Malicious Software

Malware

“A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim’s data, applications, or operating system or otherwise annoying or disrupting the victim.”

Malicious software

- Programs exploiting system vulnerabilities
- Known as malicious software or malware
  - program fragments that need a host program
    - e.g. viruses, logic bombs, and backdoors
  - independent self-contained programs
    - e.g. worms, bots
  - replicating or not
- Sophisticated threat to computer systems

Malware Terminology

- Virus: *attaches itself to a program*
- Worm: *propagates copies of itself to other computers*
- Logic bomb: *"explodes" when a condition occurs*
- Trojan horse: *fakes/contains additional functionality*
- Backdoor (trapdoor): *allows unauthorized access to functionality*
- Mobile code: *moves unchanged to heterogeneous platforms*
- Auto-rooter Kit (virus generator): *malicious code (virus) generators*
- Spammer and flooder programs: *large volume of unwanted "pkts"*
- Keyloggers: *capture keystrokes*
- Rootkit: *sophisticated hacker tools to gain root-level access*
- Zombie: *software on infected computers that launch attack on others (aka bot)*

Some terms

- Payload: actions of the malware
- Crimeware: kits for building malware; include propagation and payload mechanisms
  - Zeus, Sakura, Blackhole, Phoenix
- APT (advanced persistent threats)
  - Advanced: sophisticated
  - Persistent: attack over an extended period of time
  - Threat: selected targets (capable, well-funded attackers)

Viruses

- Piece of software that infects programs
  - modifying them to include a copy of the virus
  - so it executes secretly when host program is run
- Specific to operating system and hardware
  - taking advantage of their details and weaknesses
- A typical virus goes through phases of:
  - dormant: *idle*
  - propagation: *copies itself to other program*
  - triggering: *activated to perform functions*
  - execution: *the function is performed*

Virus structure

- Components:
  - infection mechanism: enables replication
  - trigger: event that makes payload activate
  - payload: what it does, malicious or benign
- Prepended/postpended/embedded
- When infected program invoked, executes virus code then original program code
- Can block initial infection (difficult) or propagation (with access controls)

Virus structure

```
    program V :=

{goto main;
    1234567;

    subroutine infect-executable :=
        {loop:
        file := get-random-executable-file;
        if (first-line-of-file = 1234567)
           then goto loop
           else prepend V to file; }

    subroutine do-damage :=
        {whatever damage is to be done}

    subroutine trigger-pulled :=
        {return true if some condition holds}

main:   main-program :=
        {infect-executable;
        if trigger-pulled then do-damage;
        goto next;}

next:

}
```

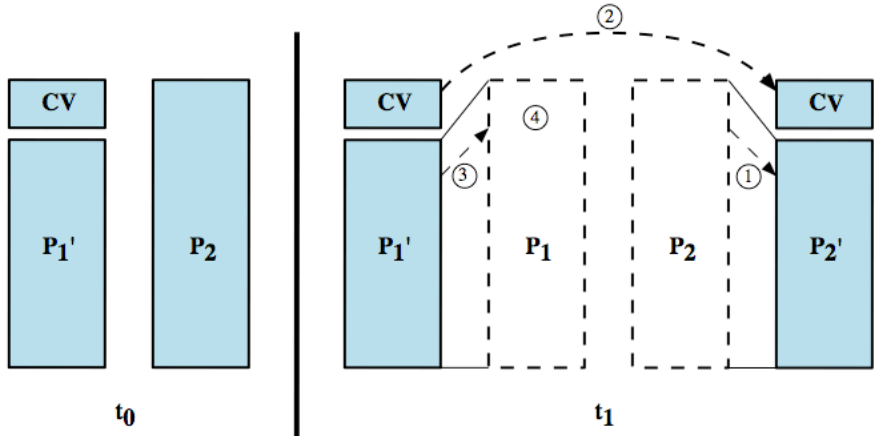
Compression virus

```
    program CV :=

{goto main;
    01234567;

    subroutine infect-executable :=
          {loop:
               file := get-random-executable-file;
          if (first-line-of-file = 01234567) then goto loop;
        (1)     compress file;
        (2)     prepend CV to file;
        }

main:   main-program :=
          {if ask-permission then infect-executable;
        (3)     uncompress rest-of-file;
        (4)     run uncompressed file;}
        }
```



P1 is infected

Virus classification

- By target
  - boot sector: ***infect a master boot record***
  - file infector: ***infects executable OS files***
  - macro virus: ***infects files to be used by an app***
  - multipartite: infects multiple ways
- By concealment
  - encrypted virus: ***encrypted; key stored in virus***
  - stealth virus: ***hides itself (e.g., compression)***
  - polymorphic virus: ***recreates with diff "signature"***
  - metamorphic virus: ***recreates with diff signature and behavior***

Michelangelo Virus

- First reported in April, 1991 in Sweden and the Netherlands.
- Michelangelo is a memory resident infector of diskette boot sectors and the hard disk master boot sector (partition table).
    - Roughly based on the Stoned virus, though very different in its behavior.

Michelangelo Virus

- The Michelangelo virus becomes memory resident the first time the system is attempted to be booted with a Michelangelo infected disk.
- Regardless of whether this boot is successful, Michelangelo will become memory resident.

Michelangelo Virus

- Once Michaelangelo is memory resident, it will infect diskette boot sectors as they are accessed.
- Master boot sector infections will result in the original master boot sector having been moved to Side 0, Cylinder 0, Sector 7 on the hard disk.
- Michelangelo activates on March 6, at which time it will format the system hard disk by overwriting it with random characters from system memory.

Michelangelo Virus Source Code

Jerusalem Virus

- File infector Virus
- Jerusalem is a logic bomb DOS virus first detected at Hebrew University of Jerusalem, in October 1987.
- On infection, the Jerusalem virus becomes memory resident (using 2kb of memory), and then infects every executable file run, except for COMMAND.COM.
- Executable files grow by 1,808 to 1,823 bytes each time they are infected, and are then re-infected each time the files are loaded until they are too large to load into memory.

Jerusalem Virus

- The Jerusalem virus is unique among other viruses of the time, as it is a logic bomb, set to go off on Friday the 13th on all years but 1987 (making its first activation date 13 May 1988).
- Once triggered, the virus not only deletes any program run that day, but also infects .EXE files repeatedly until they grow too large for the computer.
- Since the advent of Windows, these DOS interrupts are no longer used, so Jerusalem and its variants have become obsolete.

Jerusalem Virus Source Code

Macro and scripting viruses

- Became very common in mid-1990s since
  - platform independent
  - infect documents
  - easily spread
- Exploit macro capability of Office apps
  - executable program embedded in office doc
  - often a form of Basic
- More recent releases include protection
- Recognized by many anti-virus programs

E-Mail Viruses

- More recent development
- Melissa
  - exploits MS Word macro in attached doc
  - if attachment opened, macro activates
  - sends email to all on users address list and does local damage

Melissa Virus

- In late March 1999, a programmer named David Lee Smith hijacked an America Online (AOL) account and used it to post a file on an Internet newsgroup named “alt.sex.”
- The posting promised dozens of free passwords to fee-based websites with adult content. When users took the bait, downloading the document and then opening it with Microsoft Word, a virus was unleashed on their computers.
- The Melissa virus started by taking over victims’ Microsoft Word program.
- It then used a macro to hijack their Microsoft Outlook email system and send messages to the first 50 addresses in their mailing lists.
- Those messages, in turn, tempted recipients to open a virus-laden attachment by giving it such names as “sexxxy.jpg” or “naked wife” or by deceitfully asserting, “Here is the document you requested ... don’t show anyone else ;-).” With the help of some devious social engineering, the virus operated like a sinister, automated chain letter.

Melissa Virus

- Email servers at more than 300 corporations and government agencies worldwide became overloaded.
- Approximately one million email accounts were disrupted, and Internet traffic in some locations slowed to a crawl.
- Within a few days, cybersecurity experts had mostly contained the spread of the virus and restored the functionality of their networks, although it took some time to remove the infections entirely.
- An estimated $80 million for the cleanup and repair of affected computer systems.

Melissa Virus Code

Virus countermeasures

- Prevention: ideal solution but difficult
- Realistically need:
  - detection: determine what occurred
  - identification: identify the specific virus
  - removal: remove all traces
- If detected but can't identify or remove, must discard and replace infected program

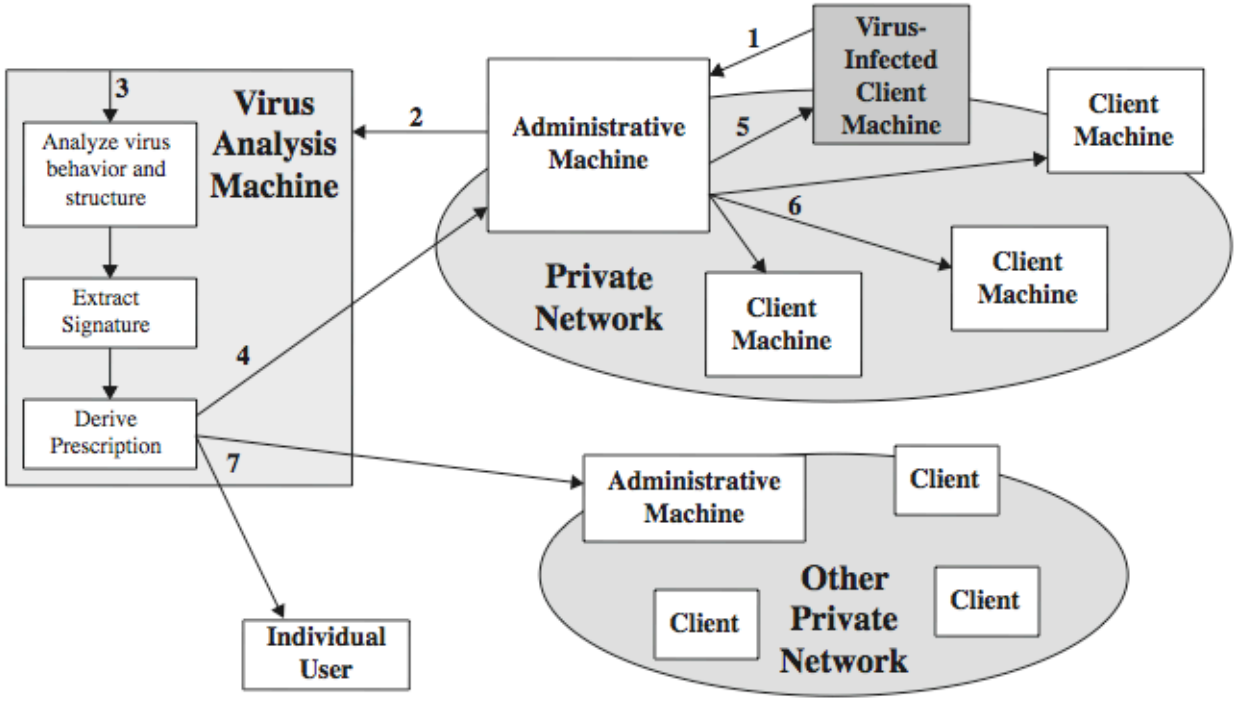
Anti-virus evolution

- Virus & antivirus tech have both evolved
- Early viruses simple code, easily removed
- As viruses become more complex, so did the countermeasures
- Generations
  - first - signature scanners (bit patterns all the same)
  - second – heuristics (integrity checks; checksums)
  - third - identify actions (find by actions they do)
  - fourth - combination packages

Generic decryption

- Runs executable files through GD scanner:
  - CPU emulator to interpret instructions
  - virus scanner to check known virus signatures
  - emulation control module to manage process
- Lets virus decrypt itself in interpreter
- Periodically scan for virus signatures
- *Let virus do the work for an antivirus program by exposing it in a controlled environment*

Digital immune system

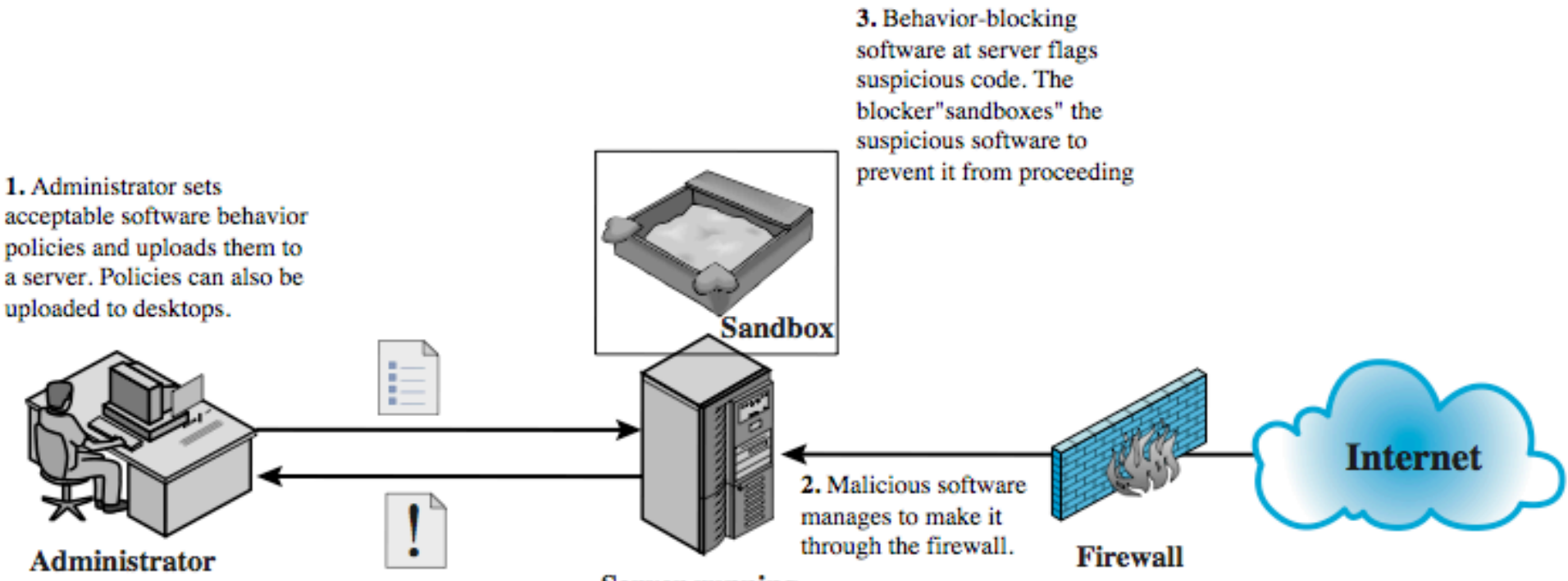


1. A monitoring pgm infers a virus, sends a copy to an adm machine
2. Adm encrypts, sends to a central analysis machine
3. Central analysis: Safe exec of virus, analyze, give a prescription
4. Prescription sent back to the adm machines
5. Adm machine forwards to all clients
6. Prescription forwarded to other organizations
7. Subscribers worldwide receive regular updates

IBM/Symantec Project

Behavior-blocking software

Integrates with the OS; looks for bad behavior



Monitored behaviors:

- -Attempts to open, view, delete, modify files
- Attempts to format drives
- Modifications to the logic of executables
- Modifications to critical system settings
- Scripting of emails to send exec contents

Worms

- Replicating program that propagates over net
  - using email, remote exec, remote login
- Has phases like a virus:
  - dormant, propagation, triggering, execution
  - propagation phase: searches for other systems, connects to it, copies self to it and runs
- May disguise itself as a system process
- Concept seen in Brunner's "Shockwave Rider"
- Implemented by Xerox Palo Alto labs in 1980's

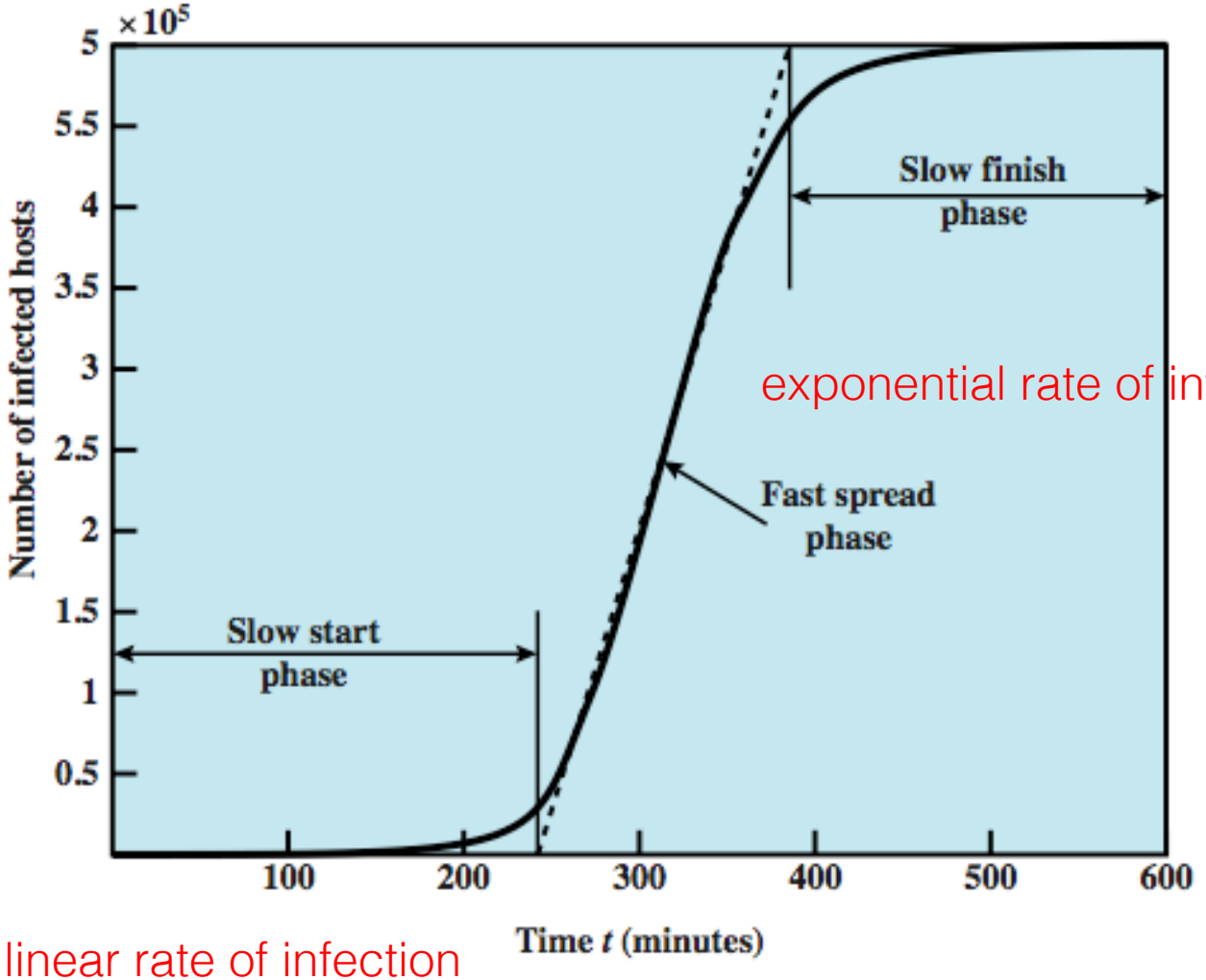
Morris worm

- One of best know worms
- Released by Robert Morris in 1988
  - Affected 6,000 computers; cost $10-$100 M
- Various attacks on UNIX systems
  - cracking password file to use login/password to logon to other systems
  - exploiting a bug in the finger protocol
  - exploiting a bug in sendmail
- If succeed have remote shell access
  - sent bootstrap program to copy worm over

Morris Worm Code

Worm Propagation Model

(based on recent attacks)



exponential rate of infection

linear rate of infection

More recent worm attacks

- Code Red
  - July 2001 exploiting MS IIS bug
  - probes random IP address, does DDoS attack
  - consumes significant net capacity when active
  - 360,000 servers in 14 hours
- Code Red II variant includes backdoor: hacker controls the worm
- SQL Slammer ***(exploited buffer-overflow vulnerability)***
  - early 2003, attacks MS SQL Server
  - compact and very rapid spread
- Mydoom ***(100 M infected messages in 36 hours)***
  - mass-mailing e-mail worm that appeared in 2004
  - installed remote access backdoor in infected systems

Code Red

- Code Red
  - July 2001 exploiting MS IIS bug
  - probes random IP address, does DDoS attack
  - consumes significant net capacity when active
  - 360,000 servers in 14 hours -
    - https://www.caida.org/archive/code-red/newframes-small-log.gif
- Code Red II variant includes backdoor: hacker controls the worm
- SQL Slammer ***(exploited buffer-overflow vulnerability)***
  - early 2003, attacks MS SQL Server
  - compact and very rapid spread
- Mydoom ***(100 M infected messages in 36 hours)***
  - mass-mailing e-mail worm that appeared in 2004
  - installed remote access backdoor in infected systems

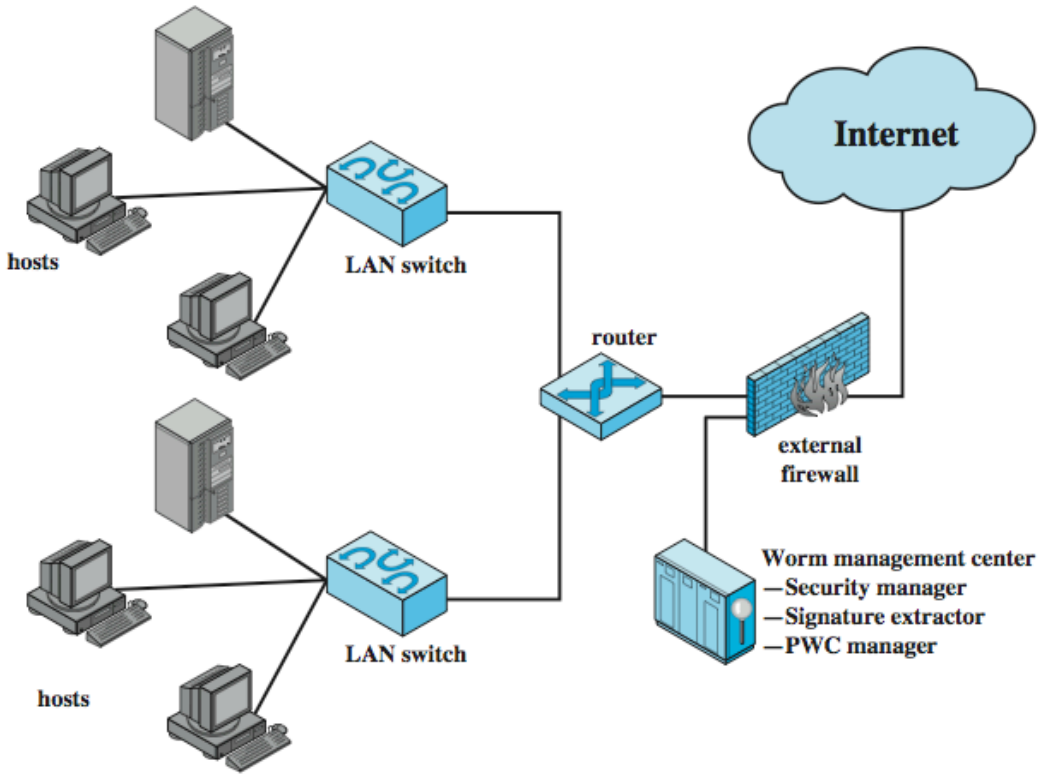
State of worm technology

- Multiplatform: not limited to Windows
- Multi-exploit: Web servers, emails, file sharing …
- Ultrafast spreading: do a scan to find vulnerable hosts
- Polymorphic: each copy has a new code
- Metamorphic: change appearance/behavior
- Transport vehicles (e.g., for DDoS)
- Zero-day exploit of unknown vulnerability (to achieve max surprise/distribution)

Worm countermeasures

- Overlaps with anti-virus techniques
- Once worm on system A/V can detect
- Worms also cause significant net activity
- Worm defense approaches include:
  - signature-based worm scan filtering: define signatures
  - filter-based worm containment (focus on contents)
  - payload-classification-based worm containment (examine packets for anomalies)
  - threshold random walk scan detection (limit the rate of scan-like traffic)
  - rate limiting and rate halting (limit outgoing traffic when a threshold is met)

Proactive worm containment



1. PWC agent monitors outgoing traffic for increased activity

2. When an agent notices high traffic, it informs the PWC manager; mgr propagates to other hosts

3. Hosts receive alert and decide if to ignore (based on time of last incoming pkt)

4. Relaxation period (based on threshold)

Mobile code

- Scripts, macros or other portable instructions
- Popular ones: JavaScript, ActiveX, VBScript
- Heterogeneous platforms
- From a remote system to a local system
- Can act as an agent for viruses, works, and Trojan horses
- Mobile phone works: communicate the Bluetooth connections (e.g., CommWarrior on Symbian but attempts on Android and iPhone)

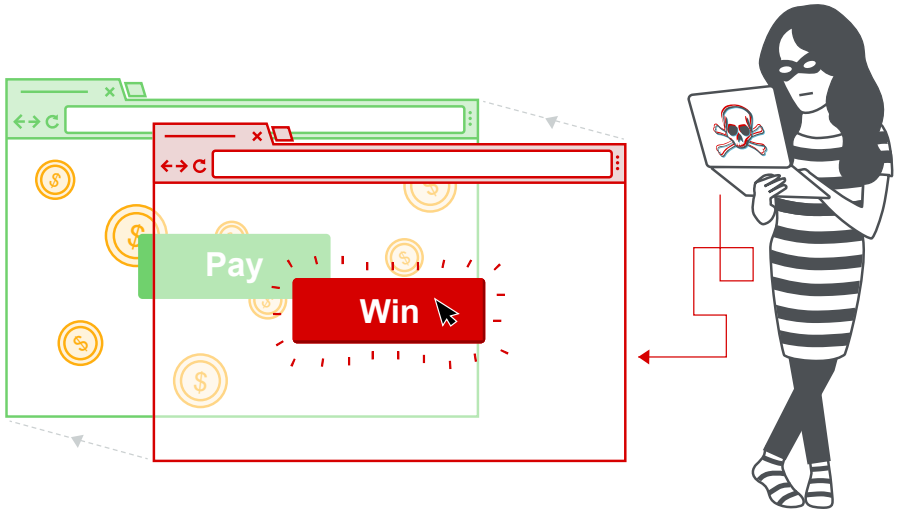
Client-side vulnerabilities

- Drive-by-downloads: common in recent attacks
- Exploits browser vulnerabilities (when a user visits a website controlled by the attacker or a compromised website)
- Clickjacking

Clickjacking

- Clickjacking is an interface-based attack in which a user is tricked into clicking on actionable content on a hidden website by clicking on some other content in a decoy website.

- A web user accesses a decoy website (perhaps this is a link provided by an email) and clicks on a button to win a prize.



Clickjacking

```
<head>
        <style>
                #target_website {
                        position:relative;
                        width:128px;
                        height:128px;
                        opacity:0.00001;
                        z-index:2;
                        }
                #decoy_website {
                        position:absolute;
                        width:300px;
                        height:400px;
                        z-index:1;
                        }
        </style>
</head>
...
<body>
        <div id="decoy_website">
        ...decoy web content here...
        </div>
        <iframe id="target_website" src="https://vulnerable-website.com">
        </iframe>
</body>
```

- The target website iframe is positioned within the browser so that there is a precise overlap of the target action with the decoy
- Absolute and relative position values are used to ensure that the target website accurately overlaps the decoy regardless of screen size, browser type and platform.
- The z-index determines the stacking order of the iframe and website layers.
- The opacity value is defined as 0.0 (or close to 0.0) so that the iframe content is transparent to the user.

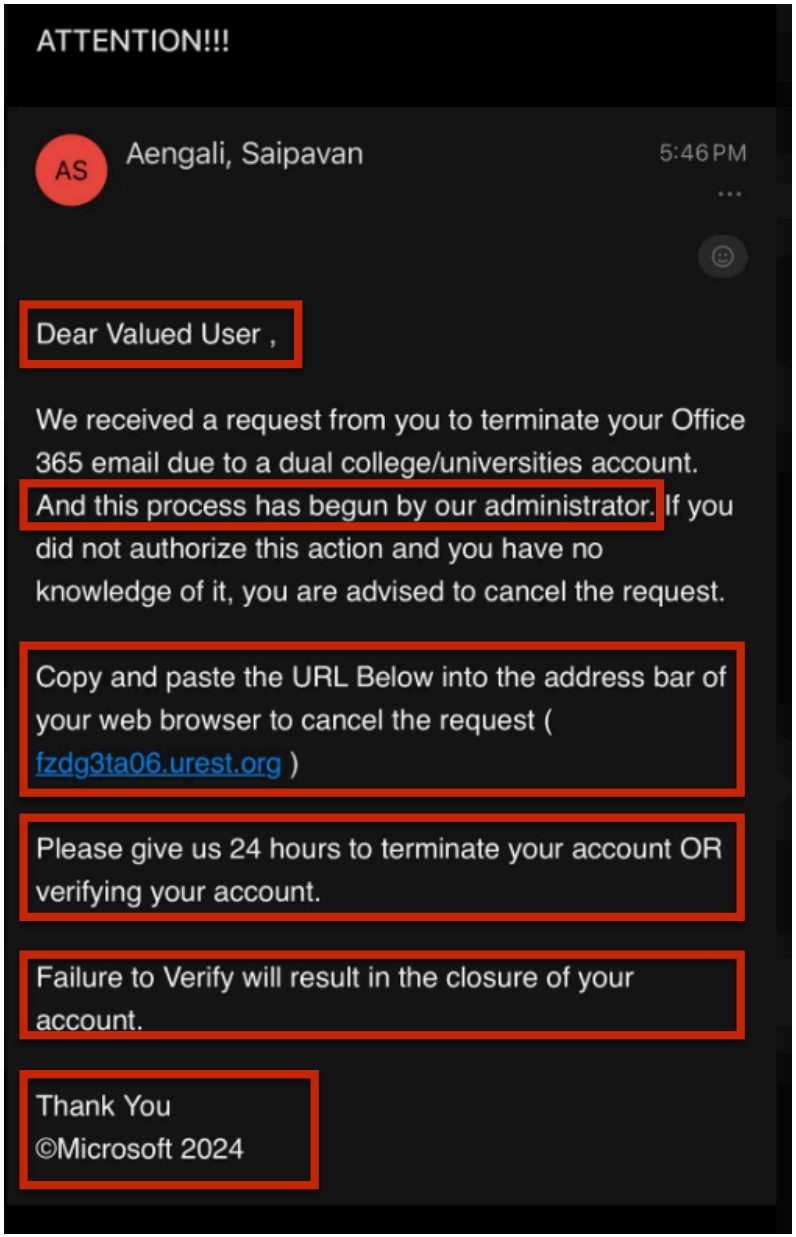
Social engineering, spam, email, Trojans

- Spam (much better protection now)
- Trojan horse: looks like a useful tool but contains hidden code

Spam / Phishing

- Tip 1: Name of Sender
- Tip 2: Domain Name
- Tip 3: Check Spelling and Grammar
    - Spammers often make spelling and grammar errors in their emails. If you receive an email from a organization like your bank filled with these errors, you can probably call it safe to assume it is not legit.
- Tip 4: Generic Greetings
    - Check for generic greetings such as "Dear Valued Customer". When you apply to online services, you usually provide them with your first name, so they should already have it and be capable of including it in their email to you.
- Tip 5: Urgency
    - Professional organizations, such as your bank will not require you to perform any sort of action in an extremely limited amount of time. If an email requires you to do something with X amount of hours, that is often a good indication of spam.
- Tip 6: Don't Click Links
    - if an organization, such as your bank requires you to log in to your account to do something, be suspicious about links that apparently take you to their log in page. As an alternative, log on to the website the way you normally would in your browser and navigate from there.
- Tip 7: Personal Information
    - Pay attention to requests for personal information. Organizations such as eBay, Amazon, PayPal, and your bank will never randomly ask you for personal information, especially over email.

How Many Red Flags?



https://www.troyhunt.com/thanks-fedex-this-is-why-we-keep-getting-phished/

Payload

- Data destruction, theft
- Data encryption (ransomware)
- Real-world damage
  - Stuxnet: caused physical damage also (targeted to Siemens industrial control software)
- Logic bomb

Payload attack agents: bots (zombie/drone)

- Program taking over other computers and launch attacks
  - hard to trace attacks
- If coordinated form a ***botnet***
- Characteristics:
  - **remote control facility** (distinguishing factor)
    - via IRC/HTTP etc
  - spreading mechanism
    - attack software, vulnerability, scanning strategy
- Various counter-measures applicable (IDS, honeypots, …)

Uses of bots

- DDoS
- Spamming
- Sniffing traffic
- Keylogging
- Spreading malware
- Installing advertisement
- Manipulating games and polls

Payload: information theft

- Credential theft, key loggers, spyware
- Phishing identify theft
- Spear phishing (act as a trusted source for a specific target)

Payload: rootkits and backdoor

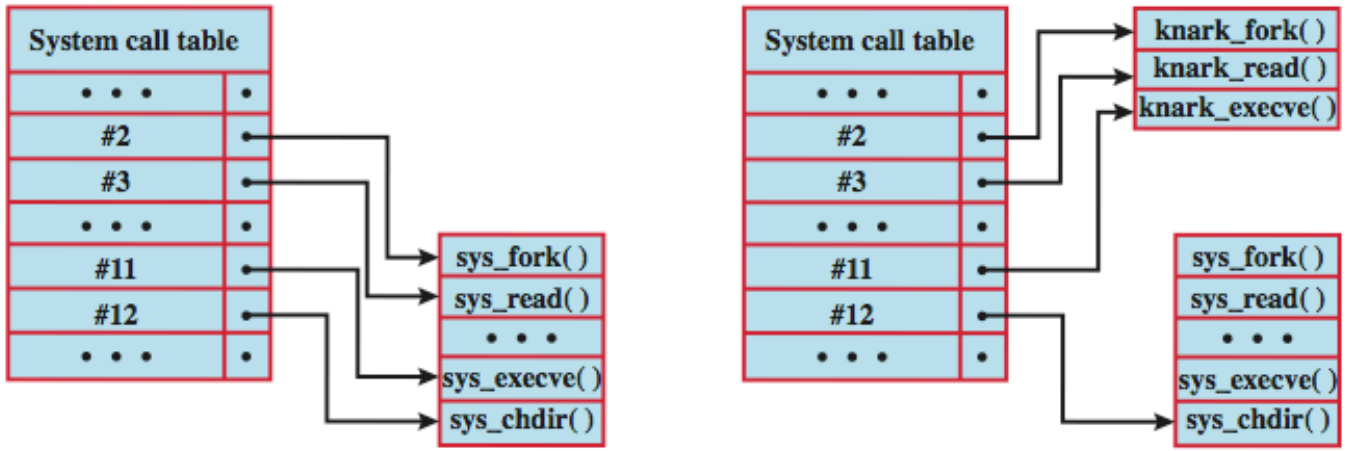
- Set of programs installed for admin access
- Malicious and stealthy changes to host O/S
- May hide its existence
  - subverting report mechanisms on processes, files, registry entries etc
- May be persistent (survives reboot) or memory-based

- Do not rely on vulnerabilities
  - installed via Trojan
  - installed via hackers
- Backdoor: often by programmers

Rootkit System Table Mods

A Unix Example

User API calls refer to a number; the system maintains a system call table with one entry per number; each number is used to index to a corresponding system routine



(a) Normal kernel memory layout

(b) After nkark install

rootkit modifies the table and the calls go to the hackers replacements

Countermeasures

- Prevention
- Detection, identification, removal
- Requirement
  - generality
  - Timeliness
  - Resiliency
  - Minimal DoS costs
  - Transparency
  - Global/local coverage (inside and outside attackers)

https://www.youtube.com/watch?v=3PAauAy-Fb4

Break

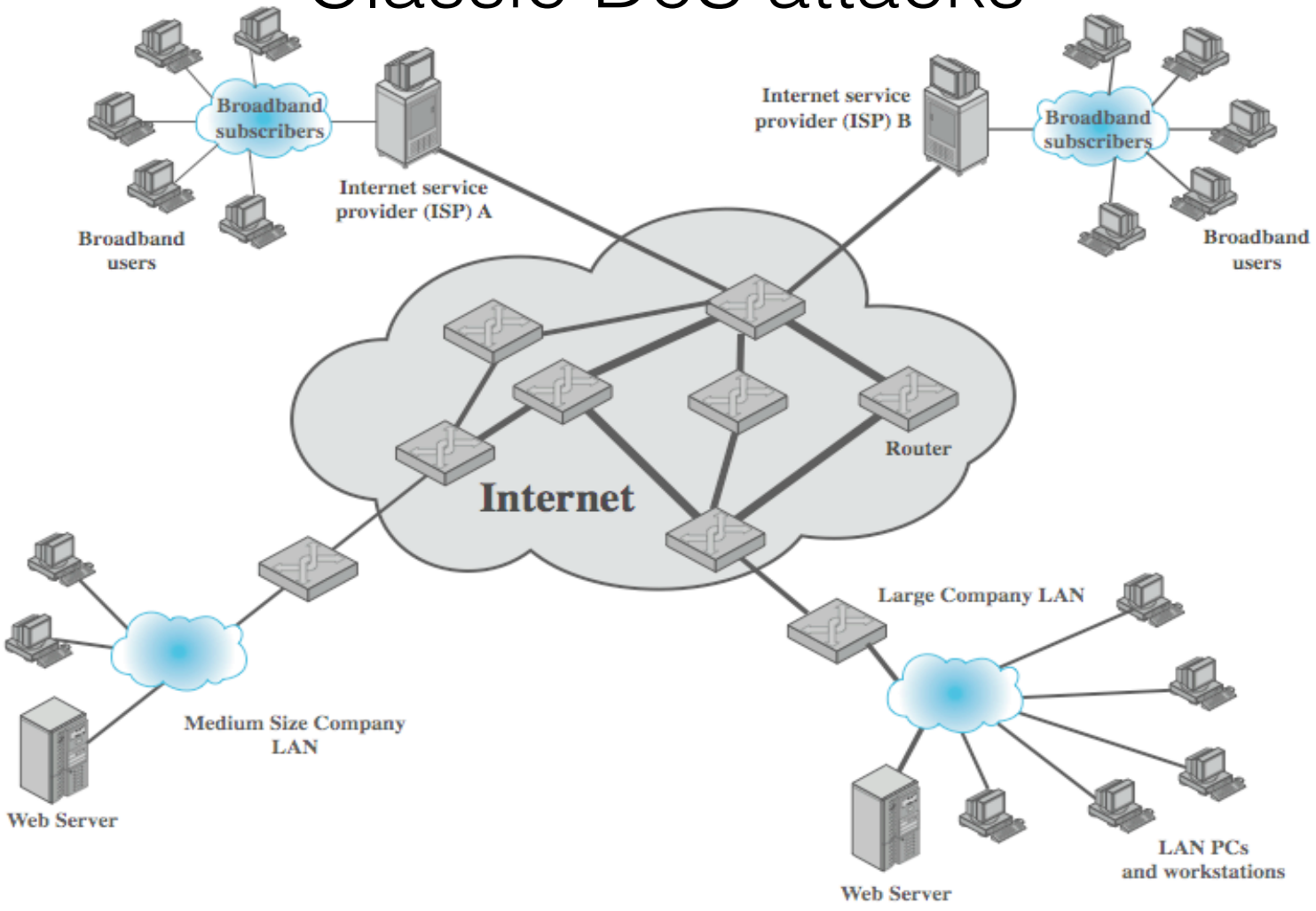
Denial-of-service

- **Denial of service** (DoS) 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
- Attacks (overload or invalid request services that consume significant resources)
  - network bandwidth
  - system resources
  - application resources
- Have been an issue for some time (25% of respondents to an FBI survey)

Classic DoS attacks

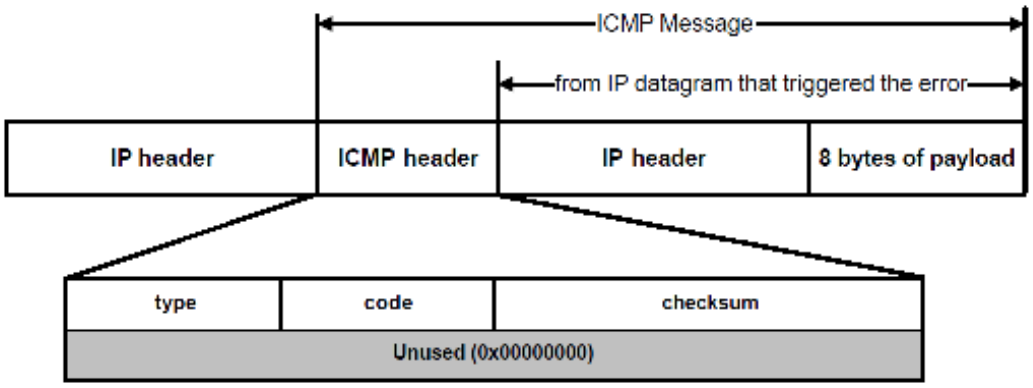
- Flooding ping command
  - Aim of this attack is to overwhelm the capacity of the network connection to the target organization
  - Traffic can be handled by higher capacity links on the path, but packets are discarded as capacity decreases
- Source of the attack is clearly identified unless a spoofed address is used
- Network performance is noticeably affected

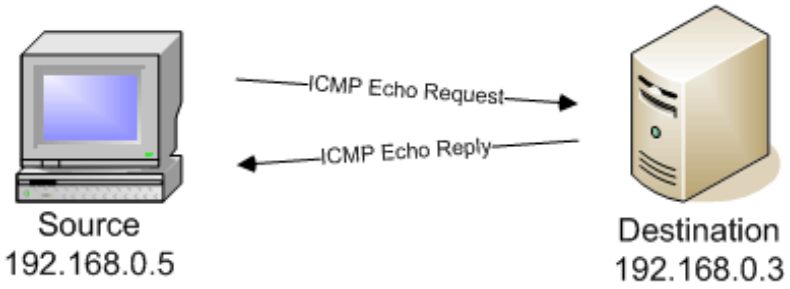
Classic DoS attacks



Internet Control Message Protocol (ICMP)

- The Internet Control Message Protocol (**ICMP**) is one of the main IP protocols; it is used by network devices, like routers, to send error messages indicating (e.g., a requested service is not available or a host or router could not be reached)





The host must respond to all echo requests with an echo reply containing the exact data received in the request message

Source address spoofing

- Use forged source addresses
  - Usually via the raw socket interface on operating systems
  - Makes attacking systems harder to identify
- Attacker generates large volumes of packets that have the target system as the destination address
- Congestion would result in the router connected to the final, lower capacity link
- Backscatter traffic
  - Advertise routes to unused IP addresses to monitor attack traffic

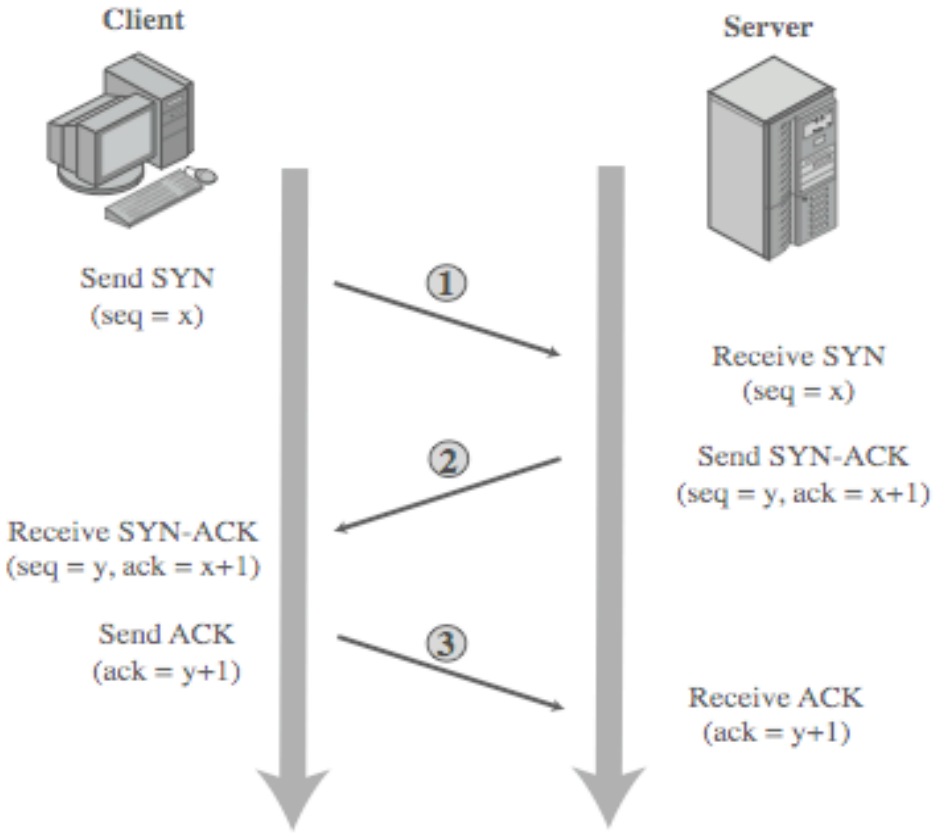
Backscatter traffic

- Security researchers (Honeypot Project) advertise blocks of unused IP addresses (no real/legit uses)
- If ICMP/connection request is made, most likely from attackers
- Monitoring provides valuable info on the type and scale of attack

SYN spoofing

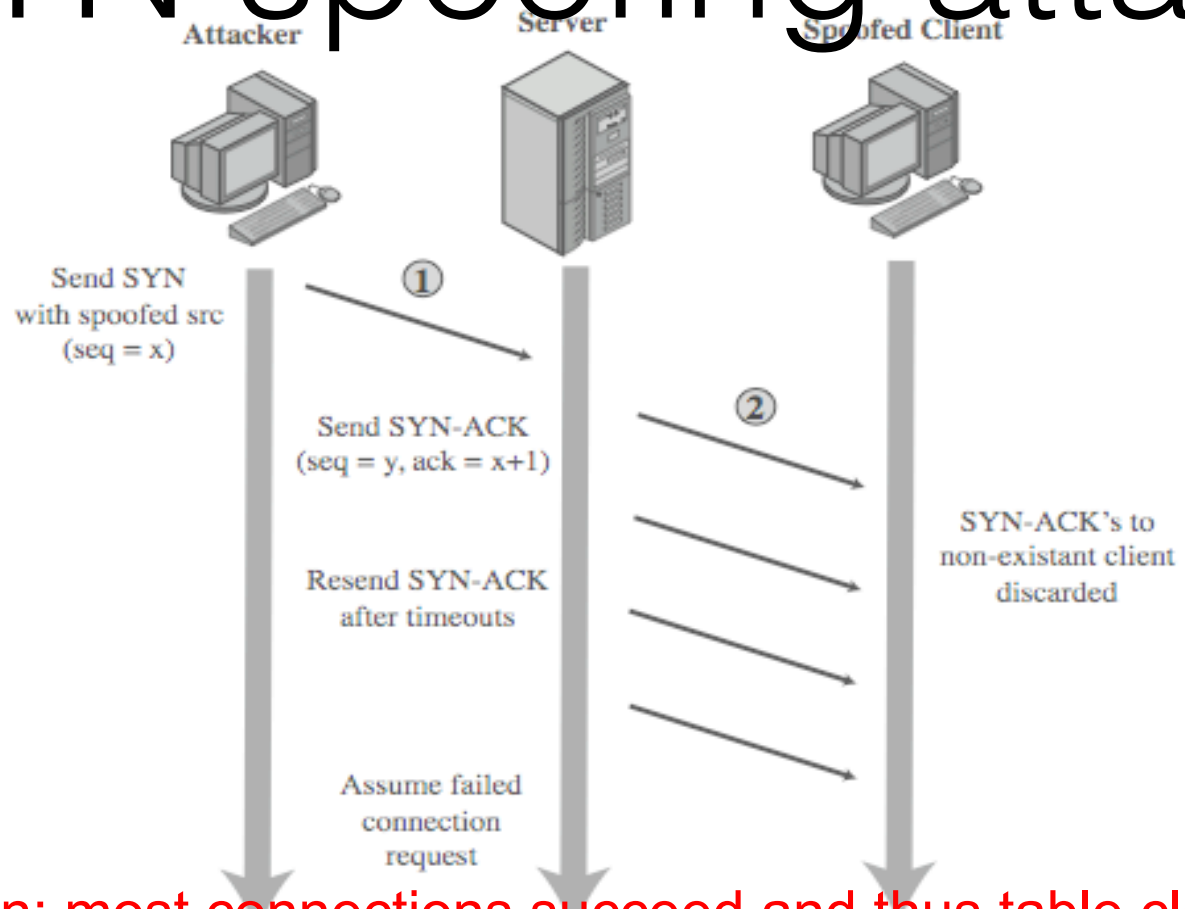
- Common DoS attack
- Attacks the ability of a server to respond to future connection requests by overflowing the tables used to manage them
- Thus legitimate users are denied access to the server
- Hence an attack on system resources, specifically the network handling code in the operating system

TCP connection handshake



syn/ack pkts
y= server seq#
x= client seq#

SYN spoofing attack



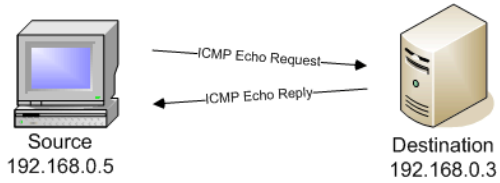
assumption: most connections succeed and thus table cleared quickly

SYN spoofing attack: attacker's source

- Attacker often uses either
  - random source addresses (addresses that may not exist)
  - or that of an overloaded server (that may not send a RST)
  - to block return of (most) reset packets
- Has much lower traffic volume
  - attacker can be on a much lower capacity link
- Objective: uses addresses that will not respond to the SYN-ACK with a RST

Types of flooding attacks

- Classified based on network protocol used
- Objective: to overload the network capacity on some link to a server
- Virtually any type of network packet can be used



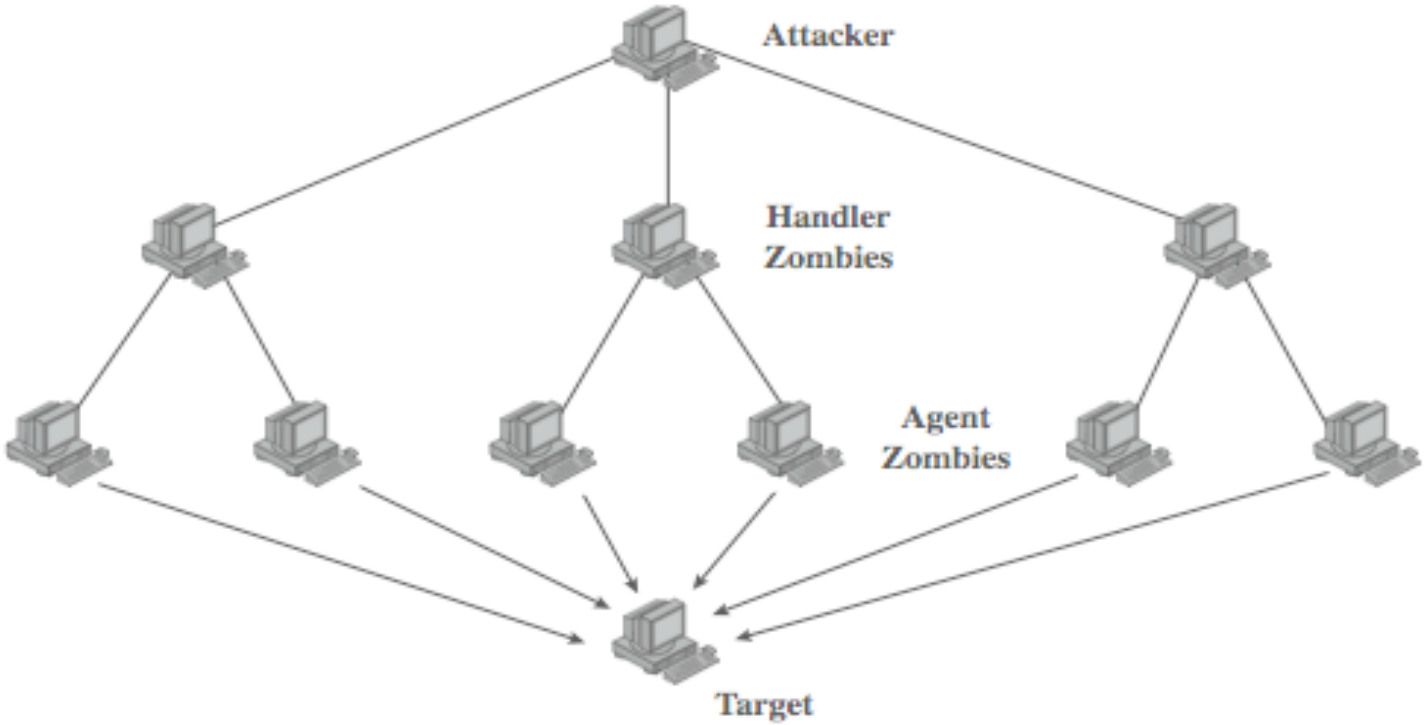
- ICMP Flood
  - Uses ICMP packets, eg ping (echo) request
  - Typically allowed through, some required
- UDP Flood
  - Alternative uses UDP packets to random ports (even if no service is available, attacker achieves its goal)
- TCP SYN Flood (SYN spoof vs SYN flood)
  - Sends TCP SYN (connection request) packets
  - But for volume attack

Distributed DoS attacks

- Have limited volume if single source used
- Multiple systems allow much higher traffic volumes to form a distributed DoS (DDoS) attack
- Often compromised PC's/workstations
  - Zombies with backdoor programs installed
  - Forming a botnet
- Example: Tribe Flood Network (TFN), TFN2K
  - did ICMP, SYN, UDPF and ICMP floods

DDoS control hierarchy

Attacker sends one command to the handler zombies;
the handler forwards to other handlers, agents



SYN Flood Attack

- Syn flood is also known as a half-open attack. In this attack, the attacker sends multiple connection requests to perform the distributed denial of service attack.

```
# hping3 -S -p 80 Target –flood
```

LAND Attack

- A LAND (local area network denial) attack is a DoS (attack that consists of sending a special poison spoofed packet to a computer, causing it to lock up.

- The security flaw was first discovered in 1997 by someone using the alias "m3lt", and has resurfaced many years later in operating systems such as Windows Server 2003 and Windows XP SP2.

#hping3 -S -p 80 127.0.0.1 -a 127.0.0.1

Random Source Attack

- An attacker can send multiple random packets with different source addresses to the target machine which may cause the Distributed denial of service attack. It is difficult to identify the actual source address after an incident occurs.Most devices on a network will, by default, respond to this by sending a reply to the source IP address.

```
hping3 -S -p 80 Target –flood –rand-source
```

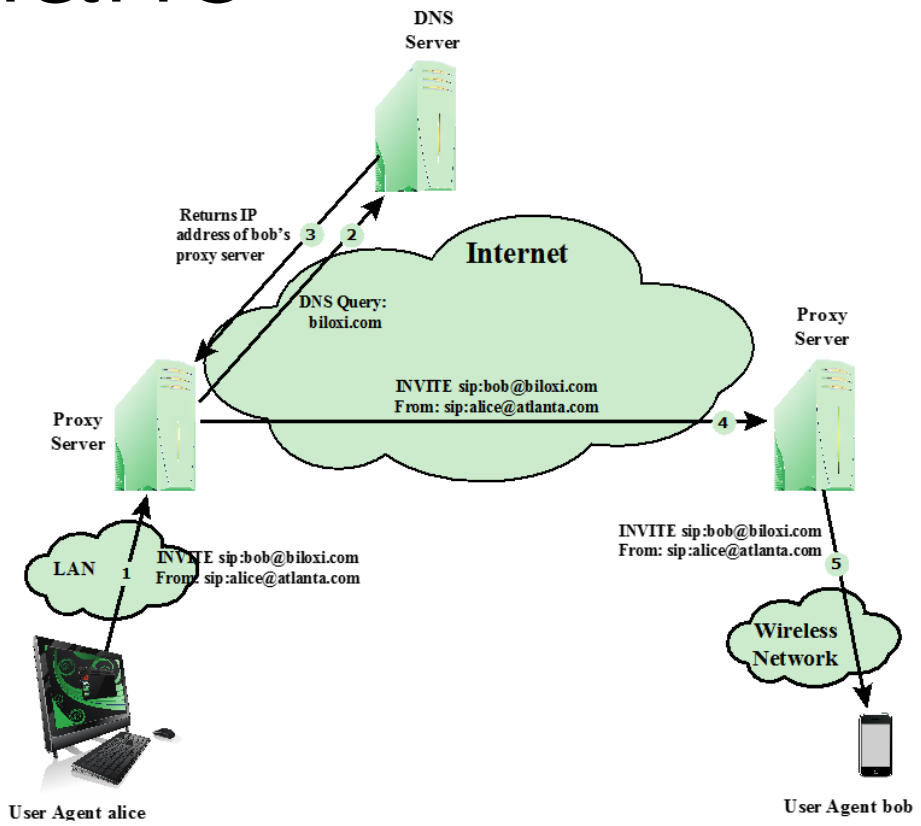
Application-based bandwidth attacks

- Force the victim system to execute resource-consuming operations (e.g., searches, complex DB queries)

- VoIP Session Initiation Protocol (SIP) flood: attacker sends many INVITE requests; major burden on the proxies
  - server resources depleted while handling requests
  - bandwidth capacity is consumed

SIP invite scenario

- Standard protocol for VoIP telephony
- Text-based protocol with a syntax similar to that of HTTP
- Two types of SIP messages: requests and responses



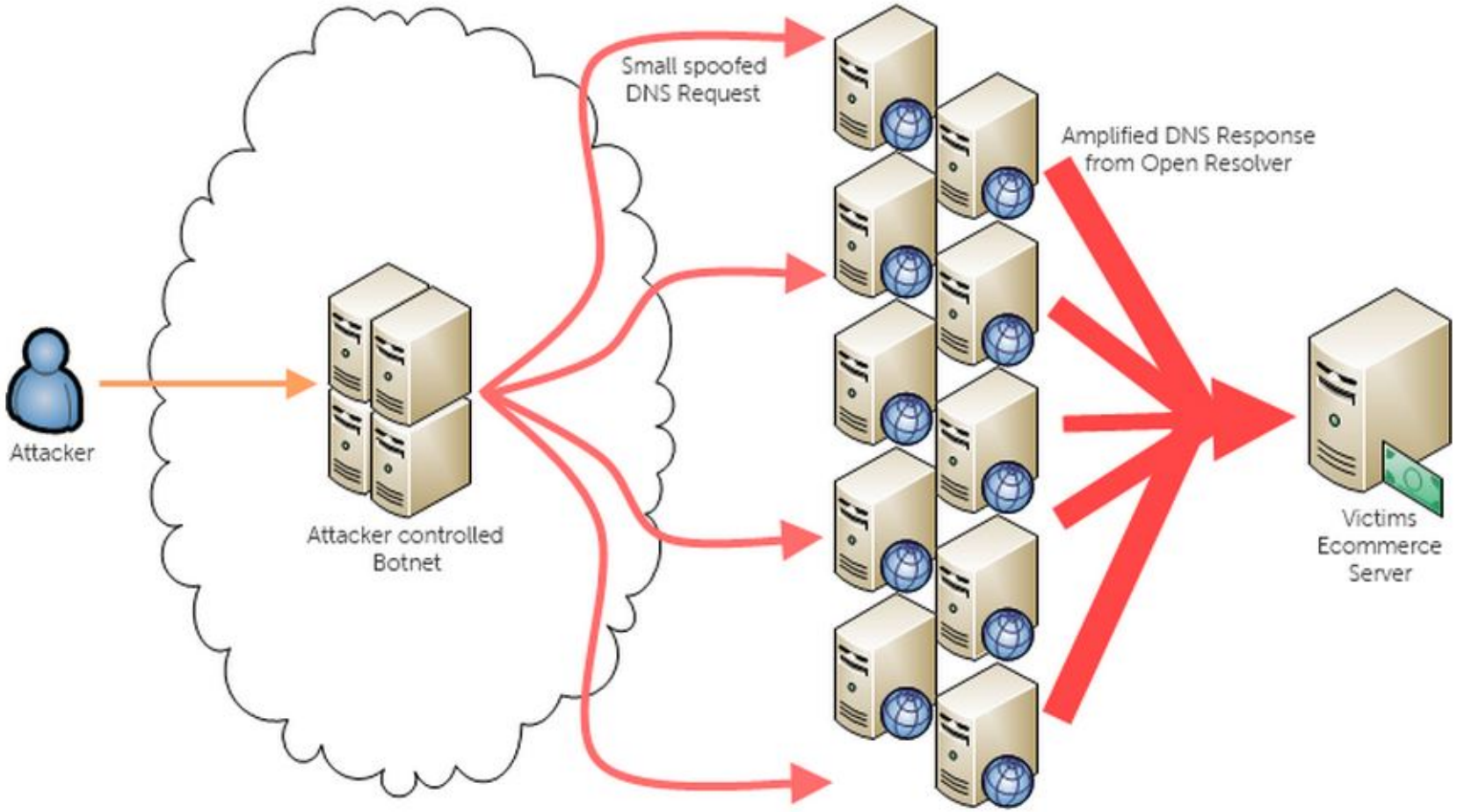
HTTP-based attacks

- Attempts to monopolize by sending HTTP requests that never complete
- Eventually consumes Web server's connection capacity
- Utilizes legitimate HTTP traffic
- Spidering: Bots starting from a given HTTP link and following all links on the provided Web site in a recursive way
- Existing intrusion detection and prevention solutions that rely on signatures to detect attacks will generally not recognize Slowloris

Reflection attacks

- Attacker sends packets to a known service on the intermediary with a spoofed source address of the actual target system
- When intermediary responds, the response is sent to the target
- “Reflects” the attack off the intermediary (reflector)
- Goal is to generate enough volumes of packets to flood the link to the target system without alerting the intermediary
- The basic defense against these attacks is blocking spoofed-source packets

Reflection attacks



SMURF Attack

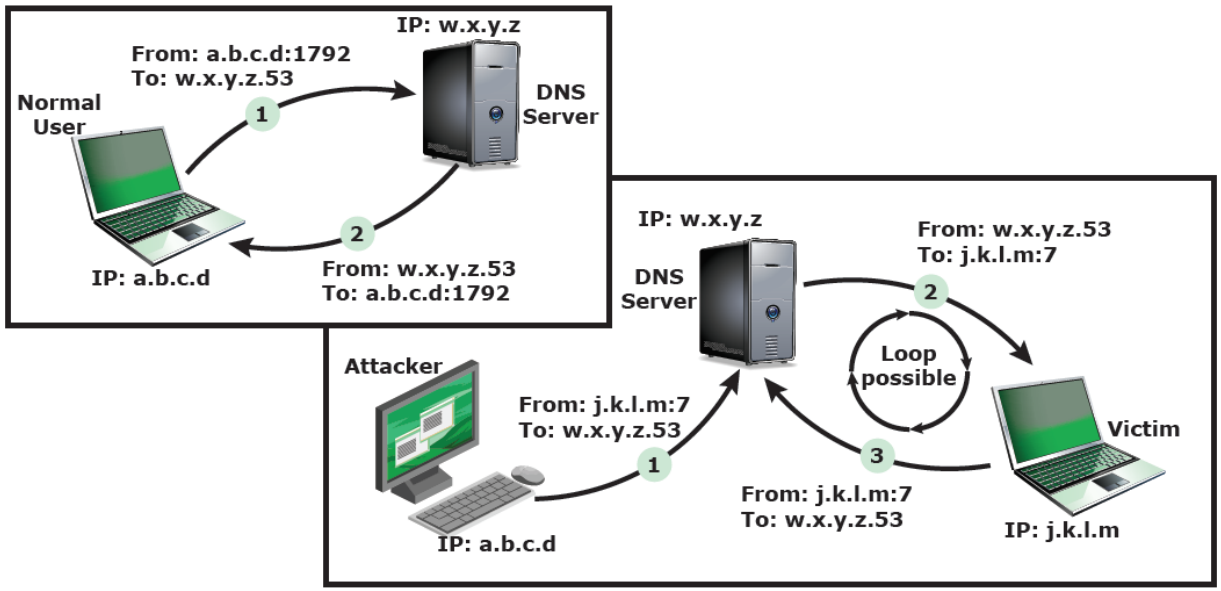
- Distributed denial-of-service attack in which large numbers of Internet Control Message Protocol (ICMP) packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP broadcast address.
- Most devices on a network will, by default, respond to this by sending a reply to the source IP address.

```
hping3 -1 --flood -a 192.168.33.123 192.168.1.255
```

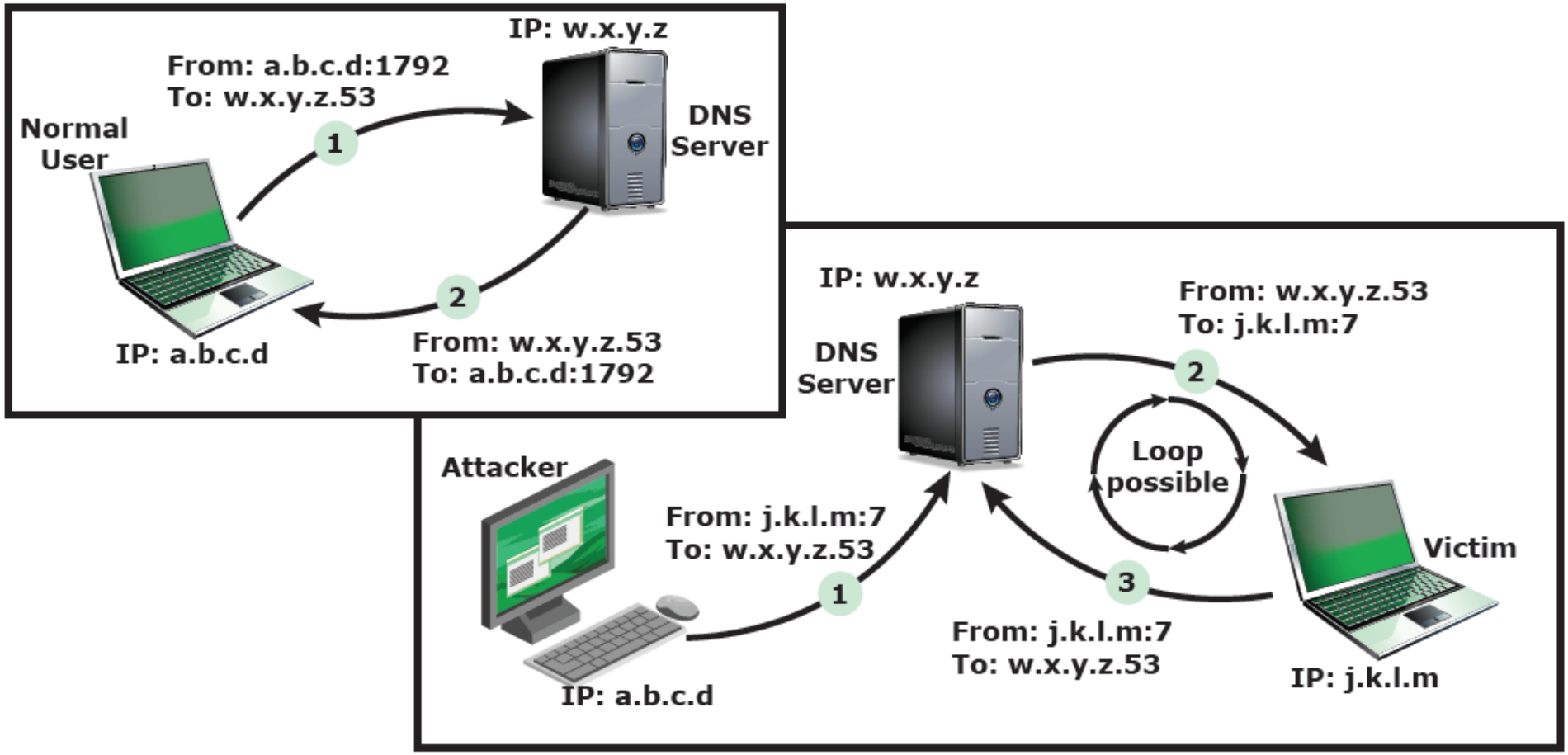
When both the source and destination address in the original packet are set to the broadcast address of the target network, things start to get out of hand quickly. All hosts receive an echo request, but all replies to that are broadcast again to all hosts. Each host will receive an initial ping, broadcast the reply and get a reply from all n-1 hosts. An amplification factor of n for a single host, but an amplification factor of n^2 for the network.

Reflection attacks

- Further variation creates a self-contained loop between intermediary and target (attacker spoofs using port 7 requiring echoes)
- Fairly easy to filter and block



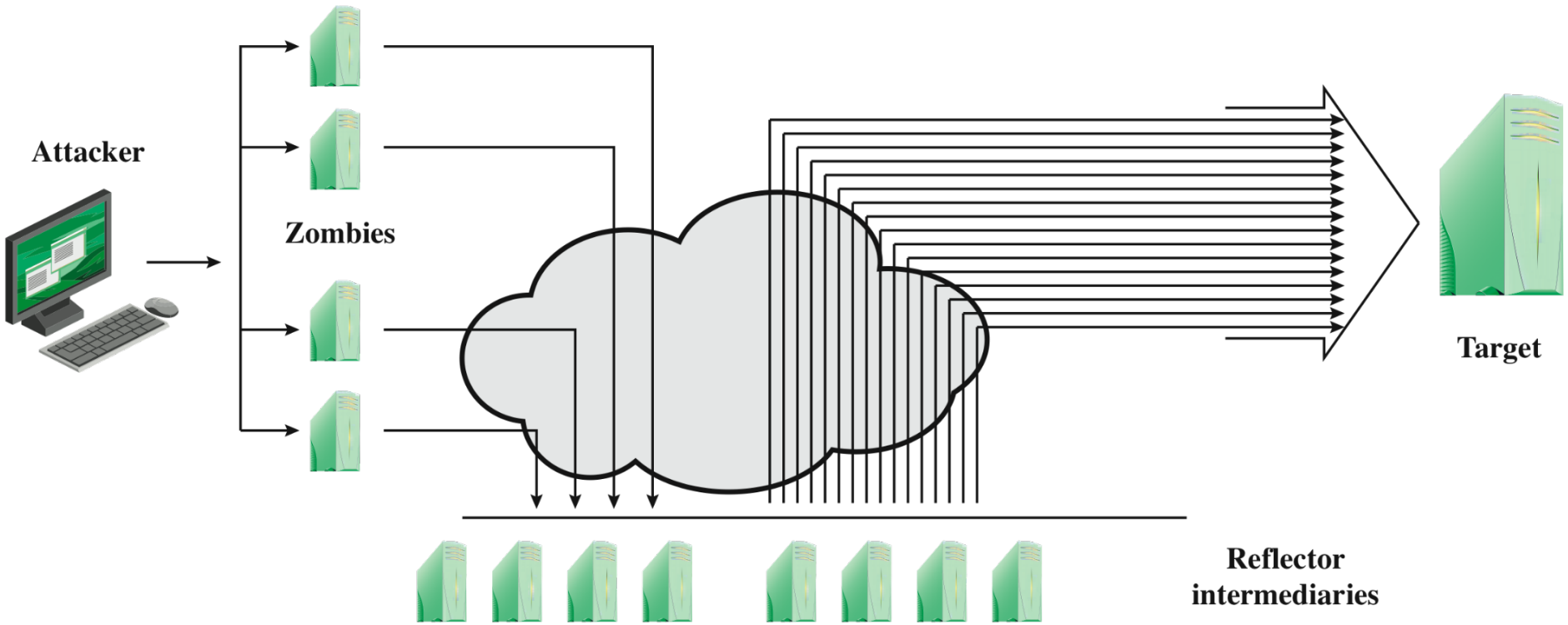
DNS reflection attacks



DNS amplification attacks

- Use packets directed at a legitimate DNS server as the intermediary system
- Attacker creates a series of DNS requests containing the spoofed source address of the target system
- Exploit DNS behavior to convert a small request to a much larger response (amplification)
- Target is flooded with responses
- Basic defense against this attack is to prevent the use of spoofed source addresses

Amplification attacks



Can take advantage of broadcast address of some network

Four lines of defense against DDoS attacks

- Attack prevention and preemption (before attack)
- Attack detection and filtering (during the attack)
- Attack source traceback and identification (uring and after the attack)
- Attack reaction (after the attack)

DoS attack prevention

- Block spoofed source addresses
  - On routers as close to source as possible
- Filters may be used to ensure path back to the claimed source address is the one being used by the current packet
  - Filters must be applied to traffic before it leaves the ISP's network or at the point of entry to their network
- Use modified TCP connection handling code
  - Cryptographically encode critical information in a cookie that is sent as the server's initial sequence number
  - Legitimate client responds with an ACK packet containing the incremented sequence number cookie
  - Drop an entry for an incomplete connection from the TCP connections table when it overflows

Attack prevention

- Rate controls in upstream distribution nets
  - On specific packets types e.g. some ICMP, some UDP, TCP/SYN
  - *Impose limits*
- Use modified TCP connection handling
  - Server sends SYN ***cookies*** when table full (reconstruct table data from the cookie from legit clients)
  - Sr selective or random drop when table full

Attack prevention

- Block IP directed broadcasts
- Block suspicious services and combinations
- Manage application attacks with a form of graphical puzzle (captcha) to distinguish legitimate human requests
- Use mirrored and replicated servers when high-performance and reliability is required

Responding to attacks

- Good incidence response plan
  - Details on how to contact technical personal for ISP
  - Needed to impose traffic filtering upstream
  - Details of how to respond to the attack
- Implement anti-spoofing, directed broadcast, and rate limiting filters
- Ideally have network monitors and IDS to detect and notify abnormal traffic patterns

Responding to attacks

- Identify type of attack
  - Capture and analyze packets
  - Design filters to block attack traffic upstream
  - Or identify and correct system/application bug
- Have ISP trace packet flow back to source
  - May be difficult and time consuming
  - Necessary if planning legal action
- Implement contingency plan
  - Switch to alternate backup servers
  - Commission new servers at a new site with new addresses
- Update incident response plan