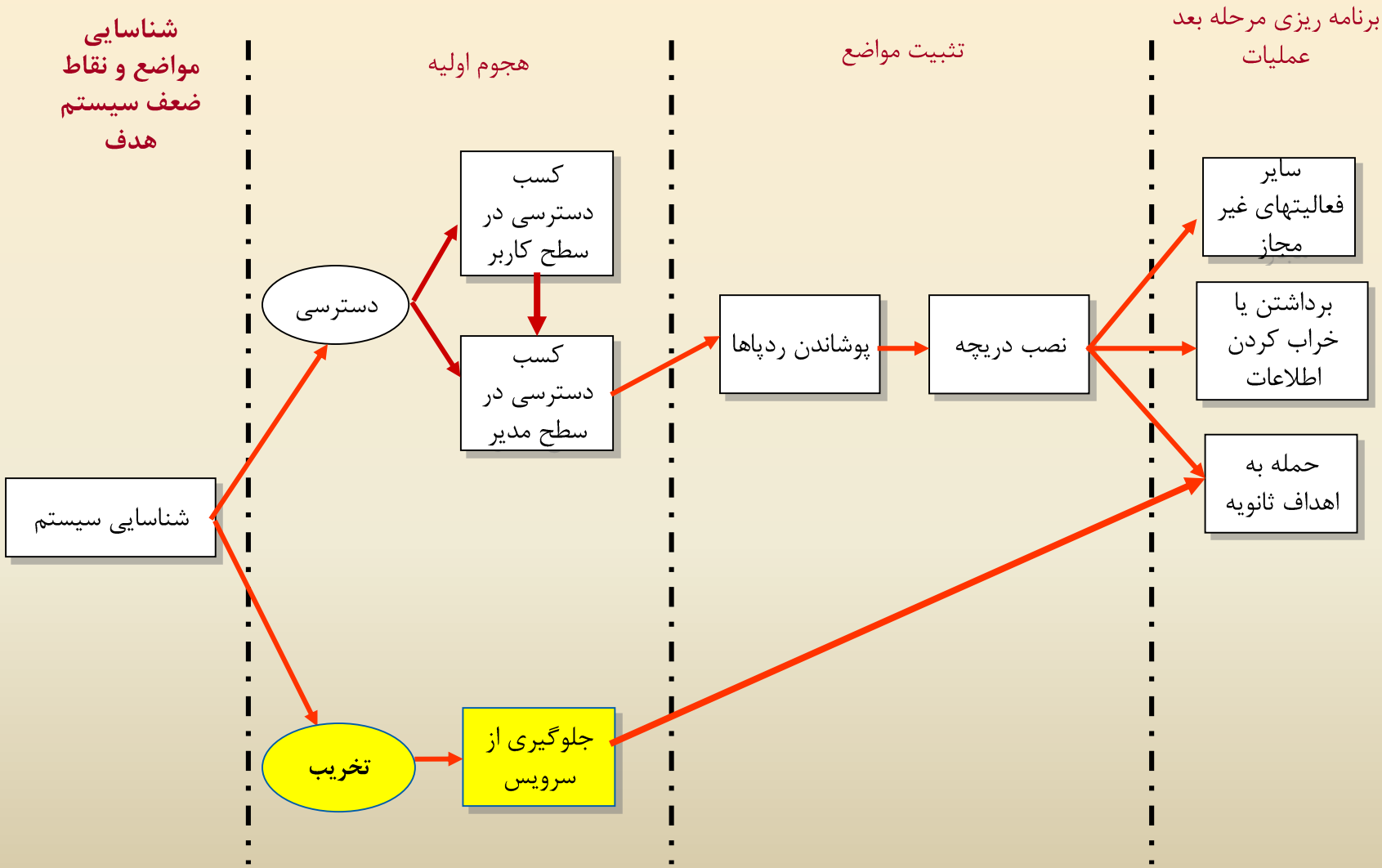
مروری بر نفوذگری و امنیت در سیستم‌های کامپیوتری

هجوم به قصد تخریب

علی فانیان
دانشگاه صنعتی اصفهان

روند نمای کلی انجام یک حملهٔ کامپیوتری



Contents

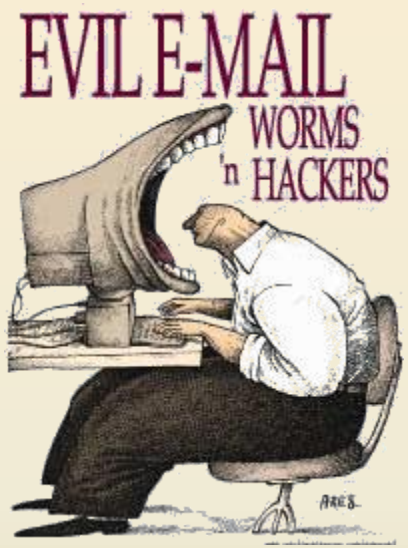
- Denial of Service attacks
  - Concepts
  - Samples of attacks
- Malicious Logic attacks
  - Concepts
  - Viruses

Denial of Service Attack

- “Attack in which the primary goal is to deny the victim(s) access to a particular resource.”
- Possible impacts: reboot your computer, Slows down computers- Certain sites, Applications become inaccessible

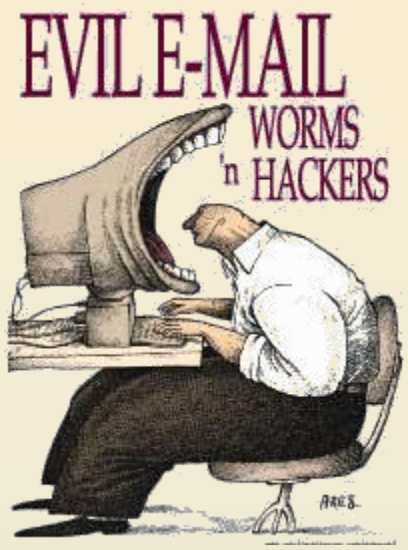
Results expected

- Denial-of-service attacks can essentially disable your computer or your network. Depending on the nature of your enterprise.



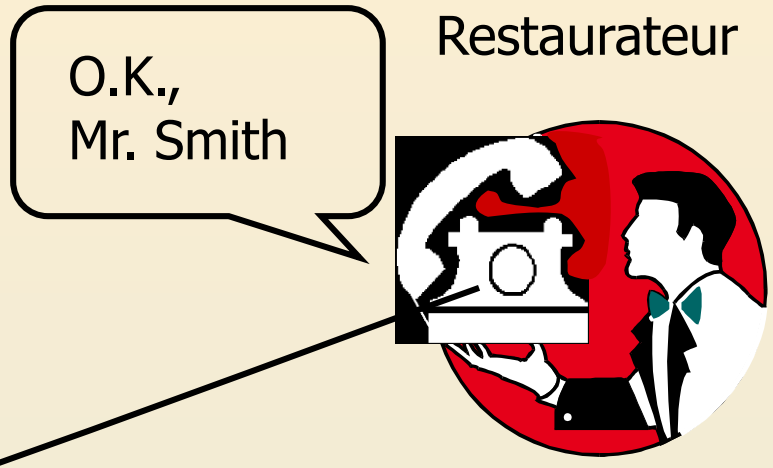
Results expected

- Some denial-of-service attacks can be executed with limited resources against a large, sophisticated site. This type of attack is sometimes called an "asymmetric attack“. For example, an attacker with an old PC and a slow modem may be able to disable much faster and more sophisticated machines or networks.

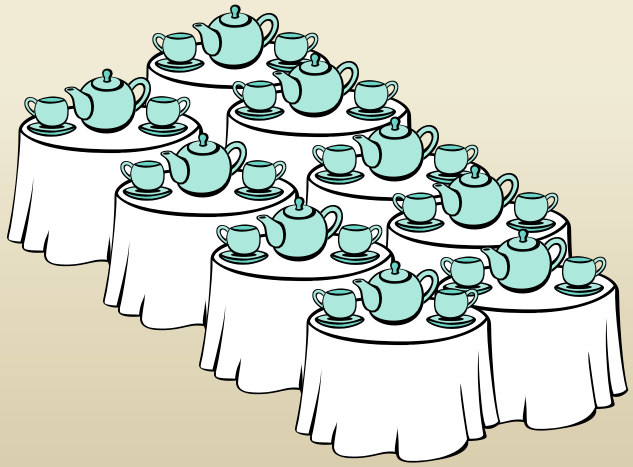


How to take down a restaurant?

Saboteur



Restaurateur



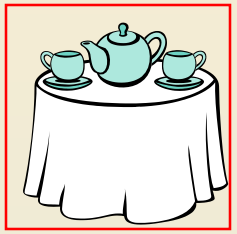
Saboteur vs. Restaurateur

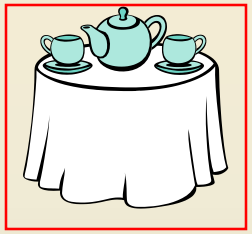
How to take down a restaurant?

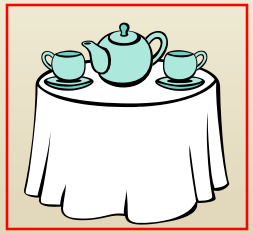
Restauranteur

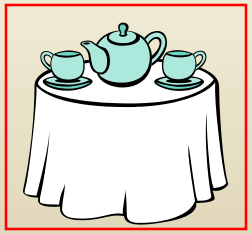
Saboteur

No More Tables!









Categories of DoS attack

- Flood attack - This is when a system gets too much internet traffic (people trying to connect to it). The traffic uses bandwidth and the internet servers slow down and eventually stop.
- Logic and software attacks - Internet packets are sent that are supposed to use bugs in the software or system. These attacks are easier to defend against because firewall or software patches usually correct the problem.

Categories of DoS attack

- Distributed Denial-of-Service attack - This type of attack uses either flood attacks or logic attacks, but it uses many different computers under the attacker's control (see Botnet). This type of attack is one of the most often used, and usually against company websites. This type of attack is often the hardest to prevent, track, and stop.

Samples

- Ping of Death
- Smurf & Fraggle
- Land attack
- Synchronous Flooding

Ping of Death

- With a Ping of Death attack, an echo packet is sent that is larger than the maximum allowed size of 65,536 bytes. The packet is broken down into smaller segments, but when it is reassembled, it is discovered to be too large for the receiving buffer. Subsequently, systems that are unable to handle such abnormalities either crash or reboot.
- You can perform a Ping of Death from within Linux by typing

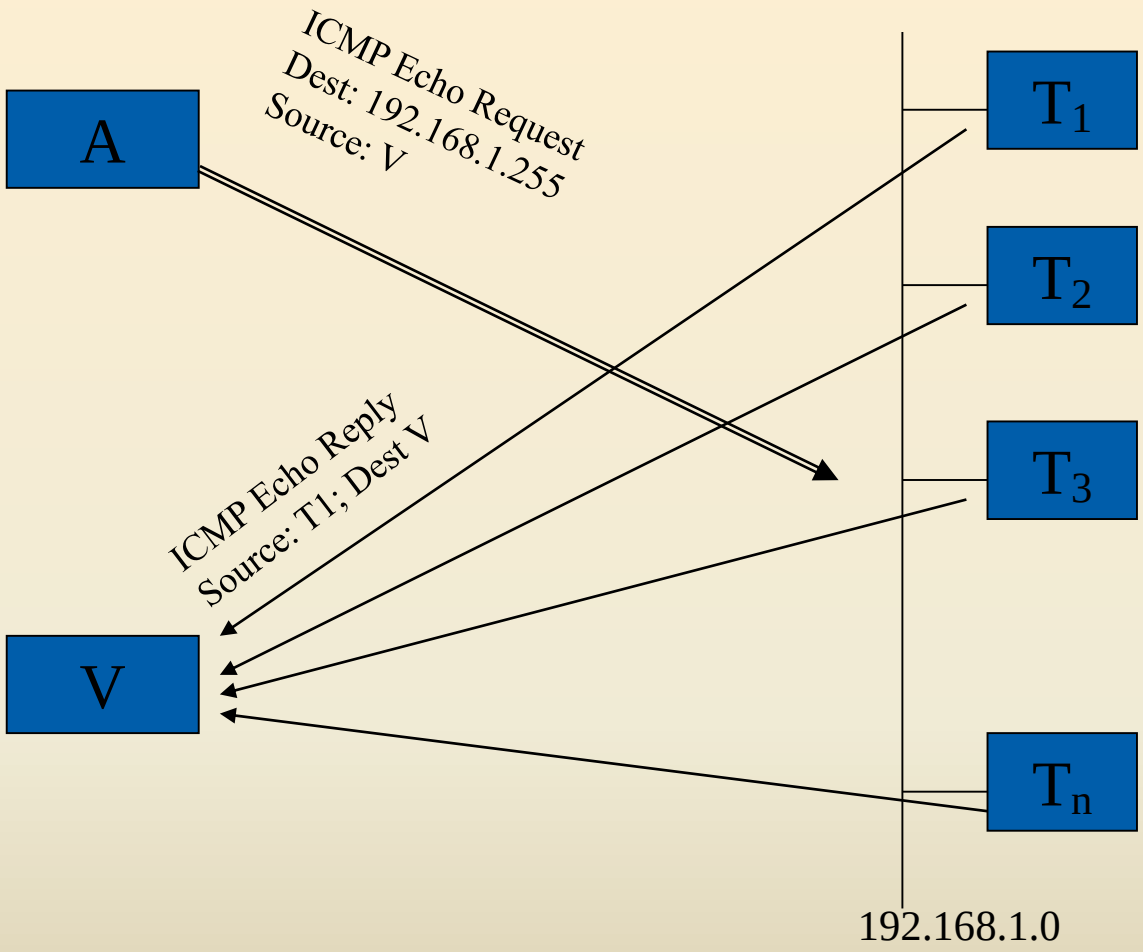
  ping –s 65537.

- Tools:
  - Jolt, Sping, ICMP Bug, IceNewk

Smurf

- A Smurf attack is another DoS attack that uses ICMP. Here, a request is sent to a network broadcast address with the target as the spoofed source. When hosts receive the echo request, they send an echo reply back to the target.
  - Sending multiple Smurf attacks directed at a single target in a distributed fashion might succeed in crashing it.

Smurf

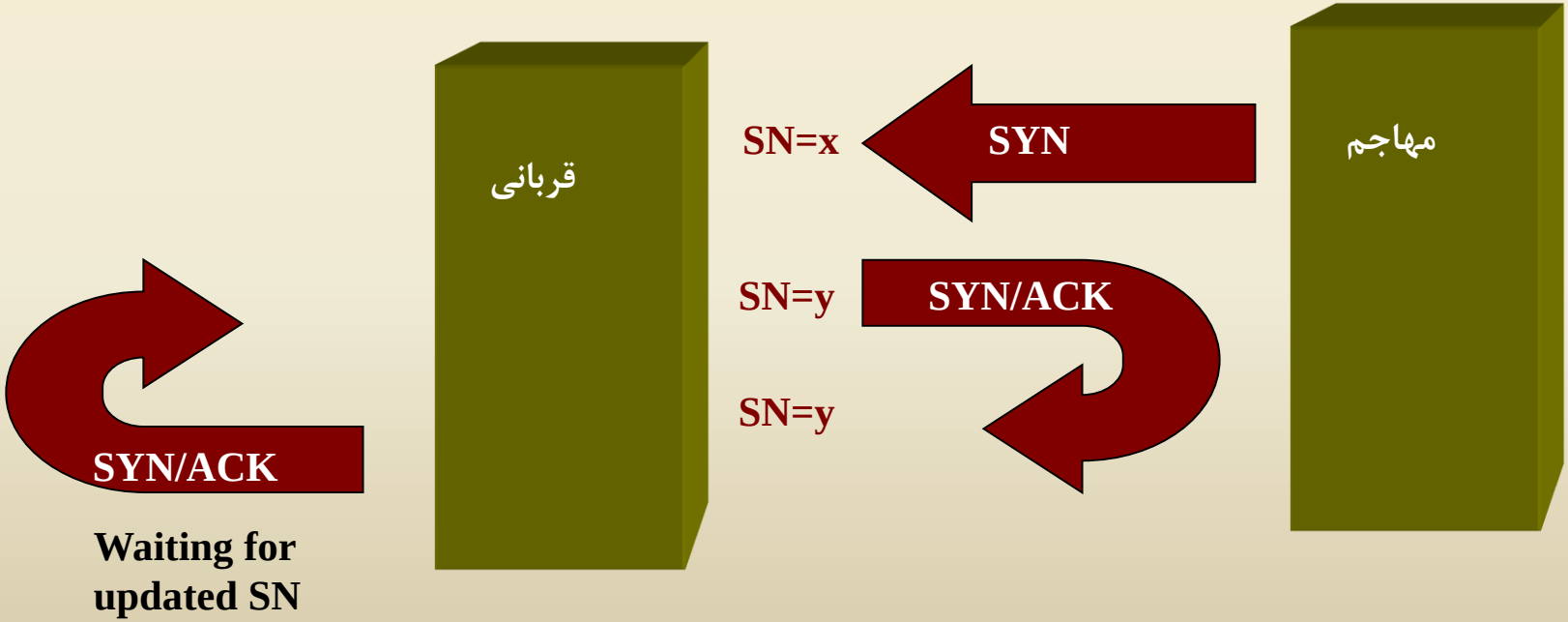

ICMP Echo Request
Dest: 192.168.1.255
Source: V
A
ICMP Echo Reply
Source: T1; Dest V
V
T1
T2
T3
Tn
192.168.1.0

LAND Attack

- In a LAND attack, a TCP SYN packet is sent with the same source and destination address and port number. When a host receives this abnormal traffic, it often either slows down or comes to a complete halt as it tries to initiate communication with itself in an infinite loop.
- Although this is an old attack (first reportedly discovered in 1997), both Windows XP with service pack 2 and Windows Server 2003 are vulnerable to this attack.
- HPing can be used to craft packets with the same spoofed source and destination address.

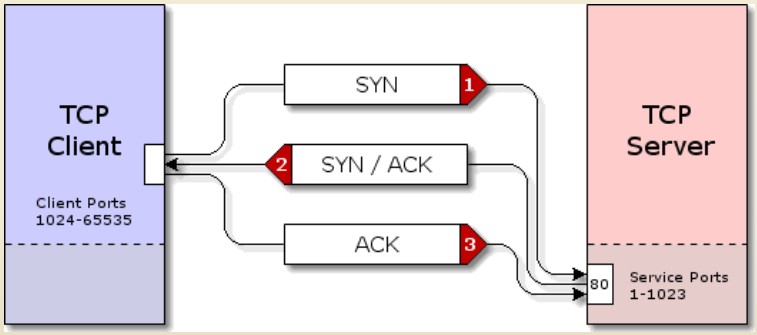
LAND Attack

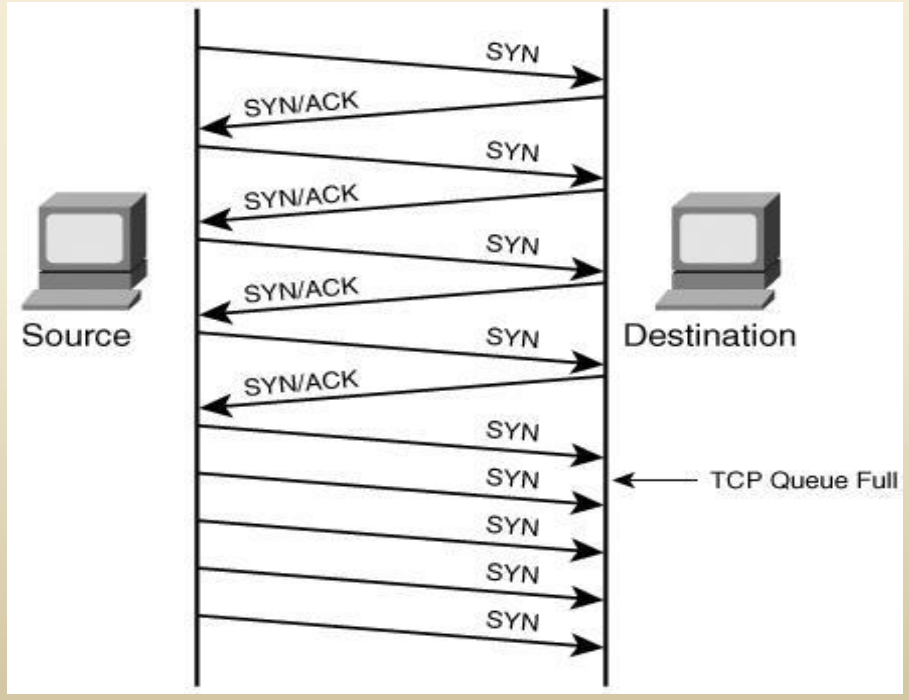
- هنگامی که قربانی SYN را دریافت می کند، شماره ترتیب را به روز کرده، ACK می فرستد، سپس بسته ای با شماره ترتیب مشابه دریافت می کند و آن را با همان شماره ترتیب برای فرستنده می فرستد تا توسط او اصلاح شود
- چون شماره ترتیب هرگز به روز نمی شود، قربانی دچار حلقه بی نهایت می شود!



Synchronous flood

• Attacker will send a flood of syn packet but will not respond with an ACK packet. The TCP/IP stack will wait a certain amount of time before dropping the connection, a syn flooding attack will therefore keep the syn_received connection queue of the target machine filled.





Synchronous flood

- SYN floods are still successful today for three reasons:

1) SYN packets are part of normal, everyday traffic, so it is difficult for devices to filter this type of attack.
2) SYN packets do not require a lot of bandwidth to launch an attack because they are relatively small.
3) SYN packets can be spoofed because no response needs to be given back to the target. As a result, you can choose random IP addresses to launch the attack, making filtering difficult for security administrators.

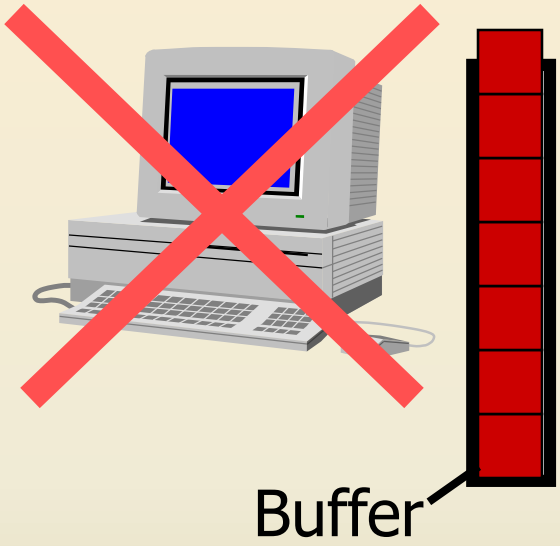
Return to our Restaurant

"TCP connection, please."

"O.K. Please send ack."

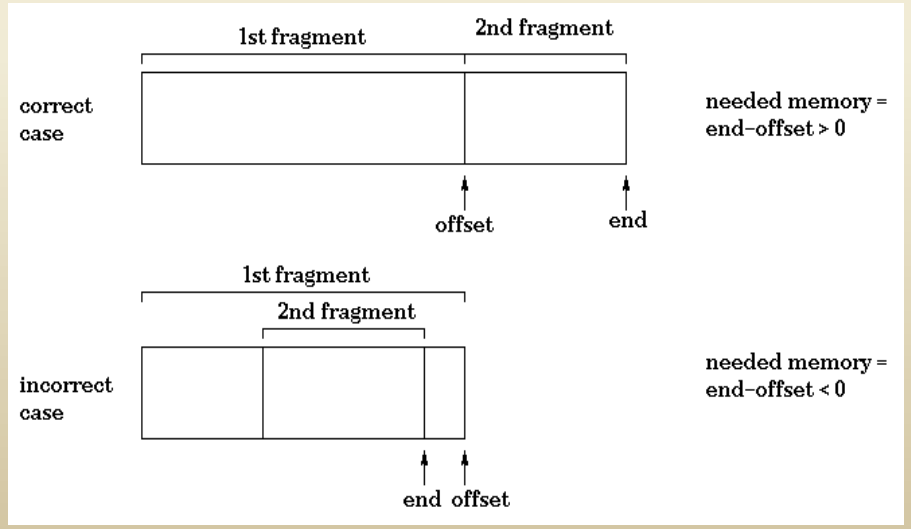
"TCP connection, please."

"O.K. Please send ack."



IP related attacks

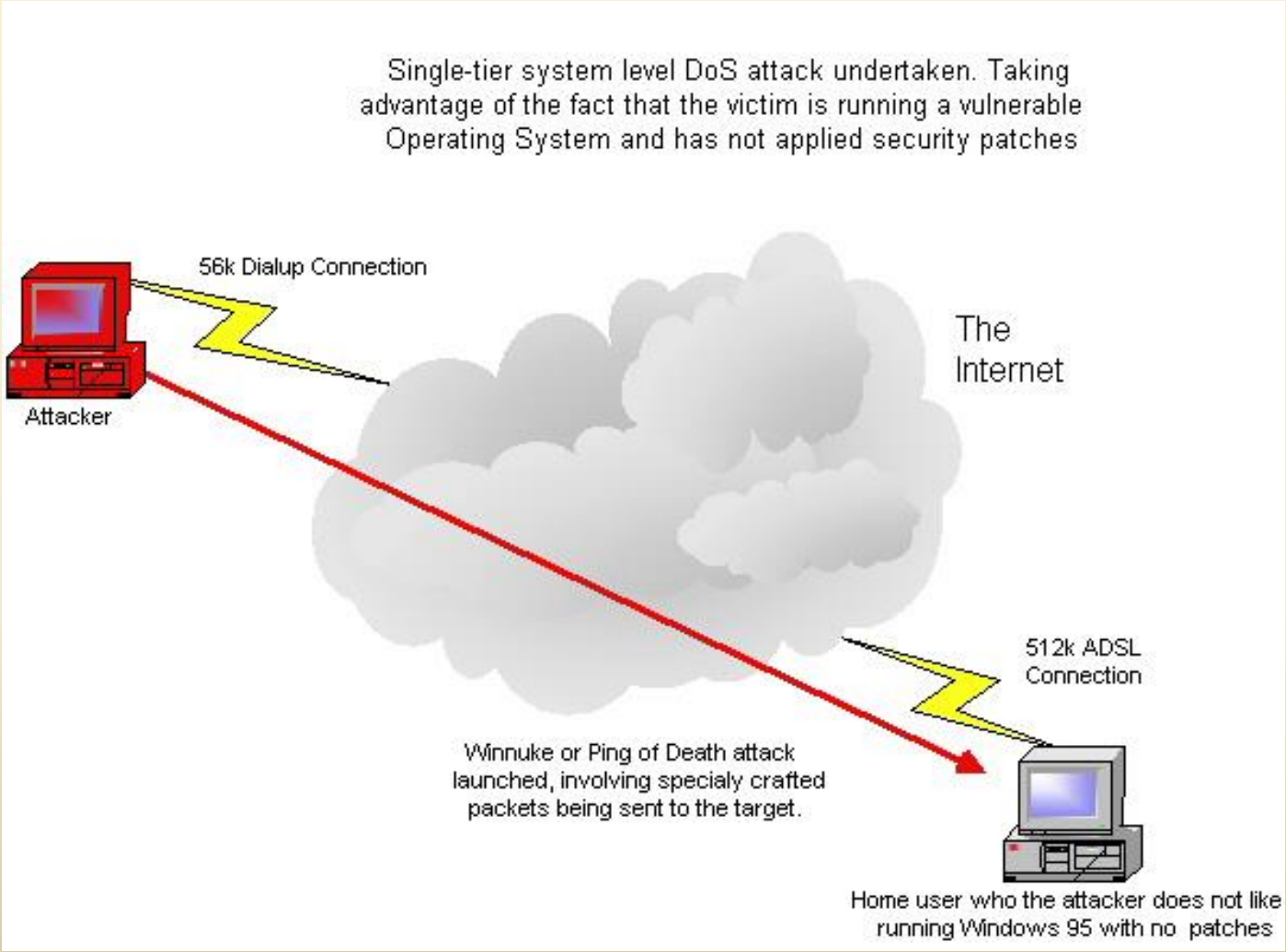
- **IP Packet options**
  - در این روش برخی از فیلد های انتخابی بسته به صورت تصادفی تغییر داده می شوند و بسته حاصل برای قربانی ارسال می شود مثلاً بیت های مربوط به کیفیت خدمات یک می شوند و لذا باعث بالا رفتن زمان پردازش **CPU** می شود
- **Tear drop**
  - در این حمله بسته ی **IP** در اثر یک افراز غلط، به قطعه هایی تقسیم می شود که همپوشانی دارند لذا قربانی نمی تواند این بسته را دوباره از قطعه هایش بسازد. این کار باعث می شود سیستم **Crash** کند.



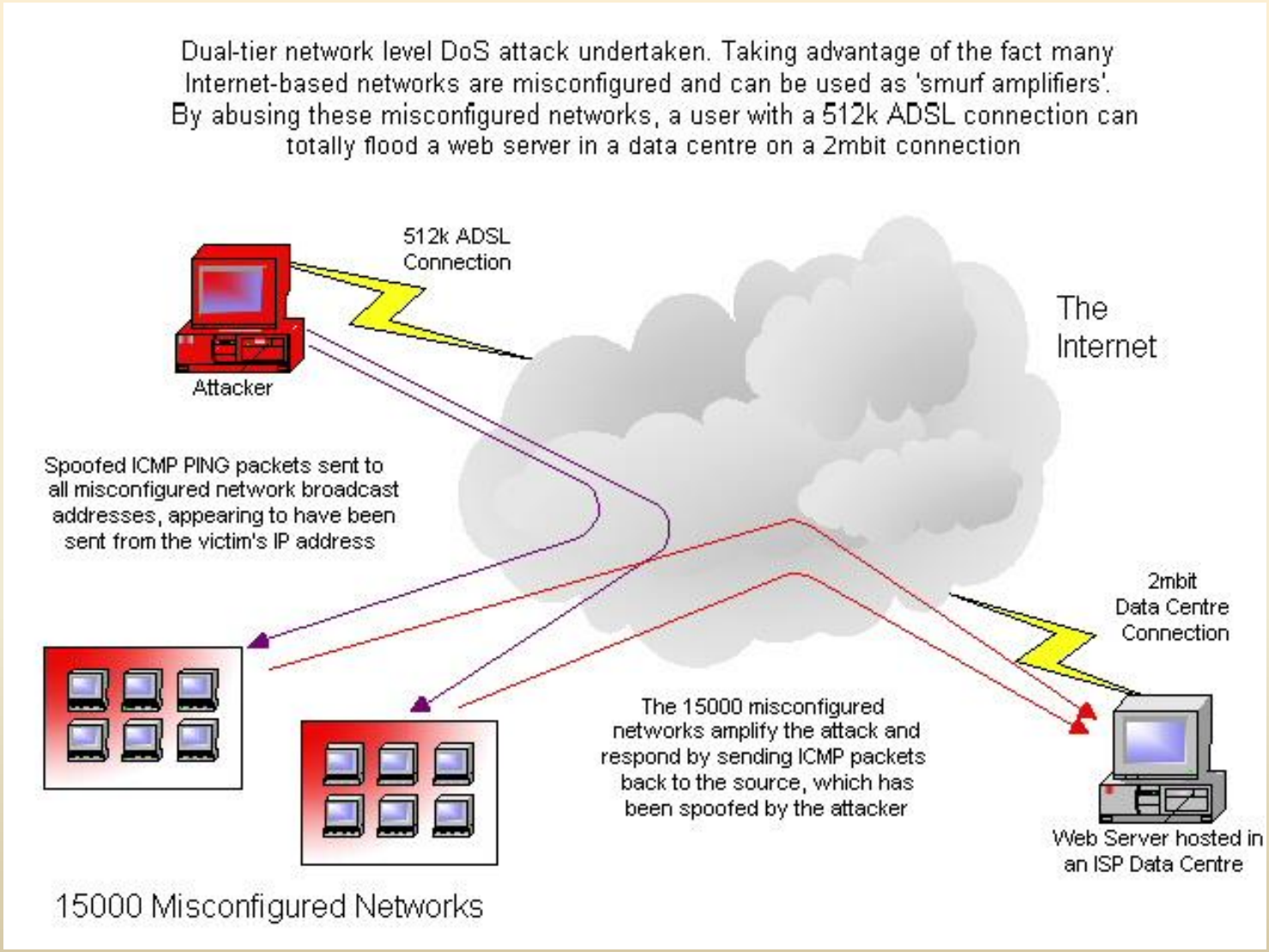
The Unnamed Attack

- attempts to cause a denial of service to the victim host, there is a gap created in the fragments.

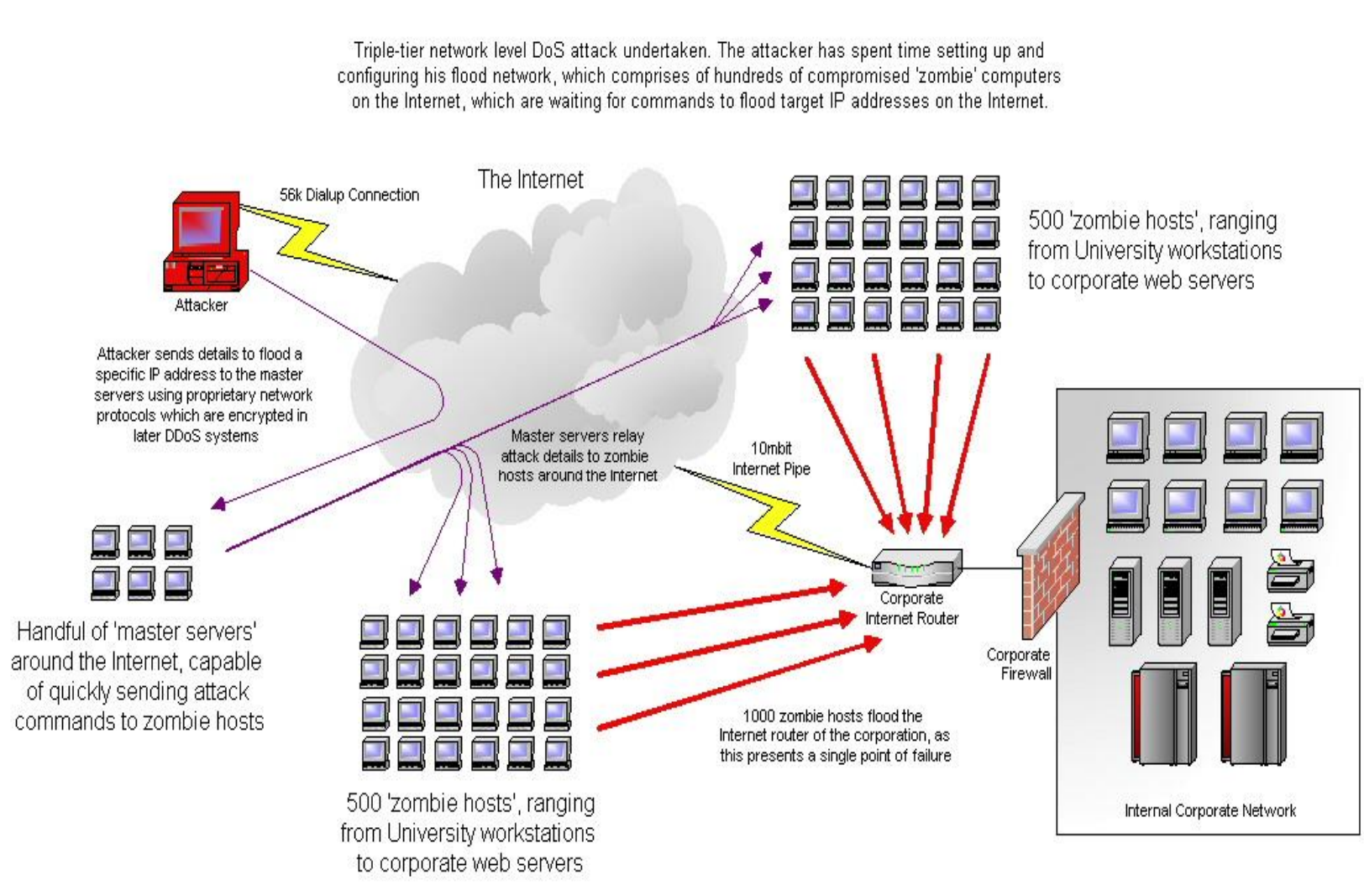
Single-tier DoS Attacks



Dual-tier DoS Attacks



Triple-tier DDoS Attacks



Attacker : Often a hacker with good networking and routing knowledge.

Master servers :Handful of back-doored machines running DDoS master software, controlling and keeping track of available zombie hosts.

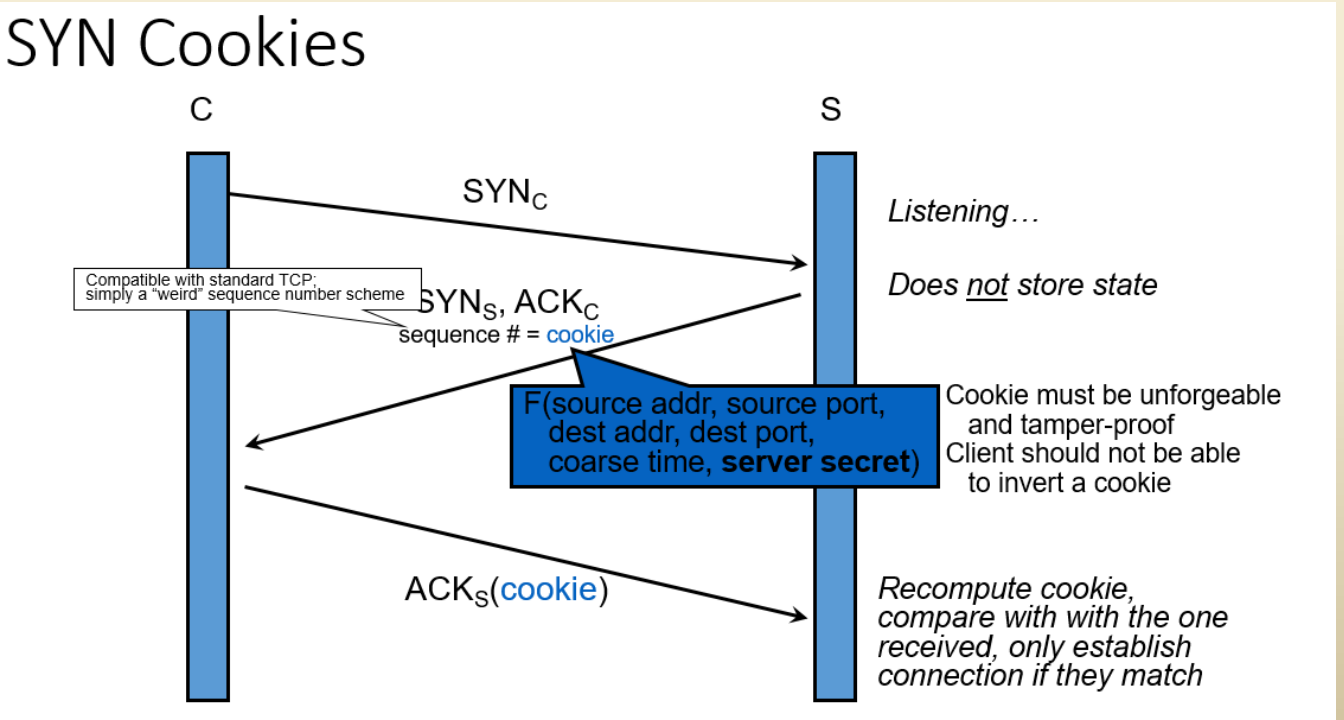
Zombie hosts : Thousands of back-doored hosts over the world

راه کارهای مقابله با حملات منع سروی‌س

- **مقابله با حملات منع سروی‌س لای‌ه کاربرد**
  - **عامل حمله**
    - **به روز نبودن سی‌ستم ها: وجود خطاهای‌ی در سروی‌س دهنده و سرو که موجب crash کردن سی‌ستم می شود**
    - **ارسال درخواستهای کوچک که منجر به پردازش زی‌اد می شود**
  - **راه کار مقابله**
    - **به روز رسانی متداوم**
    - **استفاده از مکانی‌زم های مبتنی بر reverse proxy WAF**
    - **حذف درخواستهای ربات محور**

راه کارهای مقابله با حملات منع سروی‌س

- **مقابله با حملات منع سروی‌س مبتنی بر پروتکل**
  - **عامل حمله**
    - **استفاده ار ی‌ک پروتکل شبکه به شکل نامتعارف آن**
  - **راه کار مقابله**
    - **اصلاح پروتکل**

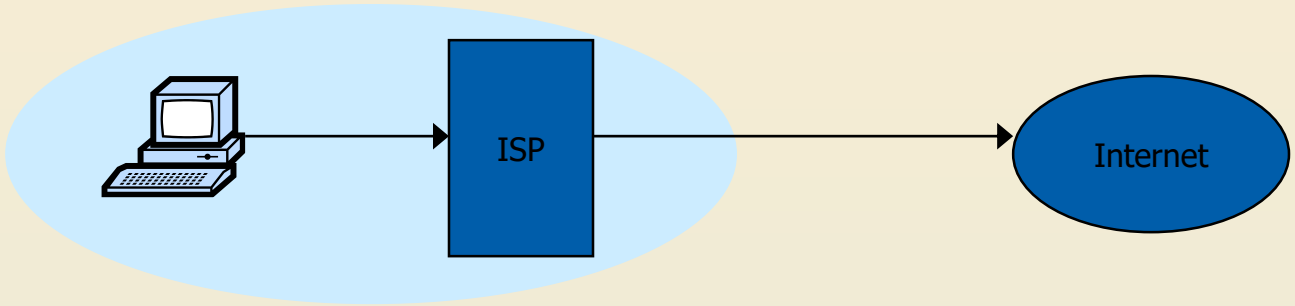


راه کارهای مقابله با حملات منع سروی‌س

- مقابله با حملات منع سروی‌س حجمی
  - عامل حمله
    - ارسال حجم زی‌ادی از بسته هابه شبکه
  - راه کار مقابله
    - متنوع اما برخی ناکارآمد

1. Ingress filtering

- Big problem: DDoS with spoofed source IPs
- Question: how to find packet origin?

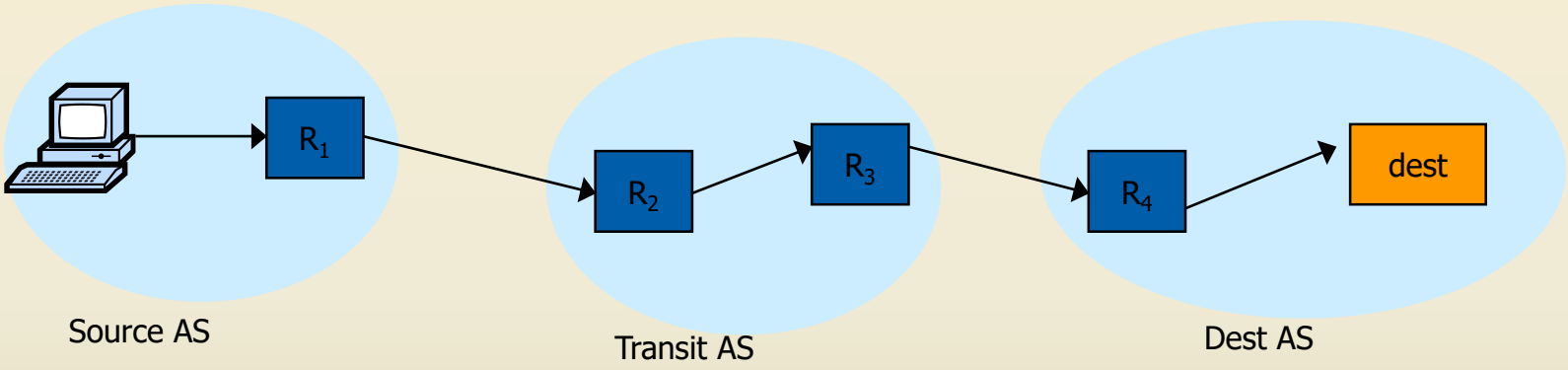


- Ingress filtering policy: ISP only forwards packets with legitimate source IP. (see also SAVE protocol)

Implementation problems

- ALL ISPs must do this. Requires global trust.
  - If 10% of ISPs do not implement $\Rightarrow$ no defense

- Another non-solution: enforce source IP at peer AS



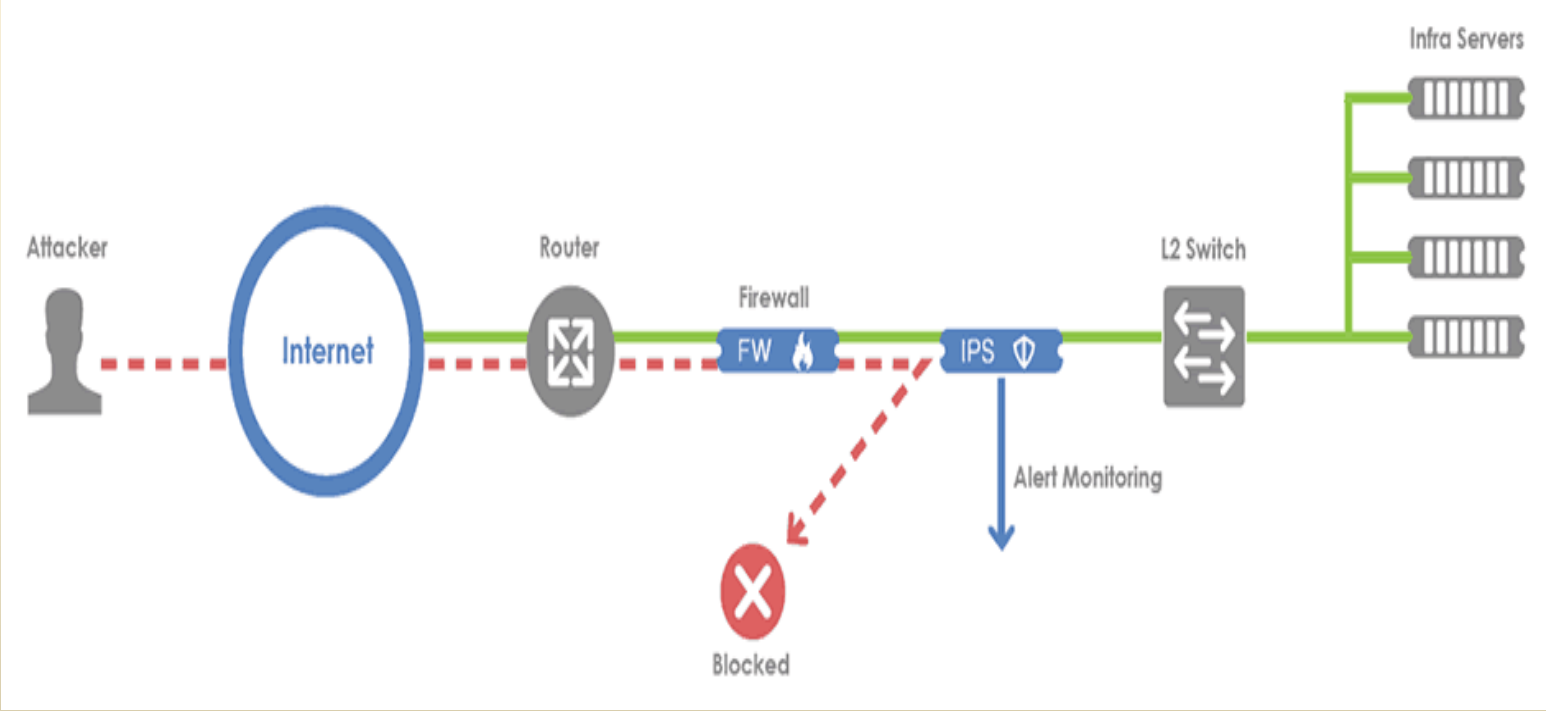
- Can transit AS validate packet source IP? No …

روی‌کردهای مطرح برای مقابله با حملات DoS

- **محدود کردن نرخ اتصال:**
  - معمولا حملات DoS اتصالات زی‌ادی را در حی‌ن حمله ای‌جاد می‌کنند، در حالی که در حالت نرمال تعداد کمی اتصال برقرار می‌شود.
  - محدود کننده‌های نرخ اتصال می‌تواند تعداد اتصالاتی که هر سروی‌س‌گی‌رنده می‌تواند برقرار کند را محدود کنند.

روی‌کردهای مطرح برای مقابله با حملات DoS

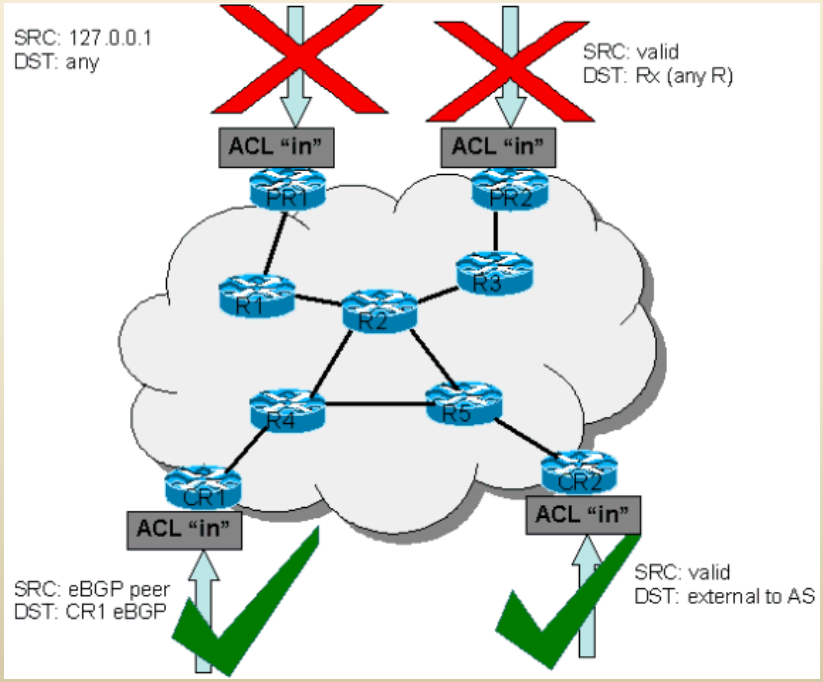
- **سی‌ستم‌های جلوگی‌ری از نفوذ (IPS):**
  - **ی‌ک IPS با تحلی‌ل کل ترافی‌ک ی‌ک سازمان می‌تواند در شناسـای‌ی و مقابلـه بـا حملات DDoSای که ی‌کی از سرورهای آن سازمان را هدف قرار داده‌اند، موثر باشـد. ای‌ن سی‌ستم قادر به شناسای‌ی امضای حملات منع‌سروی‌س متعددی بوده و سپس به صورت اتوماتی‌ک پاسخ مناسب تعی‌ی‌ن شده برای آن امضا را اجرا می‌کند.**



روی‌کردهای مطرح برای مقابله با حملات DoS

- **IPSها به خوبی سروی‌س‌دهنده‌های موجود در شبکه‌ی ی‌ک سازمان را از حملات محافظت می‌کند اما نمی‌توانند به صورت کامل اثرات بسی‌اری از حمله‌های DoS جدی‌د و پی‌چی‌ده‌ی امروزی را مرتفع کند، همچنی‌ن نمی‌توانند مانع اشباع شدن لی‌نک upstream سازمان شوند.**

- **ACLها:**
  - **در صورت ناخواسته تشخی‌ص داده شدن ترافی‌ک عبوری، آن را فی‌لتر می‌کنند.**



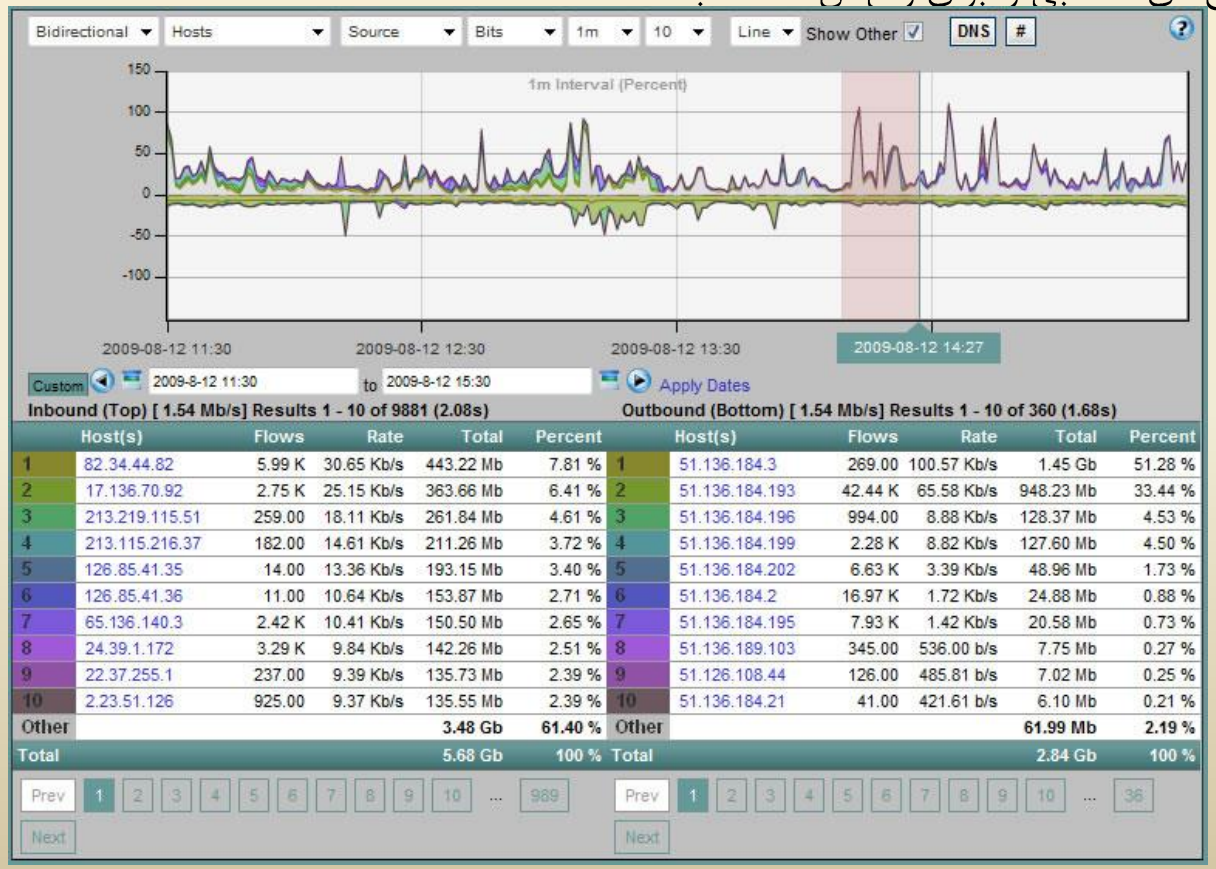
روی‌کردهای مطرح برای مقابله با حملات DoS

- هنگامی کـه حملـه‌ی DDoS از آدرس‌هـای جعلـی زی‌ـادی (ی‌ـک رنـج از آدرس) سرچشمه گرفته باشد، ACLها به تنهای‌ی قادر بـه جلـوگی‌ری از چنـی‌ن حملـه‌ی DDoS گسترده‌ای نی‌ستند (چرا که تعداد آدرس‌های موجود بسی‌ار زی‌ـاد خواهنـد بود)
- همچنی‌ن امکان تمای‌ز قای‌ل شدن بی‌ن ترافی‌ک قانونی و حمله نی‌ز وجـود نخواهـد داشت.

- **Blackholing (Null routing):**
  - می‌تواند به طور موثری همه‌ی ترافی‌کی را که دارای ی‌ک مبـدا خـاص و ی‌ـا مقصـد خاصی است را مسدود کند.
  - نمی‌تواند بی‌ن ترافی‌ک قانونی و حمله تمای‌زی قای‌ل شود، به عنوان مثال هنگامی که ترافی‌ک وب خروجی از ی‌ک مبدا ترافی‌ک نرمالی باشد اما ترافی‌ک DNS خروجـی آن حمله باشد.

روی‌کردهای مطرح برای مقابله با حملات DDoS

- **تشخی‌ص جری‌ان‌های ناهنجار با تحلی‌ل داده‌های netflow:**
  - معمولا مسی‌ری‌اب‌ها اطلاعات netflow را استخراج کرده و برای تحلی‌ل بی‌شتر به ی‌ک کلکتور خارجی ارسال می‌کنند. تحلی‌ل ای‌ن اطلاعات صادر شده به مدی‌ران شبکه کمک می‌کند تهدی‌دات روی داده را شناسـای‌ی و پاسخ‌های مناسبی را برای رفع آن‌ها انتخاب کنند.

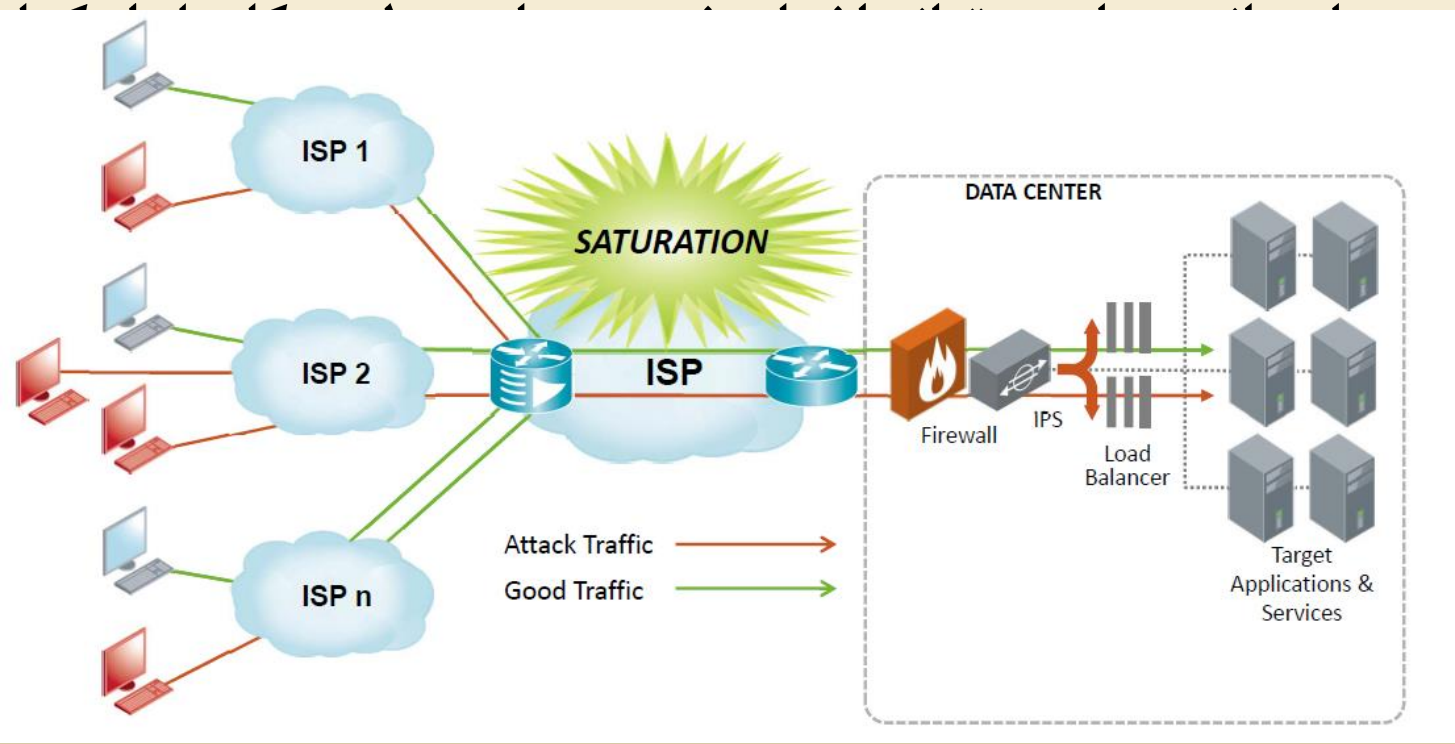


	Host(s)	Flows	Rate	Total	Percent
1	82.34.44.82	5.99 K	30.65 Kb/s	443.22 Mb	7.81 %
2	17.136.70.92	2.75 K	25.15 Kb/s	363.66 Mb	6.41 %
3	213.219.115.51	259.00	18.11 Kb/s	261.84 Mb	4.61 %
4	213.115.216.37	182.00	14.61 Kb/s	211.26 Mb	3.72 %
5	126.85.41.35	14.00	13.36 Kb/s	193.15 Mb	3.40 %
6	126.85.41.36	11.00	10.64 Kb/s	153.87 Mb	2.71 %
7	65.136.140.3	2.42 K	10.41 Kb/s	150.50 Mb	2.65 %
8	24.39.1.172	3.29 K	9.84 Kb/s	142.26 Mb	2.51 %
9	22.37.255.1	237.00	9.39 Kb/s	135.73 Mb	2.39 %
10	2.23.51.126	925.00	9.37 Kb/s	135.55 Mb	2.39 %
Other				3.48 Gb	61.40 %
Total				5.68 Gb	100 %

	Host(s)	Flows	Rate	Total	Percent
1	51.136.184.3	269.00	100.57 Kb/s	1.45 Gb	51.28 %
2	51.136.184.193	42.44 K	65.58 Kb/s	948.23 Mb	33.44 %
3	51.136.184.196	994.00	8.88 Kb/s	128.37 Mb	4.53 %
4	51.136.184.199	2.28 K	8.82 Kb/s	127.60 Mb	4.50 %
5	51.136.184.202	6.63 K	3.39 Kb/s	48.96 Mb	1.73 %
6	51.136.184.2	16.97 K	1.72 Kb/s	24.88 Mb	0.88 %
7	51.136.184.195	7.93 K	1.42 Kb/s	20.58 Mb	0.73 %
8	51.136.189.103	345.00	536.00 b/s	7.75 Mb	0.27 %
9	51.126.108.44	126.00	485.81 b/s	7.02 Mb	0.25 %
10	51.136.184.21	41.00	421.61 b/s	6.10 Mb	0.21 %
Other				61.99 Mb	2.19 %
Total				2.84 Gb	100 %

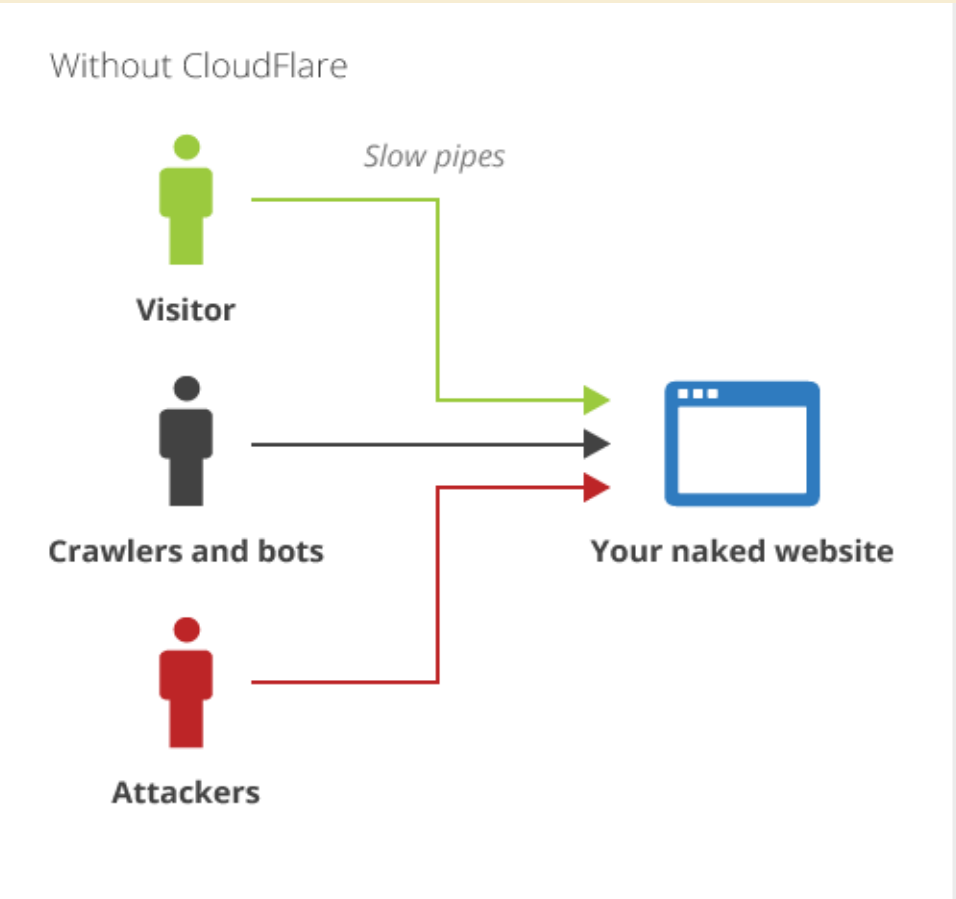
چالش اصلی جلوگی‌ری از حملات منع سروی‌س (DoS)

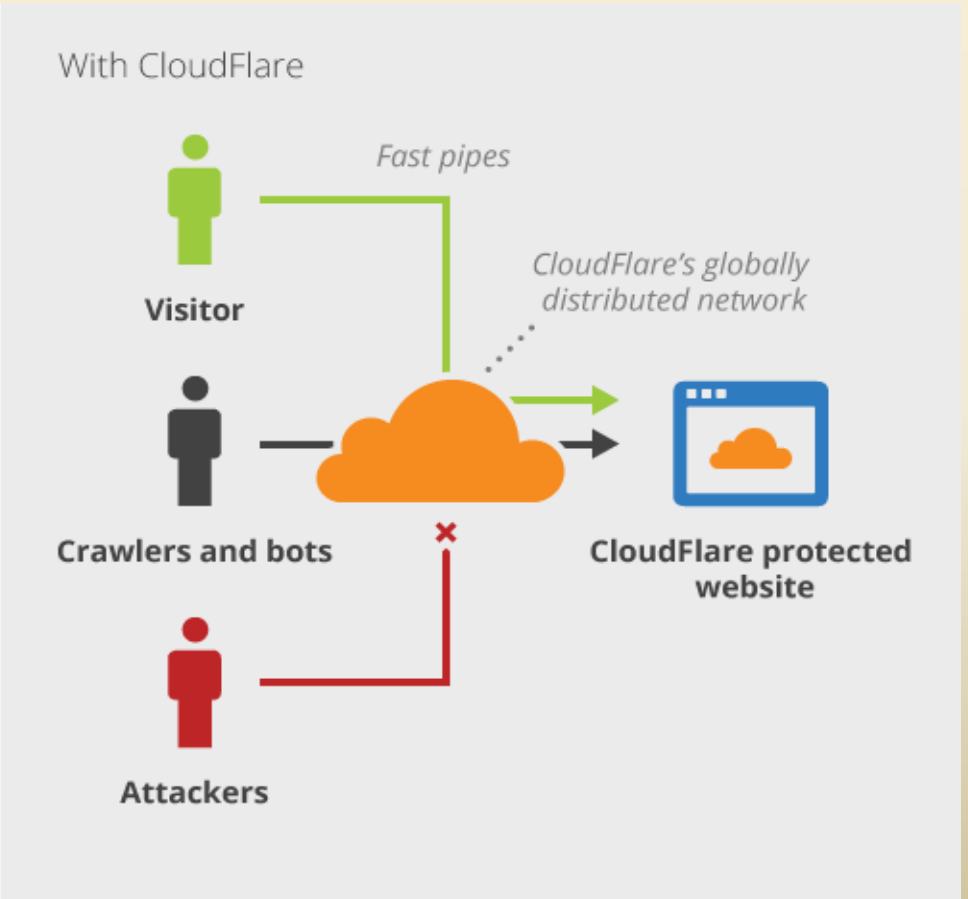
- **راه‌های سنتی جلوگی‌ری از حملات، مانند سی‌ستم‌های جلوگی‌ری از نفـوذ (IPS) به خوبی سرورهای مورد محافظت و زی‌رشبکه‌ها را در برابر حمـلات محافظت می‌کنند.**
- **با ای‌ن وجود ای‌ن روش‌ها از لی‌نک upstream که در حی‌ن ی‌ک حمله با بی‌فتـد،**



سامانه مقابله با حملات DDoS

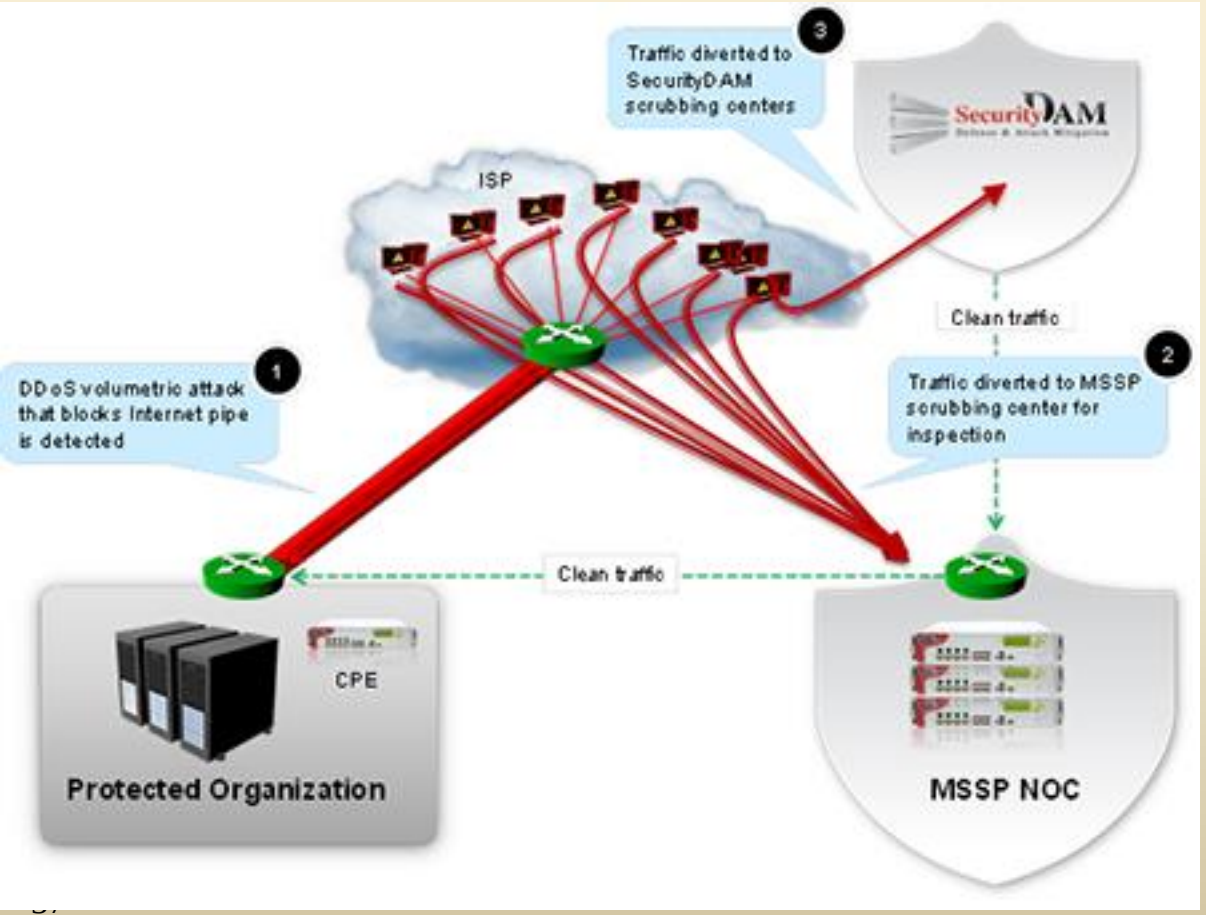
- روش مبتنی بر CDN
  - انتقال ترافی‌ک ی‌ک وب سای‌ت به سمت ی‌ک Cloud
  - مانند Cloudflar





سامانه مقابله با حملات DDoS

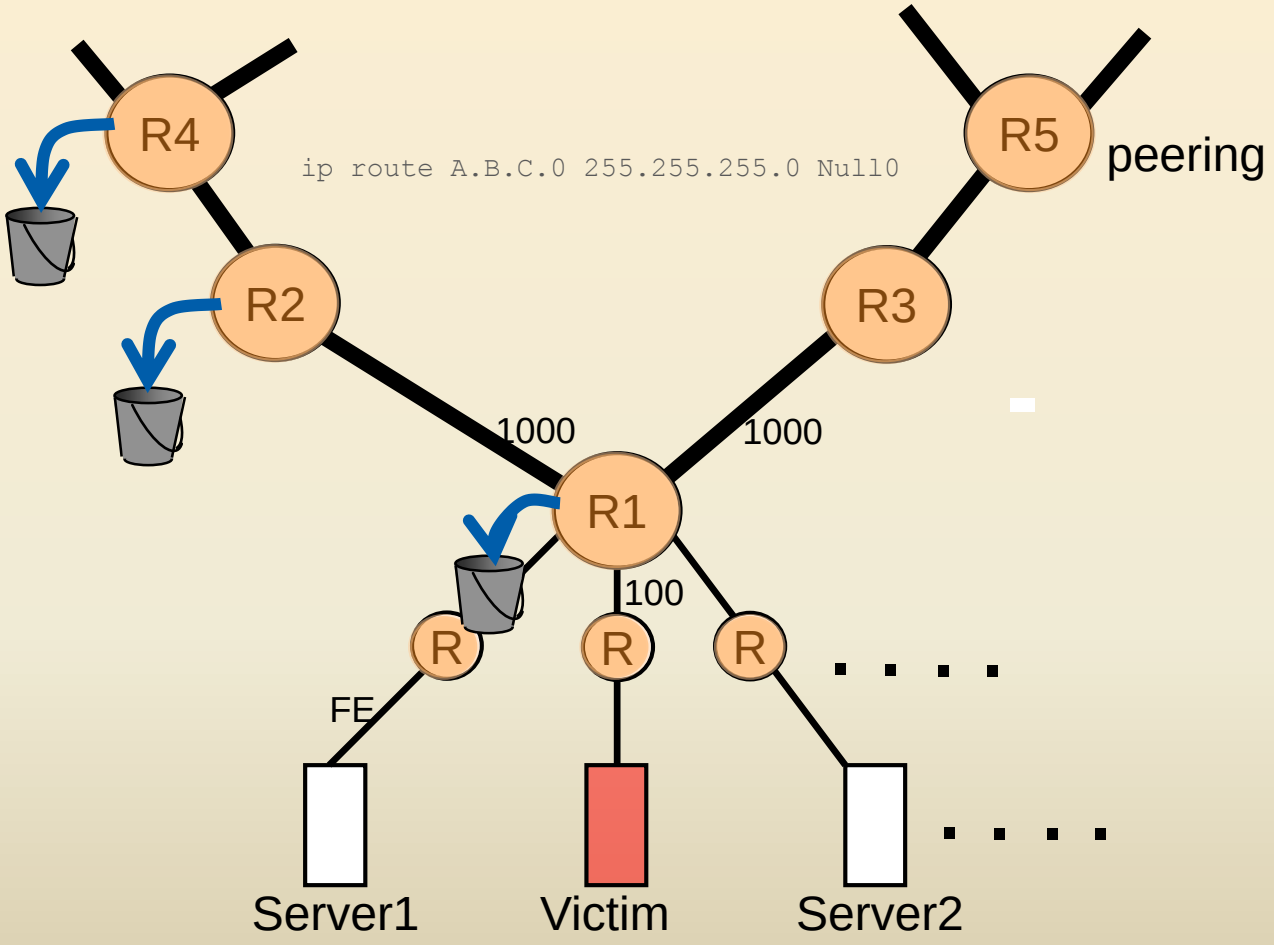
- DDoS Mitigation
  - Scrubbing Center –



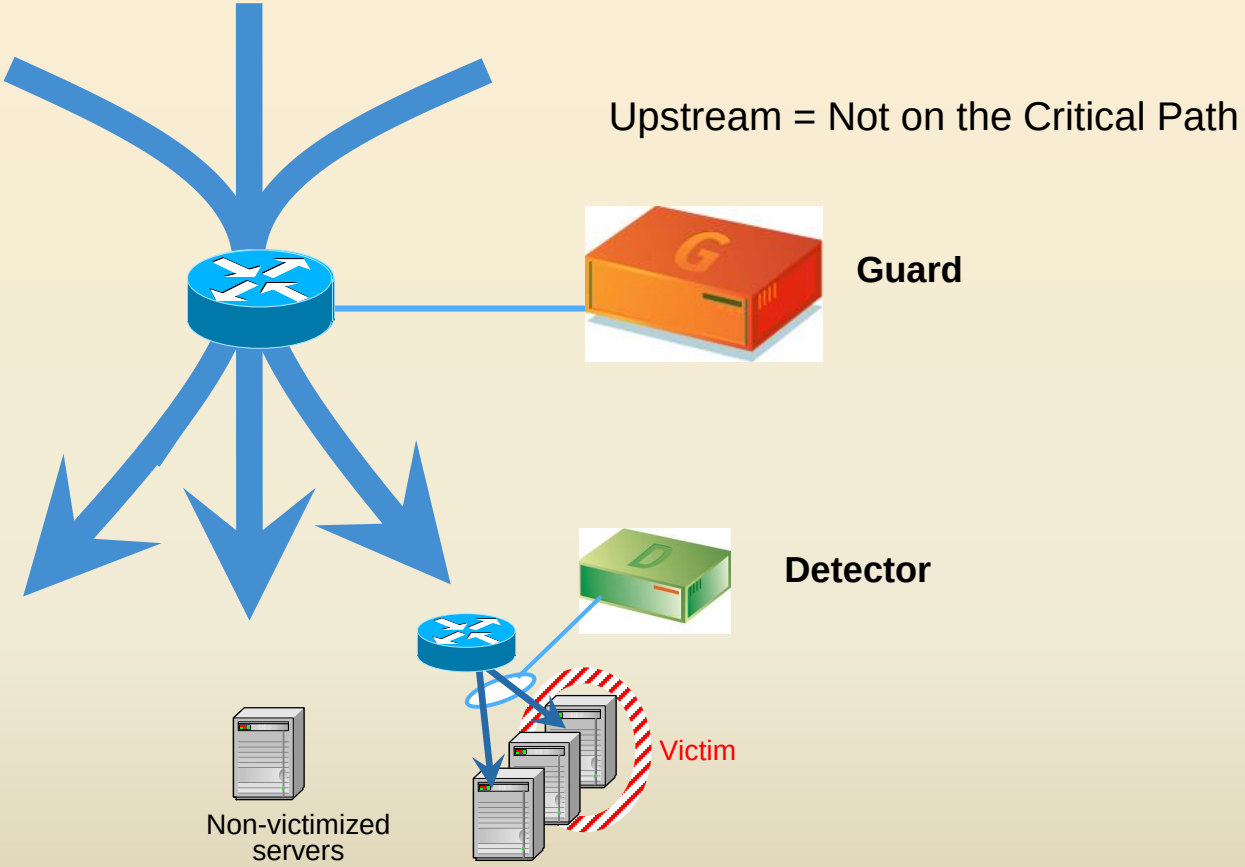
سامانه مقابله با حملات DDoS

- **DDoS Mitigation**
  - **Inline Method** –

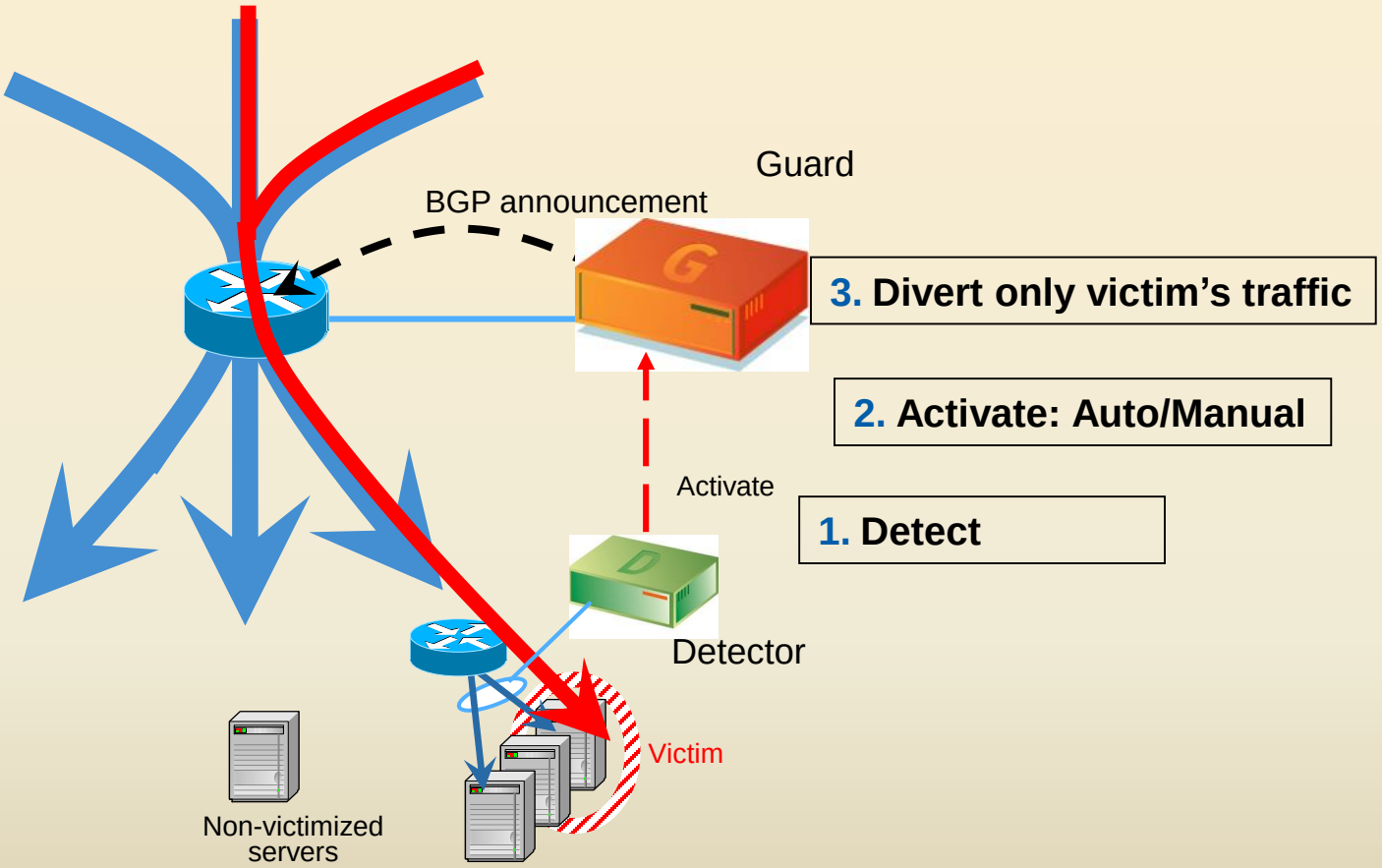
In line method DDoS Mitigation



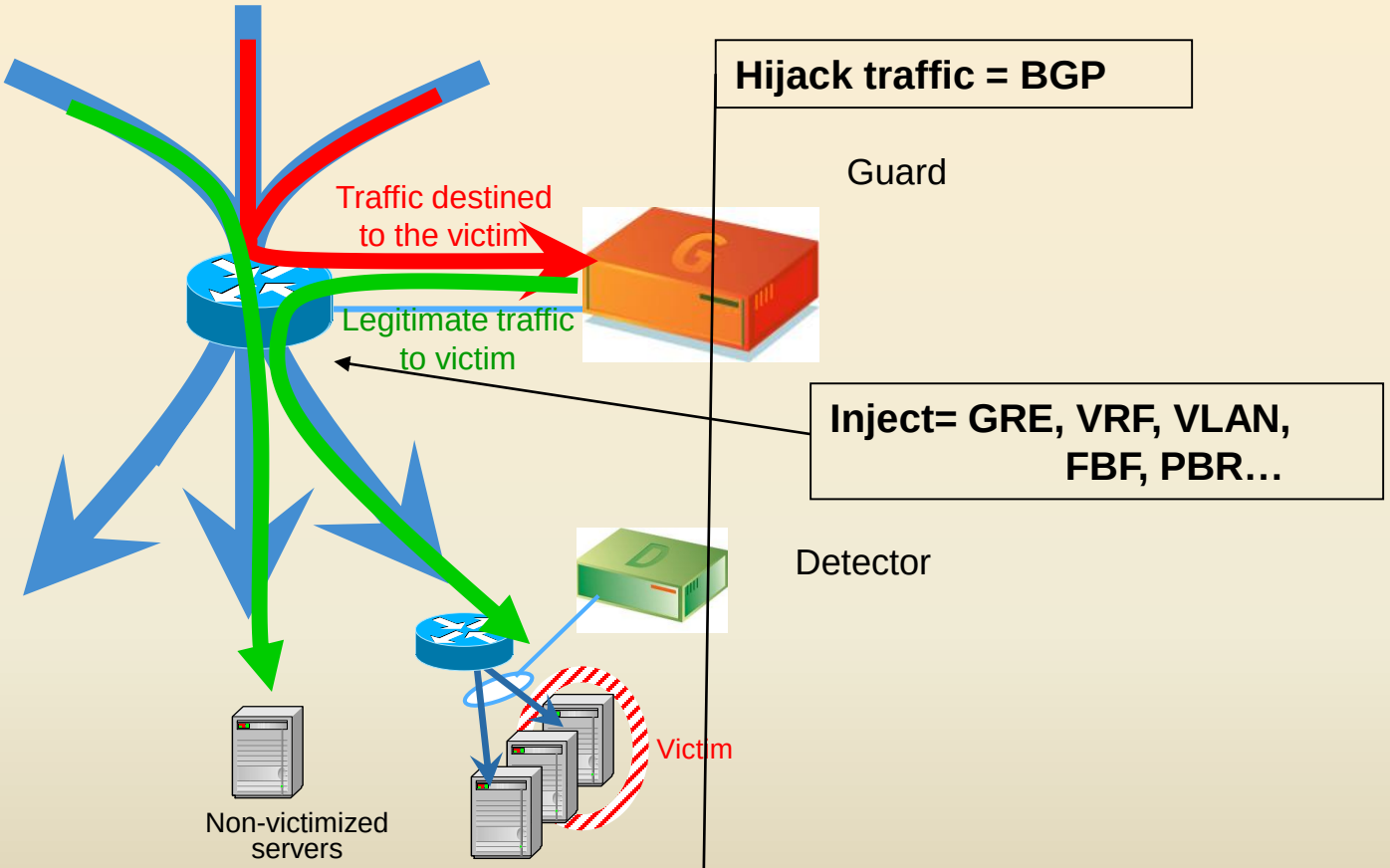
In line method DDoS Mitigation



In line method DDoS Mitigation



In line method DDoS Mitigation



Contents

- Denial of Service attacks
  - Concepts
  - Samples of attacks
- Malicious Logic attacks
  - Concepts
  - Viruses

Program Security

- Secure Programs: behave as expected
  - Unexpected behavior is a "program security flaw"
  - Happens because of an existing "vulnerability"

- IEEE Terminology

  Human error →

  Fault (incorrect code) →

  Failure (incorrect system behavior; external)

Patching

- One way of addressing faults: test, discover faults, patch them
- Problems:
  - No guarantee all faults are found
  - No guarantee the patch does not add another fault
    - Pressure leads to hurried patches
  - Because the entire system cannot be redesigned, there's a limit to how much a single patch can fix because it is constrained not to affect the rest of the system (for example, a definition of a variable that is passed on to several different modules, but creates a fault only in one)

Faults will always exist

- Human error
- Complexity of system
  - The study of security finds more possibilities for flaws while software engineering proceeds to find new software techniques
- Non-malicious and malicious faults

Malicious Logic

- Pfleeger definition: "*Hardware, software, or firmware capable of performing an unauthorized function on an information system.*"
- Bishop definition: "*a set of instructions that cause a site's policy to be violated*"
- Also known as malicious code or malware
- Unintentionally faulty code can cause the same/similar effects

Types of malicious logic

Trojan Horses

- Bishop definition: “a program with an overt effect (documented or known) and a covert effect (undocumented or unexpected)

Types of malicious logic

Virus

- Bishop definition: “a program that inserts itself into one or more files and then performs some (possibly null) action”
- Self replicating code, parasitic (attaches to “good” code)
- Can be
  - “resident” (attaches itself to memory and can execute after its host program is done) or
  - “transient” (active only while its host is executing)

Types of malicious logic – contd.

- Worms
  - Self replicating, spread through networks
  - Stand-alone, not attached to another piece of logic

- Logic Bombs
  - Bishop definition: “a program that performs an action that violates the security policy when some external event occurs”
  - Waits for a trigger condition
  - Time bomb!

Types of malicious logic – contd.

- Trapdoors/Backdoors
  - Alternative means of executing code
    - Intentional – legitimate and malicious purposes

- ActiveX, Java code
  - Execution of malicious code via Java applets, ActiveX scripts
  - Malicious mobile code

Types of malicious logic – contd.

- Bacteria
  - Virus or worm that “absorbs all of some class of resource”
  - For example: self-replicating piece of code fills up disk
- Hybrids
  - Usually a mixture of above

What we talk about now

- Virus (used as a generic term for malicious code)
  - Types of viruses
  - Means of attaching
  - Anatomy of a simple virus
  - More sophisticated virus
  - Virus detection methods
  - Antivirus mechanisms

Types of virus

- Classification by where they attach
  - Boot sector viruses
  - Parasitic viruses
  - Multipartite
    - Can infect either boot sectors or applications

- Classification by type of code
  - Binary viruses: usually written in assembly language then assembled to form executable image (binary file); attaches to other binary files or boot sector.
  - Macro viruses: written in high-level macro language then interpreted (possibly after pre-processing); attaches to other files that support same macro language

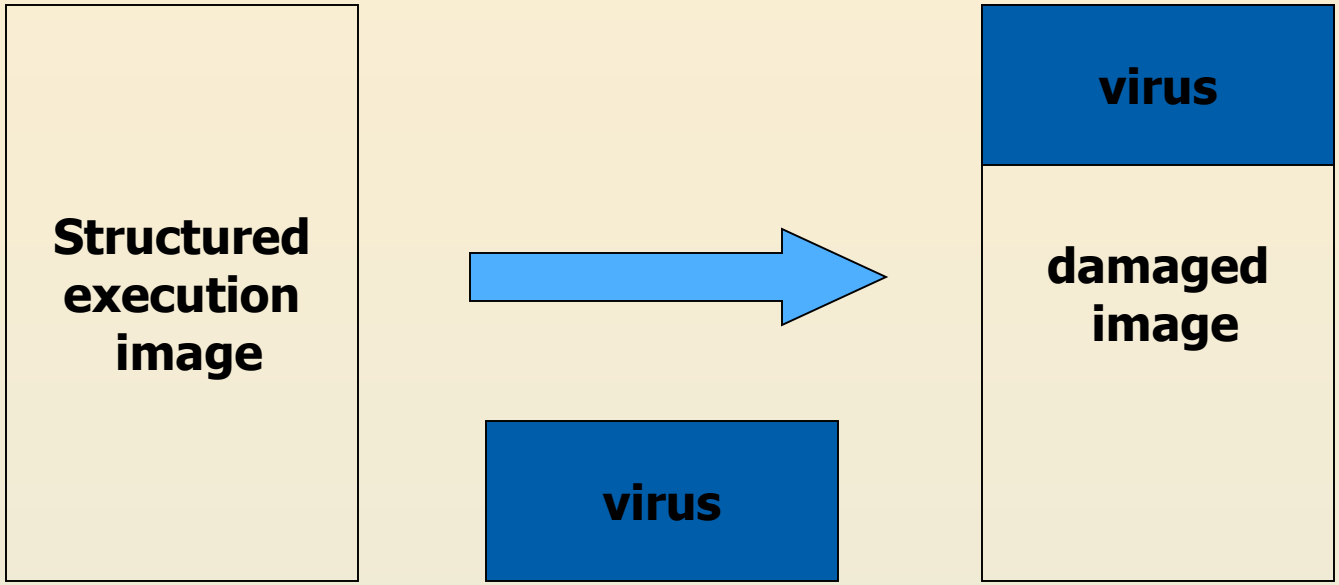
Types of viruses – contd.

- Polymorphic viruses
  - Mutate like biological viruses
- Stealth Viruses
  - Hard to detect
- TSRs (Terminate Stay Resident)
  - Memory resident viruses
  - Stay active in memory after application has terminated
- LKMs (Loadable Kernel Modules)
  - Future of Unix based viruses
- Encrypted viruses
  - Encrypts all virus code except a small decryption routine

Virus logic

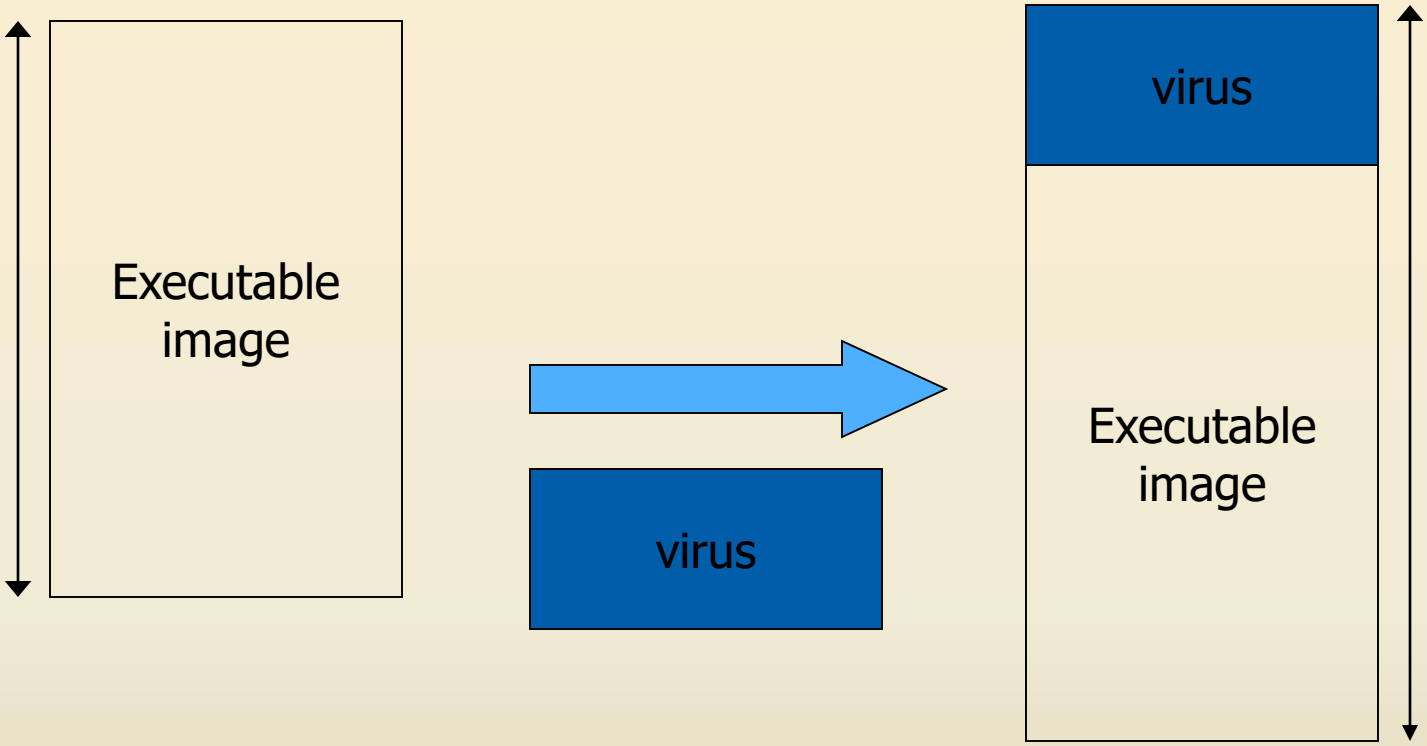
- Virus includes code to
  - Search for files to infect
  - Replicate
    - Make copy of self
    - Attach to file/boot sector
  - Reduce evidences of detection
    - Ideally, should execute quickly then pass control to infected program's normal code
    - Intercept system calls
    - Fool antiviral tools

Means of attaching: overwriting (virus *replaces* part of program)



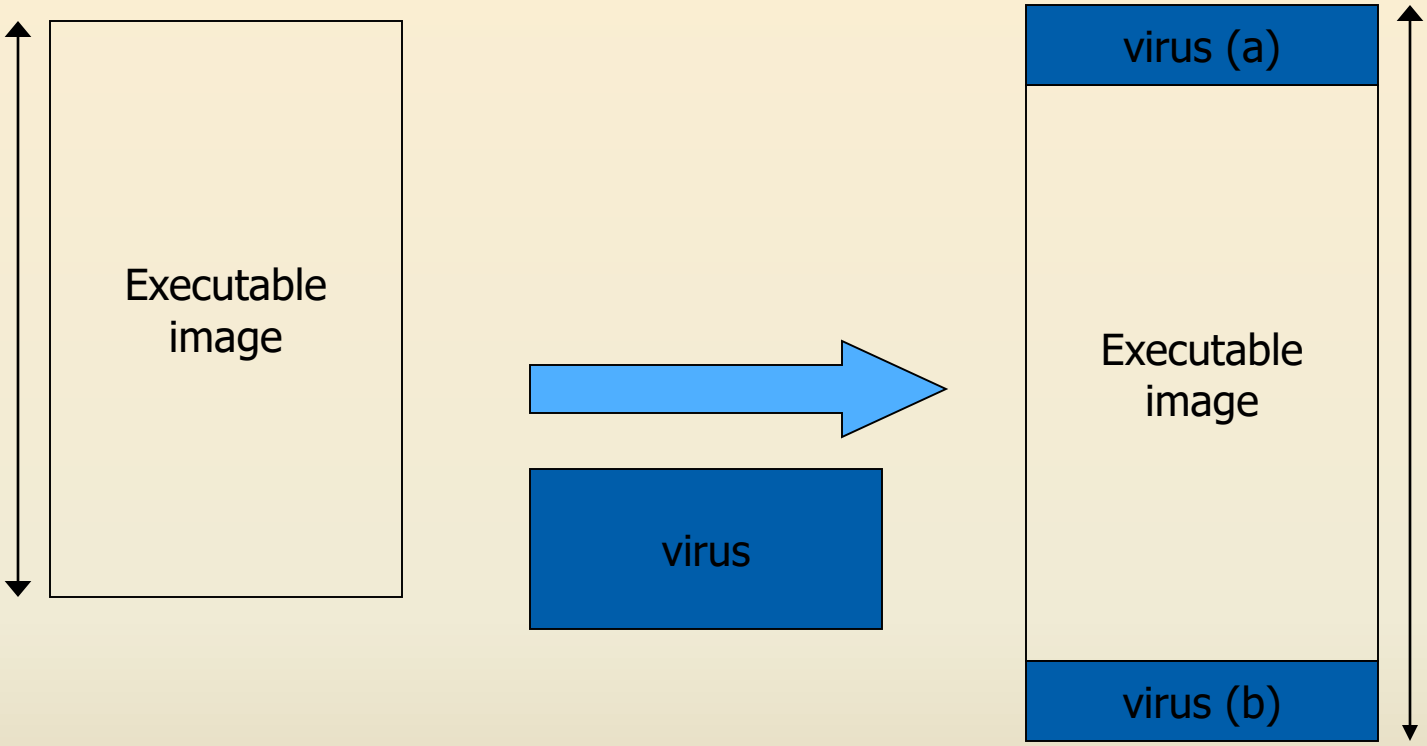
- Virus overwrites an executable file
- Easiest mechanism
- Since original program is damaged easily detected

Means of attaching: at the beginning (virus is *appended* to program)



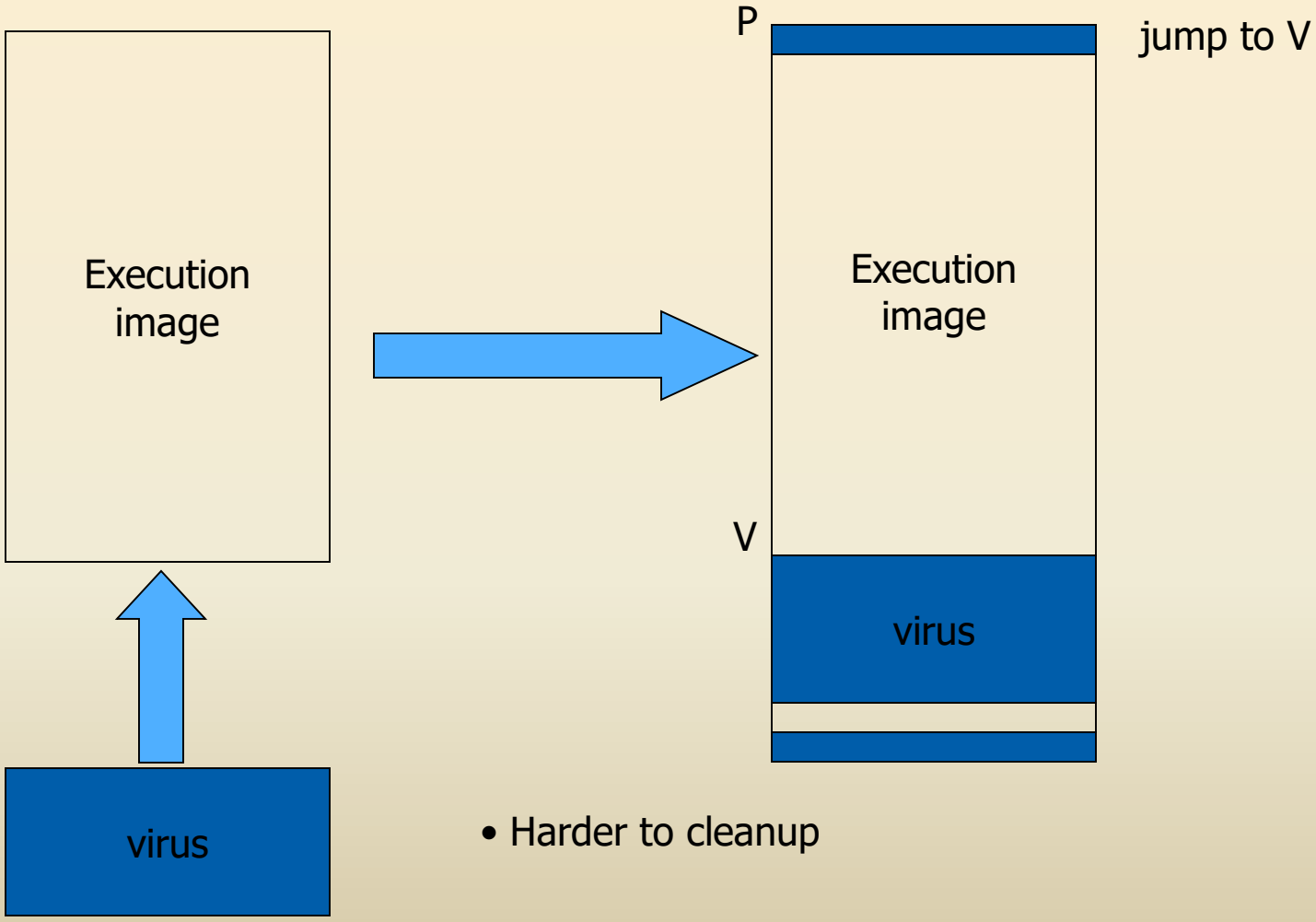
- Improved stealth because original program is intact
  - ✓ If original program is large, copying it may be slow
  - ✓ File size grows if multiple infections occur

Means of attaching: beginning and end (virus *surrounds* program)



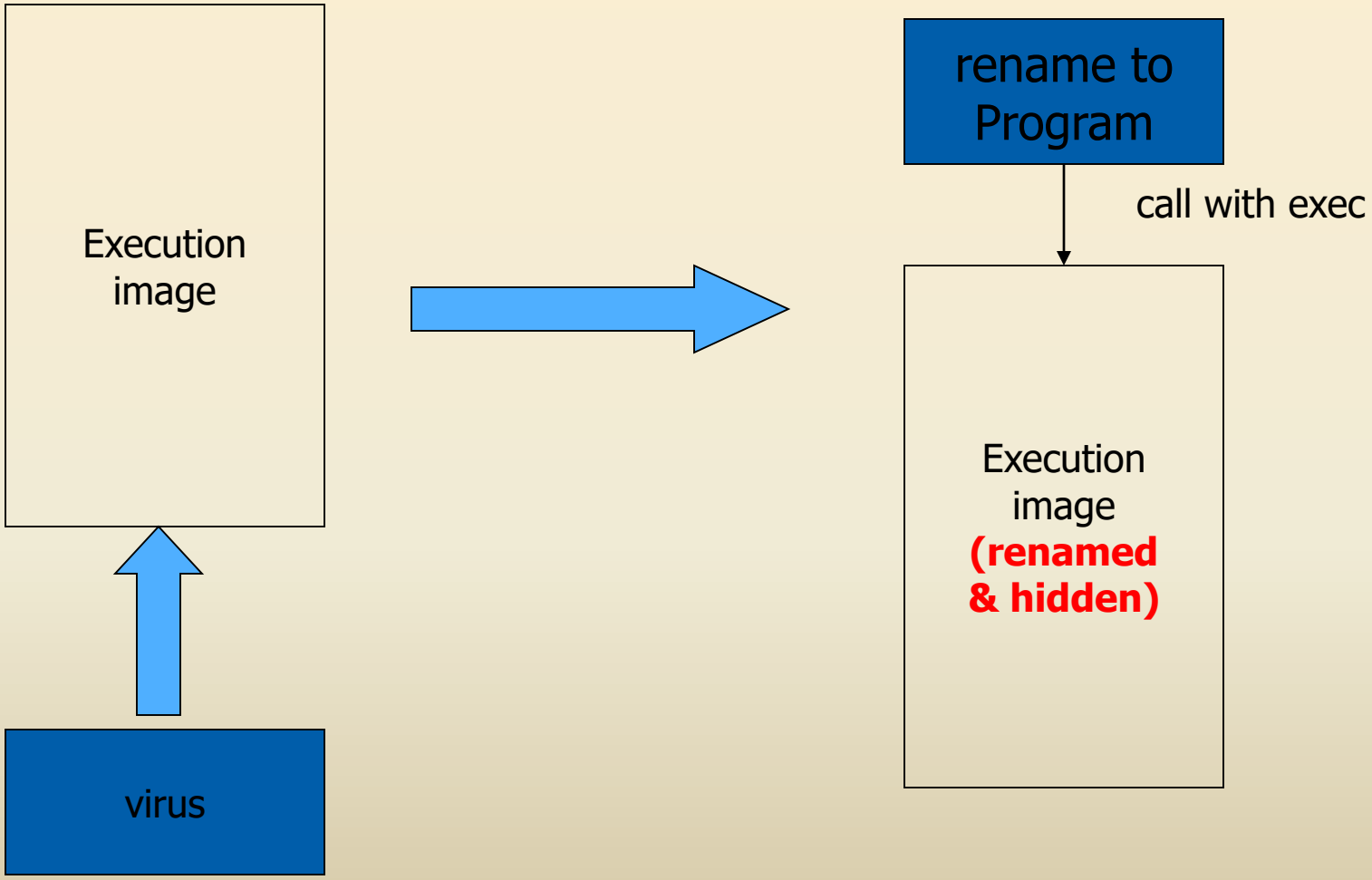
- Properties of appended virus
  - ✓Ability to clean up and avoid detection

Means of attaching: intersperse (virus is *integrated* into program)



- Harder to cleanup

Means of attaching: companions



Five major detection methods

- Integrity checking
  - Look for modified files by comparing old and new checksum
  - No software updates required
  - Requires maintenance of virus free checksums
  - Unable to detect stealth viruses
- Interrupt monitoring
  - Attempts to locate and prevent a viruses' interrupt calls
  - Poor system utilization
- Memory detection
  - Depends on recognition of known viruses' location and code in memory

Five major detection methods

- Signature scanning
  - Recognizes viruses' unique "signature": a pre-identified hex
  - Need to maintain current signature files and scanning engine refinements
  - False positives
- Heuristics/Rule based
  - Uses a set of rules to effectively parse through files and identify code
  - Uses expert systems or neural networks
  - Depends on current rule-set

(Detection can be performed on-access or on-demand)

Properties of a good signature

- Should always appear in the virus, so there won't be any false negatives
- Should not appear in (m)any other files, so there won't be (m)any false positives
- Should be reasonably short, for efficient scanning
- For simple viruses, it's easy to find good signatures but for complicated ones …!

Polymorphic Viruses

- Polymorphic = “many forms”
- Goal: Foil virus scanners by changing virus code each time virus replicates, so that it will be difficult to find a good signature
- Approaches:
  - Encrypt virus with random key
    - Note: Goals and techniques are different than in the encryption techniques we studied earlier. XOR with stored key is sufficient.
  - “Mutate” virus by making small changes that don’t affect the semantics of the code
  - Nearly 2 billion similar codes can be evolved from a single code
  - Requires algorithm based matching instead of simple string based matching

Replication of encrypted virus

- Copy decryption engine to infected file
- Select new key and copy it to the infected file
- For each byte of the encrypted portion of the virus:
  - take decrypted byte
  - encrypt it with the new key
  - copy it to the infected file
- Result: different replicas of virus have different byte patterns, so difficult to find signature

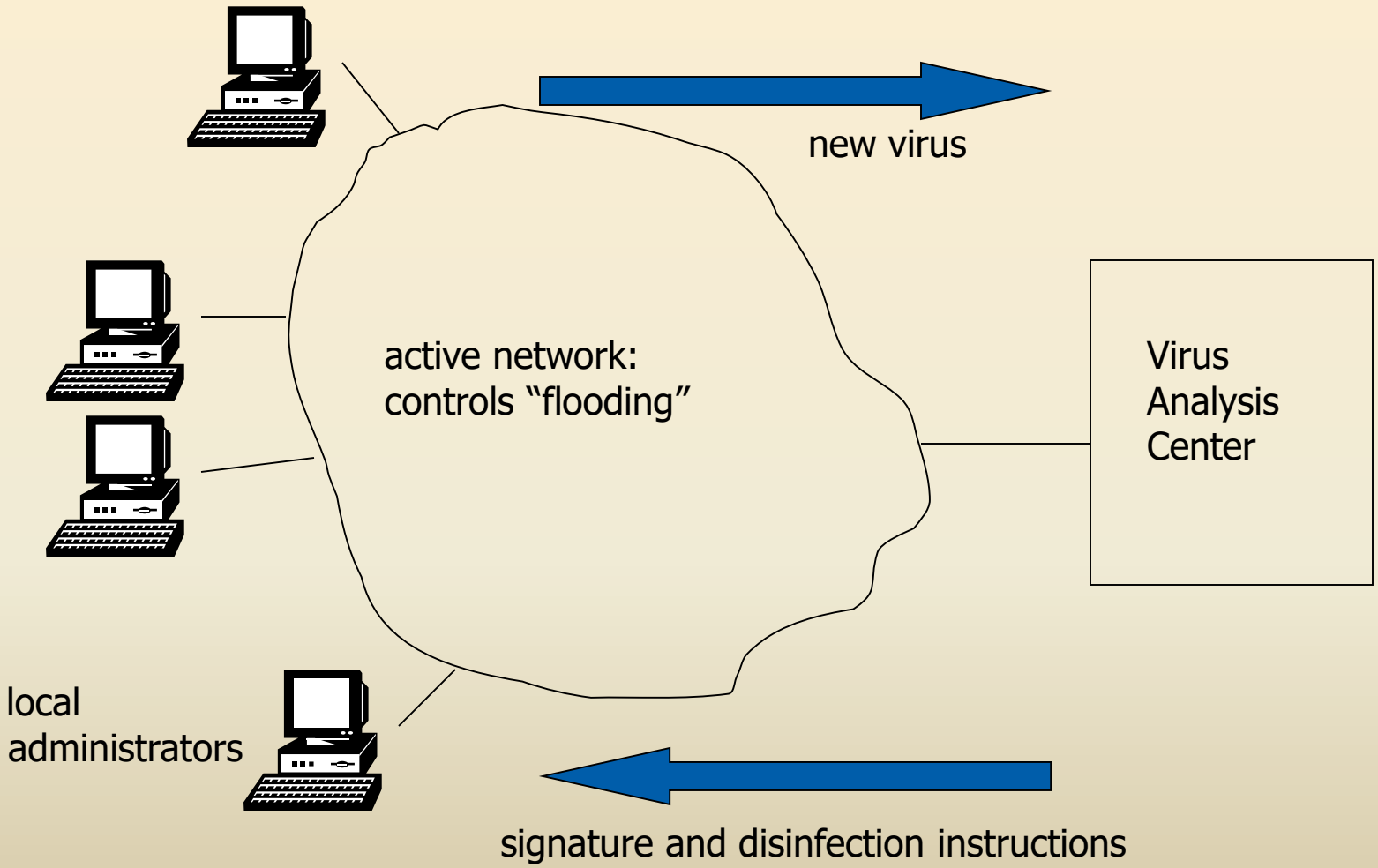
Anti-virus tools' answer to encryption

- Select the signature from the unencrypted portion of the code, i.e. the decryption engine
- Problems:
  - Anti-virus tools usually want to determine which virus is present, not just determine that some virus is present (in order to "disinfect").
    - Can emulate the decryption then further analyze the decrypted code.
  - virus writers have responded by obscuring the encryption engine through mutations
- It's a game of cat and mouse!

Virus Analysis

- Analysis of virus by human expert
  - slow: by the time signature has been extracted, posted to AV tool database, downloaded to users, virus may have spread widely.
    - pre-1995: 6 months to a year for virus to spread world-wide
    - now: days or hours
  - labor-intensive: too many new viruses
    - currently, 8-10 new viruses per day
  - can't handle epidemics:
    - queue of viruses to be analyzed overflows
- Automated analysis, e.g. "Immune System"
    - developed at IBM Research
    - licensed to Symantec

Immune System Architecture



Signature Extraction at VAC

- Virus allowed (encouraged) to replicate in controlled environment in immune center
- This yields collection of infected files
- In addition, a collection of “clean” files is available
- Machine learning techniques used to find strings that appear in most infected files and in few clean files (e.g. award/punishment learning):
  - search files for candidate strings
    - add points if found in infected file
    - subtract points if found in clean file

Macro-viruses

- Written in macro-language
- Infect documents (as opposed to programs), such as word-processor docs, etc.
- “Attach” by modifying commonly used macros, or start-up macros
  - popular target is Normal.dot, which is opened when MS Office applications are executed
- Spread when documents are transmitted, via disks, file transfer, e-mail attachments, ...
- Macro virus dependencies:
  - Application popularity
  - Macro language depth
  - Macro implementation