CMSC 426
Principles of Computer Security

Lecture 19
Offensive Security and Hardening

Today's Topics

- Offensive security
  - What it is
  - Attacker Lifecycle
  - Common tools
- Demo
- Effective Windows Hardening

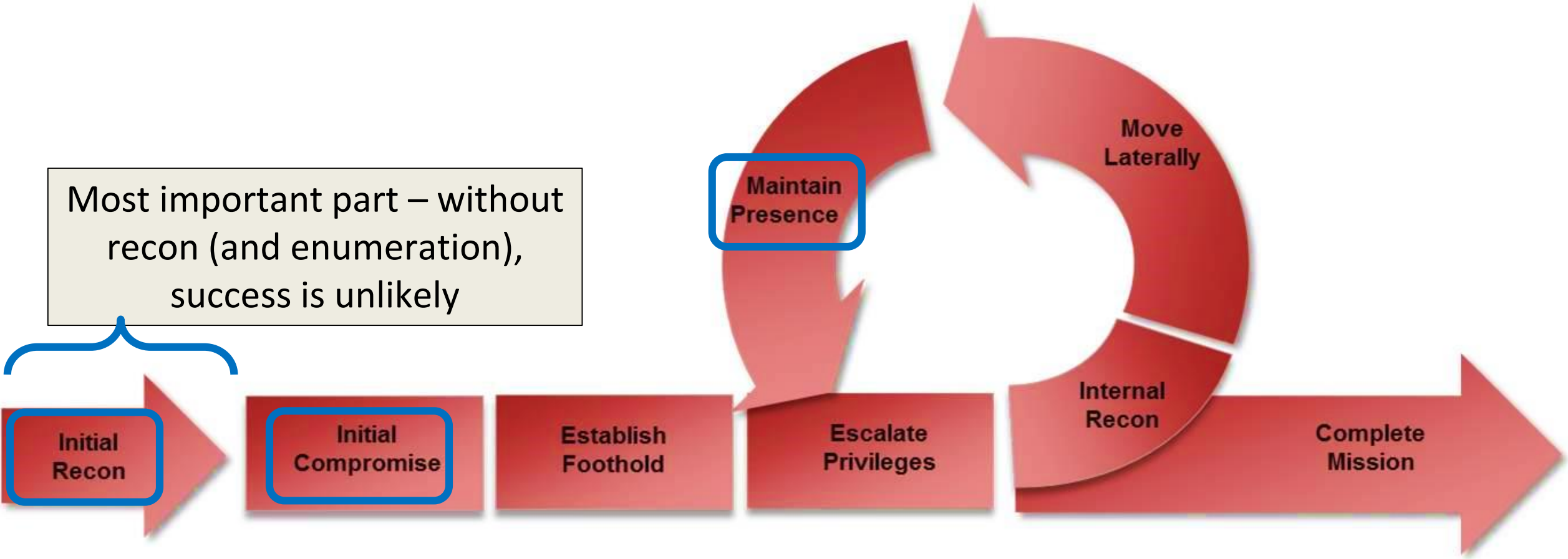
Offensive Security

What is Offensive Security?

- Subset of the security field
  - Focuses on assessing the security of machines or networks by attempting to attack them
- Goal is to be proactive rather than reactive
  - We've previously talked about figuring out who launched an attack, or tracing them back to their source
  - Much easier to prevent it in the first place
- The best defense is a good offense!

Attacker Lifecycle

Most important part – without recon (and enumeration), success is unlikely



1: Reconnaissance

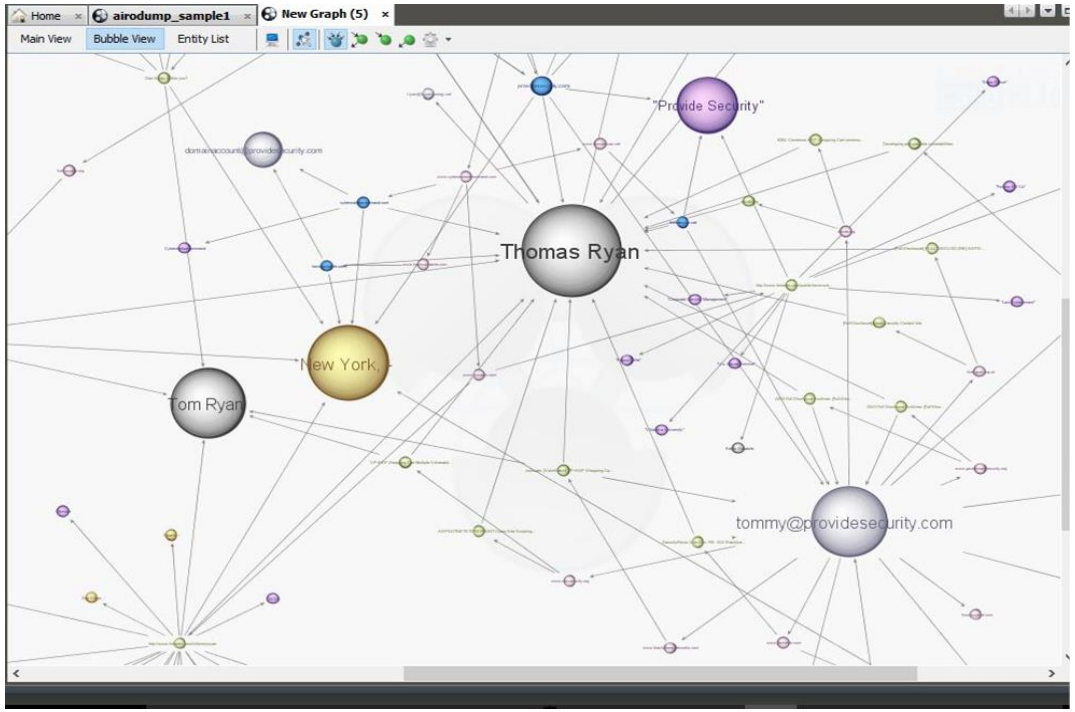
- "Knowing your target"
- Involves gathering information about the target
  - Can often be collected without the target detecting it
- Technical information
  - Network information from scanning, etc.

```
# nmap 192.168.0.245

Starting Nmap 6.00 ( http://nmap.org ) at 2014-02-23 16:26 MST
Nmap scan report for      (192.168.0.245)
Host is up (0.023s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
443/tcp  open  https
2301/tcp open  compaqdiag
5989/tcp open  wbem-https
8899/tcp open  ospf-lite
MAC Address: 00:0C:F1:8B:2D:D1 (Intel)

Nmap done: 1 IP address (1 host up) scanned in 4.76 seconds
#
```

Port scanning

1: Reconnaissance (continued)

- Business information
  - OSINT (Open Source INTelligence)
    - Going to a company's website, github, trello board, etc.
    - Information about employees, positions, technology they use
    - Real examples: passwords in code comments, keys committed to github



- Goal is to get information you can use to get in

1.5: Enumeration

- Goal in this stage is to identify exactly what versions of which services are running

- Look for known exploits and vulnerabilities for those specific versions
- Are there common misconfigurations which show up a lot with these specific technologies?
- How do you test for these misconfigurations?

2: Compromise

- Actually breaking into machines, often what people think about when they think of “hacking”
  - Not actually that exciting though
- It only takes one weak link to own an enterprise
  - Phishing emails, infected document downloads
  - Why client-side exploits are still a thing

k38@umbc.edu , Please E-Sign Form.pdf Now. Inbox ×

DocuSign <support@senzaii.net>
to me

Action Required: Please E-Sign

I am sending you this request for your electronic signature, please review and electronically sign by following the link below.

VIEW DOCUMENT

Thank You,

Shelli Hales

3: Persistence

- After you initially gain access into a network, you want to make sure you can always get back in

- This doesn’t just mean in 5 minutes, it means days, weeks, or months later
- Ideally even after reboots, resets, etc.

4: Post Exploitation

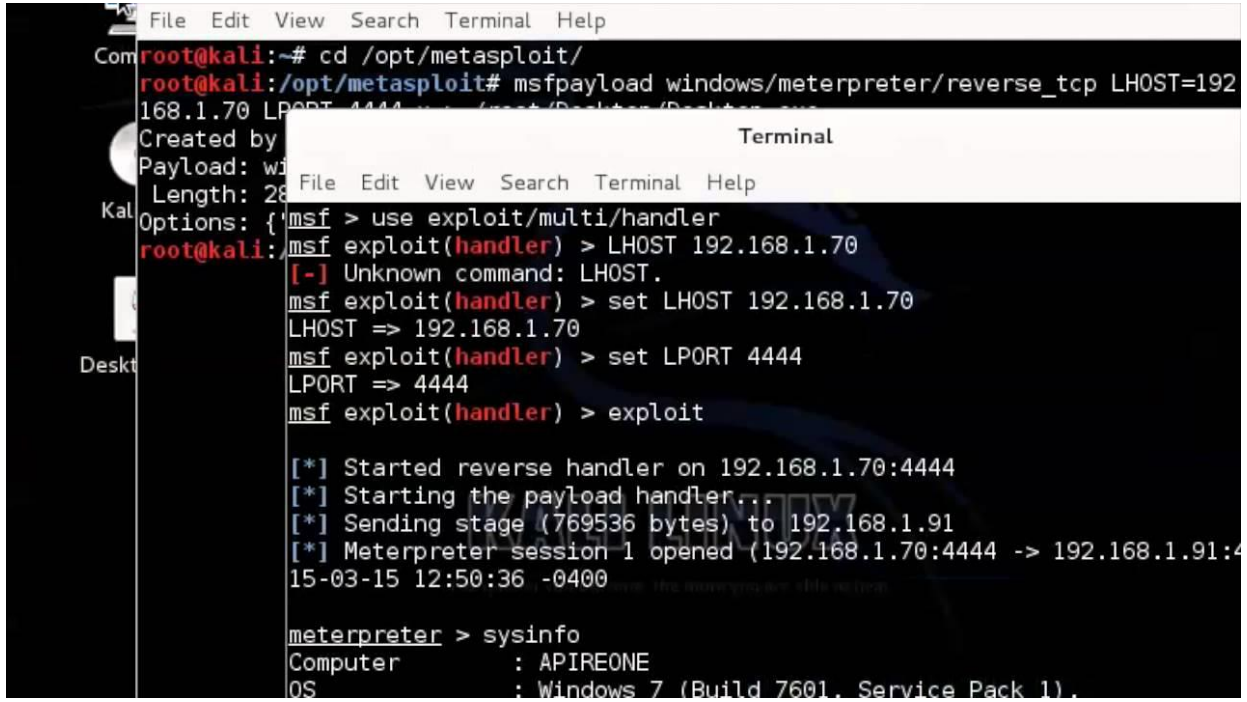
- This is what separates the skilled attackers from everyone else

- What can you do with your access?
  - Can you escalate privileges on your local machine?
  - What is accessible within the network?
  - Can you get access to file servers, internal source code, documents?
  - Can you get access to other users' machines?
  - Can you elevate your privileges on a network level?
    - To Domain Administrator?
  - How easy is it to stay undetected?

Tools and Attacks

Metasploit

- Open source attack framework, written in Ruby
  - Can be used to write and launch own exploits
  - Managing different sessions
  - Escalating privileges
  - Covers basically everything you would need
- Very much a "point-and-click tool"



Using Metasploit

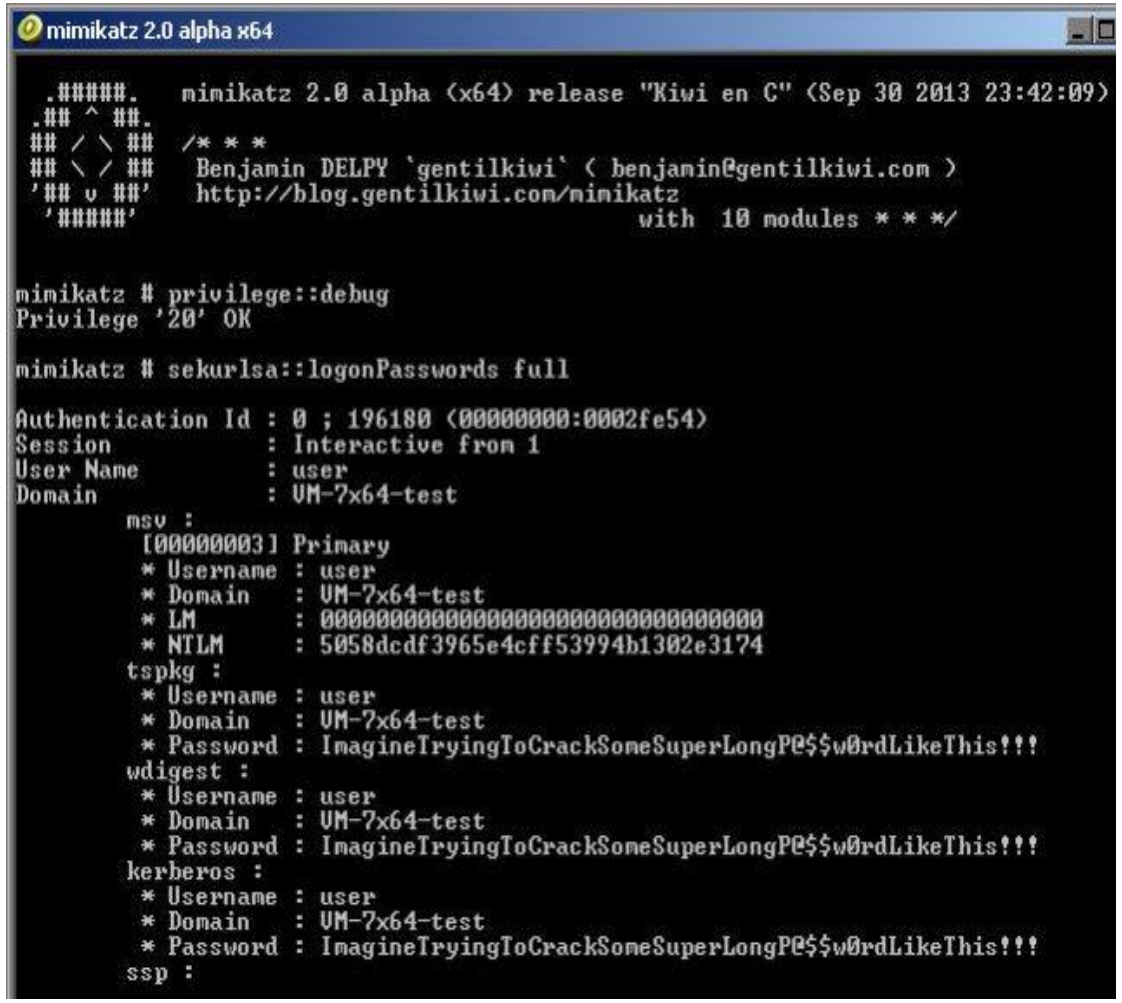
- Walks you through the major steps in launching an attack
  - Choosing and setting up an exploit
  - Checking to see if the target is vulnerable
  - Choosing and configuring a payload
  - Choosing the encoding and evasion techniques for the payload
  - Launching the attack
  - Handling the connections
    - This one is particularly useful

Recap: LSA and LSASS

- LSA (Local Security Authority)
  - Windows subsystem responsible for managing authentication and local security policy
- LSASS (LSA Subsystem Service)
  - The process in which LSA runs
  - Responsible for providing single sign-on functionality in Windows
- Does this by caching credentials in memory
  - Why is this bad?

Mimikatz

- Tool which can dump Windows passwords (and other things) from the memory
  - For ***every user*** who logged in since the last boot
- (This should scare you)



“Passing” Attacks

“Passing” Attacks

- Pass the Hash and Pass the Ticket

- Goal of both attacks is to impersonate another user
  - Pass the Hash: specifically, password hashes
  - Pass the Ticket: Kerberos tickets

- Allows attacker to
  - Gain access to new info and systems
  - Hide their tracks (and prevent discovery)

How a Pass the Hash Attack Works

- Using a stolen password hash in place of the actual password
  - (*e.g.*, obtained through Mimikatz)
- Want to authenticate as a user without having access to their plaintext password(s)
  - Fortunately, NTLM hashes are often just as good as passwords
  - Lots of Windows functions will accept a hash in place of a password
- Newer systems store just the hash in memory
  - Older systems store both the hash and the plaintext (yay?)

Exploiting a Pass the Hash Attack

- Can now be used for lateral movement or spawning processes
  - Connecting to other systems using that same user account/password
  - Launching processes under another account name

- Launching has two benefits:
  - The launching of the application/process cannot be easily traced back to the person who actually launched it, just the username who did so
  - Other users may have more privileges

How a Pass the Ticket Attack Works

- Hacker validates themselves as a specific user by presenting a Kerberos ticket (TGT or SGT) to the system

- Grab another user's cached Kerberos ticket (still valid)
  - Use that ticket as your own, with that user's privileges
  - Kerberos makes use of NTLM hashes on Windows machines

- Why does this work? Doesn't Kerberos verify ticket validity?
  - There's a 20-minute window after creation where this doesn't happen

Three Types of Tickets

- Regular Ticket
  - Intercepted tickets meant for another user
  - May have higher privileges than attacker does
- "Silver" Tickets (Service-Granting Ticket)
  - Forged ticket for specific services
  - Lets the attacker "write their own" ticket <u>for that specific service</u>
- "Golden" Tickets (Ticket-Granting Ticket)
  - Forged ticket for essentially anything
  - Lets the attacker "write their own" ticket for anything

“Fancy” Tickets

- Silver tickets
  - Requires the Service Account password hash
- Golden tickets
  - Requires the KRBTGT (Kerberos TGT) password hash
  - Allows the attacker to sort of “impersonate” the AS
    - Means they can write a ticket for any service on any machine for any user
    - Including users that don’t exist in the system
- Only way to remove a golden ticket’s effectiveness:
  - Change password *twice* (keeps old password for older tickets)

DEMO TIME!!!