to the public news list, which is Alice awaiting. – Login as Alice again (using the SQL injection) – You should now be able to see the new news item and admin's session ID.
5.	**Reflected XSS attack** • Look for a way to temporarily inject code via a specifically crafted URL (HINT: .js in URL). • Exploit this to obtain the session of the victim (again) – Let assume the admin is logged into the site when clicking on the link. It means that you have to play role of admin again: login as admin and then enter crafted URL containing XSS string into her/his browser. Note that this malicious URL could admin get by an email, and he/she clicked on it. – Following the link should make the admin visit the webserver with a specific URL, which then causes admin's browser to issue a HTTP GET to http://<IP_VM>:<PORT_VM>/news?text=<adminsession> – Therefore, admin's session should again be leaked as news item – Login as Alice and see the new news item containing admin's session ID (again).
6.	**Command Injection** • This time, you will not inject any JavaScript. You've to find a way to execute a shell command on the server by using one of the provided input fields. • Can you exploit this vulnerability to get the content of a local file, e.g., the file "secret"?

	• A "reverse shell" is a program started on the victim's machine that connects to the attacker's machine. • We need two parts for this: 1. A program waiting on your machine to accept incoming connections – Use netcat: nc -l -p 8080 -vvv on your machine to open a listener waiting on port 8080. In Windows, it is called ncat and is part of nmap package. – When the victim connects, you should be able to type commands, which will be executed on the victim's machine 2. suitable command to run on the webserver (victim VM) that "connects" a shell to the listener running on your machine – Inject that command through bash command injection, as you did in previous section. • Refer to : http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
7.	**Vulnerability Scanning Tools** • Install and use vulnerability scanning tools: *Nikto* (Unix repository) or *uniscan.* • Were these tools successful in discovering of all vulnerabilities you found on the server?
8.	**Hand-in** • SQL Explain the strings you used for the SQL injections and explain why it works. How can this vulnerability be fixed? • XSS Explain which content you inserted in the persistent attack, and how it works. Explain which content you inserted in the reflected attack, and how it works. Describe how it is possible to defend against XSS attacks. • Command Injection Explain which command you used to obtain the content of "secret", and why it works. Explain which command you used to obtain the reverse shell, and why it works. • Vulnerability Scanning Describe what vulnerabilities you were able to discover by these tools and what parameters did you use.