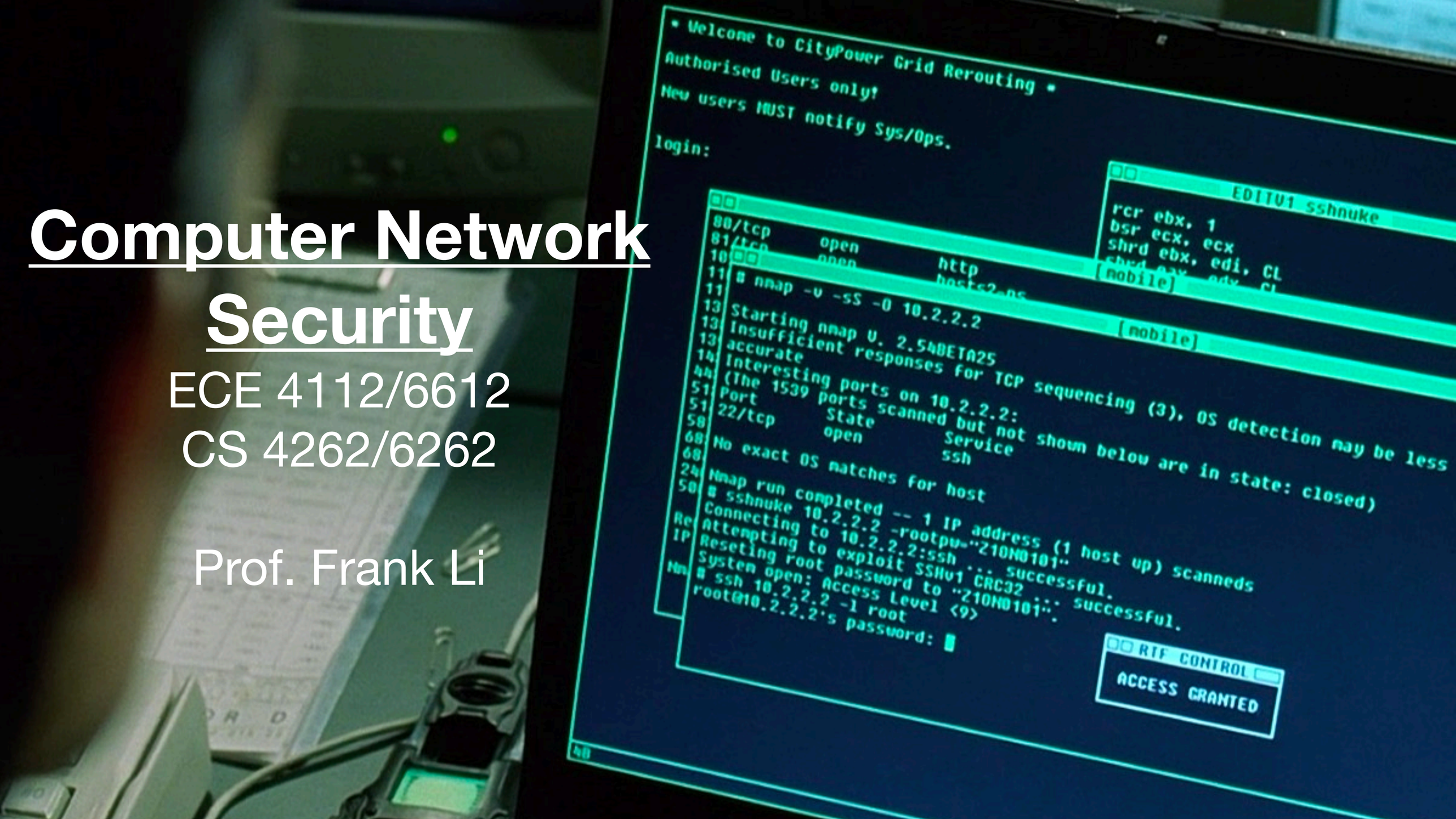
Computer Network Security

ECE 4112/6612
CS 4262/6262

Prof. Frank Li



Logistics

HW3 on web + email security due Monday Nov 13.

Quiz 2 regrades open until today.

Quiz 3 next Thursday (Nov 16):
- web security up to this Thursday's lecture (malware)

Work on your projects!

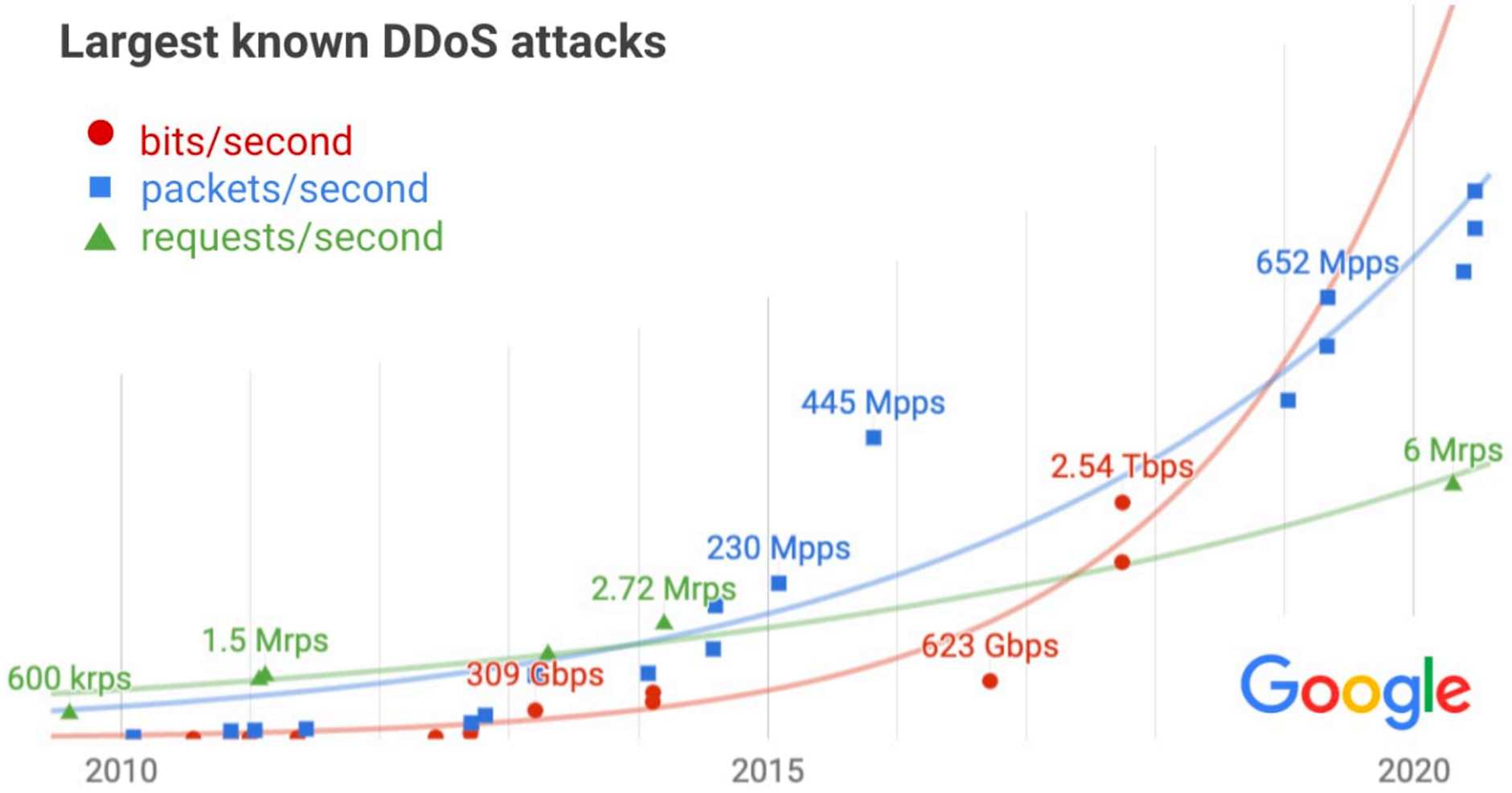
Tue, Oct 10	**No Class (Fall Break)**
Thu, Oct 12	Web security Part 1: Web attacks and defenses
Tue, Oct 17	Web security Part 2: Web attacks and defenses
Thu, Oct 19	Web security Part 3: Web attacks and defenses
Tue, Oct 24*	**Quiz 2**
Thu, Oct 26*	Authentication
Tue, Oct 31	Email Security (Spam, Phishing)
Thu, Nov 2	Network Access Control
Tue, Nov 7	DoS attacks and defenses
Thu, Nov 9	Malware, Botnet
Tue, Nov 14	Last lecture: Censorship and Anonymous Communication
Thu, Nov 16	**Quiz 3**
Tue, Nov 21	**No Class (Early Thanksgiving Break)**
Thu, Nov 23	**No Class (Thanksgiving Break)**
Tue, Nov 28*	Project: Final Project Presentations
Thu, Nov 30*	Project: Final Project Presentations
Tue, Dec 5	Final Class: Final Project Presentations
Thu, Dec 7	**Final Exam: 2:40 - 5:30 PM (Undergraduate Sections Only)**

Last Time: Denial of Service

Defenses Beyond Filtering

- So if filtering isn't fully effective...what else can we do?

- In practice, one of the main defenses today is relying on a cloud/CDN provider to "scale up" service as a defense (e.g., Cloudflare, Google, Akamai, Fastly)
  - In the benign/non-attack state, CDN wouldn't serve too many requests
  - If an attack is detected (or the site is under heavy load, potentially legitimately), the CDN would scale up to serve more requests.



Source: Google Cloud

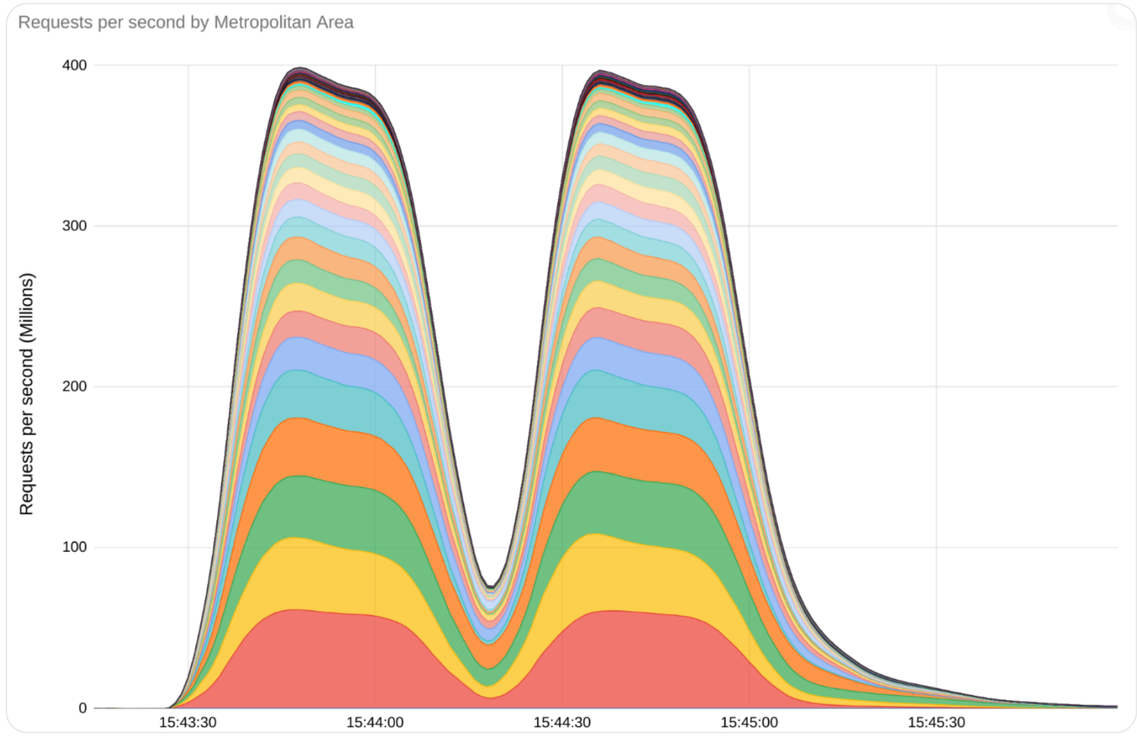
Google Cloud Blog

Contact sales | Get started for fre

Security & Identity

Google mitigated the largest DDoS attack to date, peaking above 398 million rps

October 10, 2023



Google mitigated a DDoS attack which peaked at 398 million requests per second

Source: Google Cloud

Detecting DDoS Attacks across the Internet

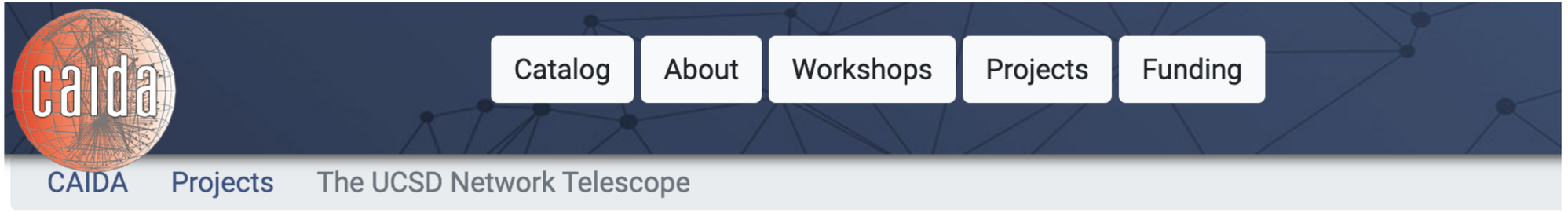
- How can we understand DDoS attacks broadly (especially if we're not a victim/target)?
  - To circumvent filtering, many DDoS attacks will spoof the IP address, most likely picking a random address.
  - That means that in many cases, victim servers will send a response to the spoofed address (called "backscatter").

- Network Telescope: inactive range of IP addresses without real hosts or network services
  - Such telescope addresses will receive spoofed responses from victims. This allows one to study both victims and the nature of attacks (assuming random address spoofing)

Network Telescopes

- Example Analysis (for IPv4):
  - Say we have a /8 subnet as a telescope.
  - Given the IPv4 address space is 2^32, this /8 subnet is 2^24 / 2^32 = 1/2^8 = 1/256 ≈ 0.39% of the IPv4 Internet.
  - Let's say we can identify which packets received by the telescope are from an attack (i.e., a signature for the attack, such as the victim/target IP address).
  - Then, we can estimate attack volume (assuming IID randomly spoofed IP source addresses).
    - » Example: Telescope receives 10K PPS attack traffic. Then we can estimate total attack volume is 10K*(256) = 2.56M PPS.

Network Telescopes



On this page

The UCSD Network Telescope

The UCSD Network Telescope is a passive traffic monitoring system built on a globally routed, but lightly utilized /9 and /10 network. Under CAIDA stewardship, this unique resource provides valuable data for network security researchers.

Network Telescopes

Target Nameserver		A	B	C
December 2020 Attack	Observed Packer Rate (PPM)	21.8K	3.8K	2.9K
	Inferred Traffic Volume	1.4 Gbps	247 Mbps	188 Mbps
	Attacker IP Count	5.79M	1.57M	1.33M
March 2021 Attack	Observed Packer Rate (PPM)	125K	123K	13K
	Inferred Traffic Volume	8 Gbps	7.8 Gbps	845 Mbps
	Attacker IP Count	7M	6.19M	823K

Table 2: Attack metrics for two DDoS attacks on TransIP. The first attack targeted nameserver A more intensely; the second targeted all three similarly.

R. Sommese et al. "Investigating the impact of DDoS attacks on DNS infrastructure." IMC 2022.

Network DoS Summary

- Network-based DoS often relies on overwhelming the network link or network processing of a system, making them unavailable. **Hard to defend against!**
  - Try and detect/filter out attack traffic
  - Try and reduce state/computation
  - "Outmuscle" the attacker by having more computational resources than them (i.e., business model for Akamai + Cloudflare)

Malware

The Problem of Malware

- Malware = malicious code that runs on a victim's system
- How does it manage to run?
  - Attacks a network-accessible vulnerable service
  - Vulnerable client connects to remote system that sends over an attack (a *drive-by*)
  - *Social engineering:* trick user into running/installing
  - "Autorun" functionality (esp. from plugging in USB device)
  - Slipped into a system component (at manufacture; compromise of software provider; substituted via MITM)
  - Attacker with local access downloads/runs it directly
    - Might include using a local "privilege escalation" exploit

What Can Malware Do?

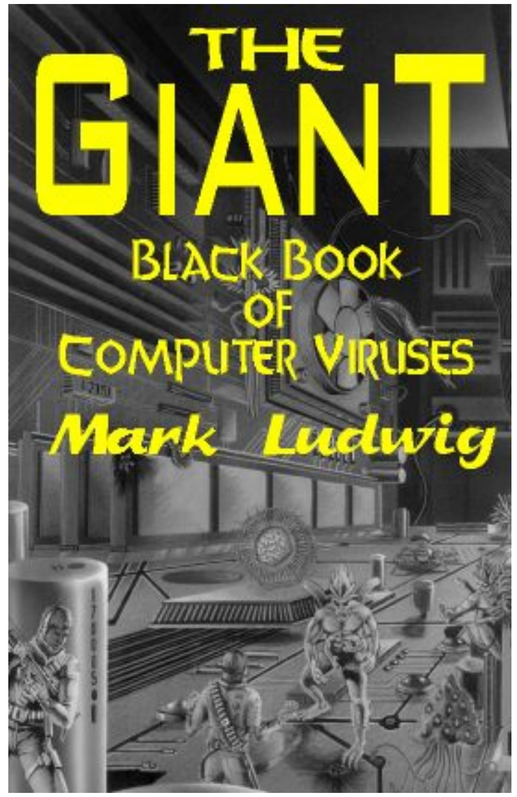
- Pretty much *anything*
  - Only subject to permissions under which it runs
- Examples:
  - Brag or exhort or extort (pop up a message/display)
  - Trash files (just to be nasty)
  - Damage hardware (!)
  - Launch external activity (spam, *online fraud*, DoS; banking)
  - Steal information (*exfiltrate*)
  - Keylogging; screen / audio / camera capture
  - Encrypt files (*ransomware*)
- Possibly delayed until condition occurs
  - “time bomb” / “logic bomb”

Malware That Automatically Propagates

- Virus = code that propagates (**replicates**) across systems by arranging to have itself *eventually executed,* creating a new additional instance
  - Generally infects by altering stored code
- Worm = code that self-propagates/replicates across systems by arranging to have itself *immediately executed* (creating new additional instances)
  - Generally infects by altering running code
  - No user intervention required
- (Note: line between these isn't always so crisp; plus some malware incorporates both approaches)

Viruses

- Opportunistic = code will eventually execute
  - Generally due to user action
    - Running an app, booting their system, opening an attachment
- Separate notions: how it *propagates* vs. what else it does when executed (*payload*)
- General infection strategy: find some code lying around, alter it to include the virus code



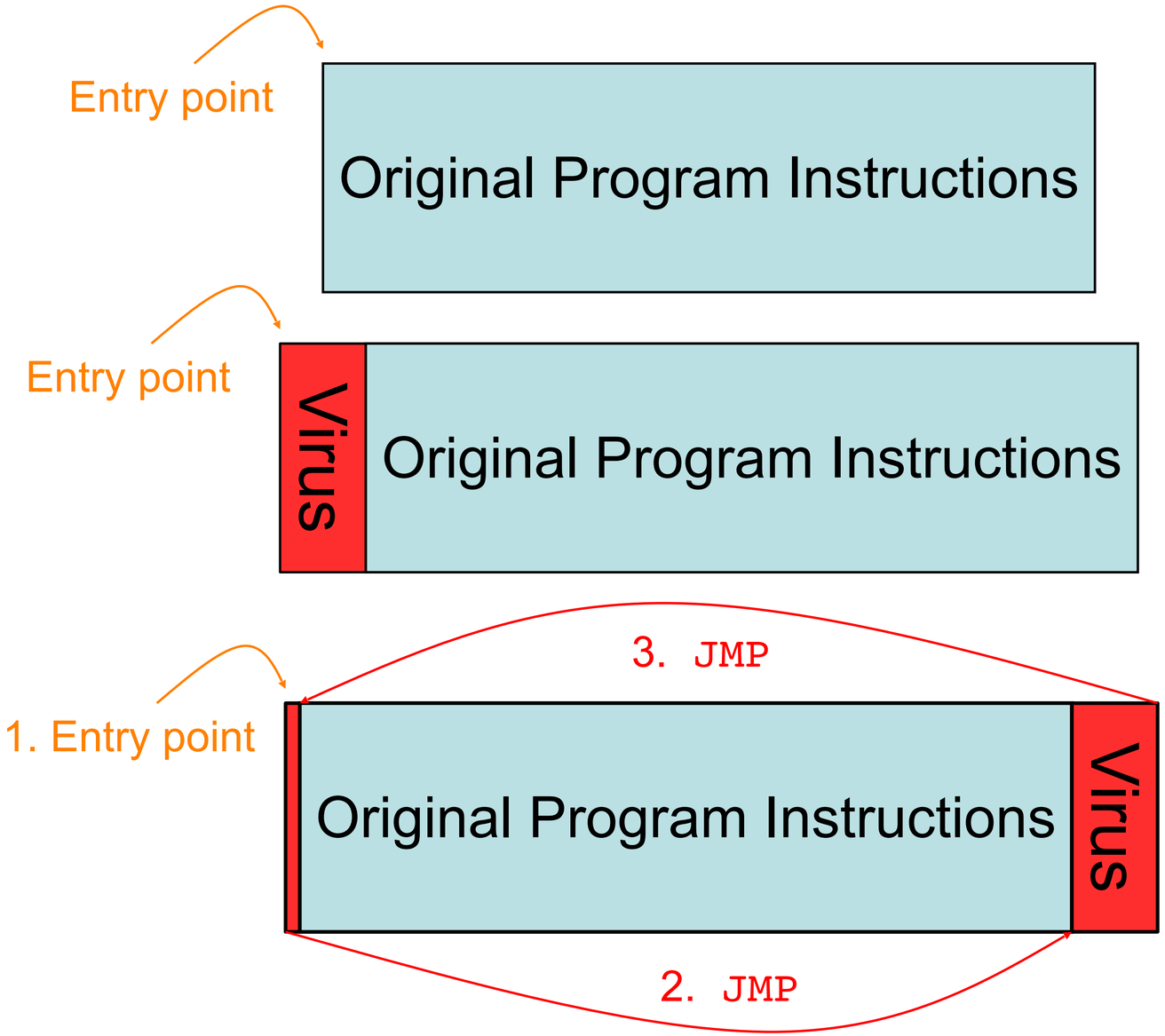
- Have been around for decades …
  - … resulting **arms race** has heavily influenced evolution of modern malware

Virus Propagation

- When virus runs, it looks for an opportunity to infect additional systems
- One example: look for USB-attached thumb drive, alter any executables it holds to include the virus
  - Strategy: when drive later attached to another system & altered executable runs, it locates and infects executables on new system's hard drive
- Or: when user sends email w/ attachment, virus alters attachment to add a copy of itself
  - Works for attachment types that include programmability
  - E.g., Word documents (macros), PDFs (run Javascript)
  - Virus can also send out such email proactively, using user's address book + enticing subject ("ILOVEYOU virus")

***autorun** is handy here!*

Virus Propagation



Original program instructions can be:

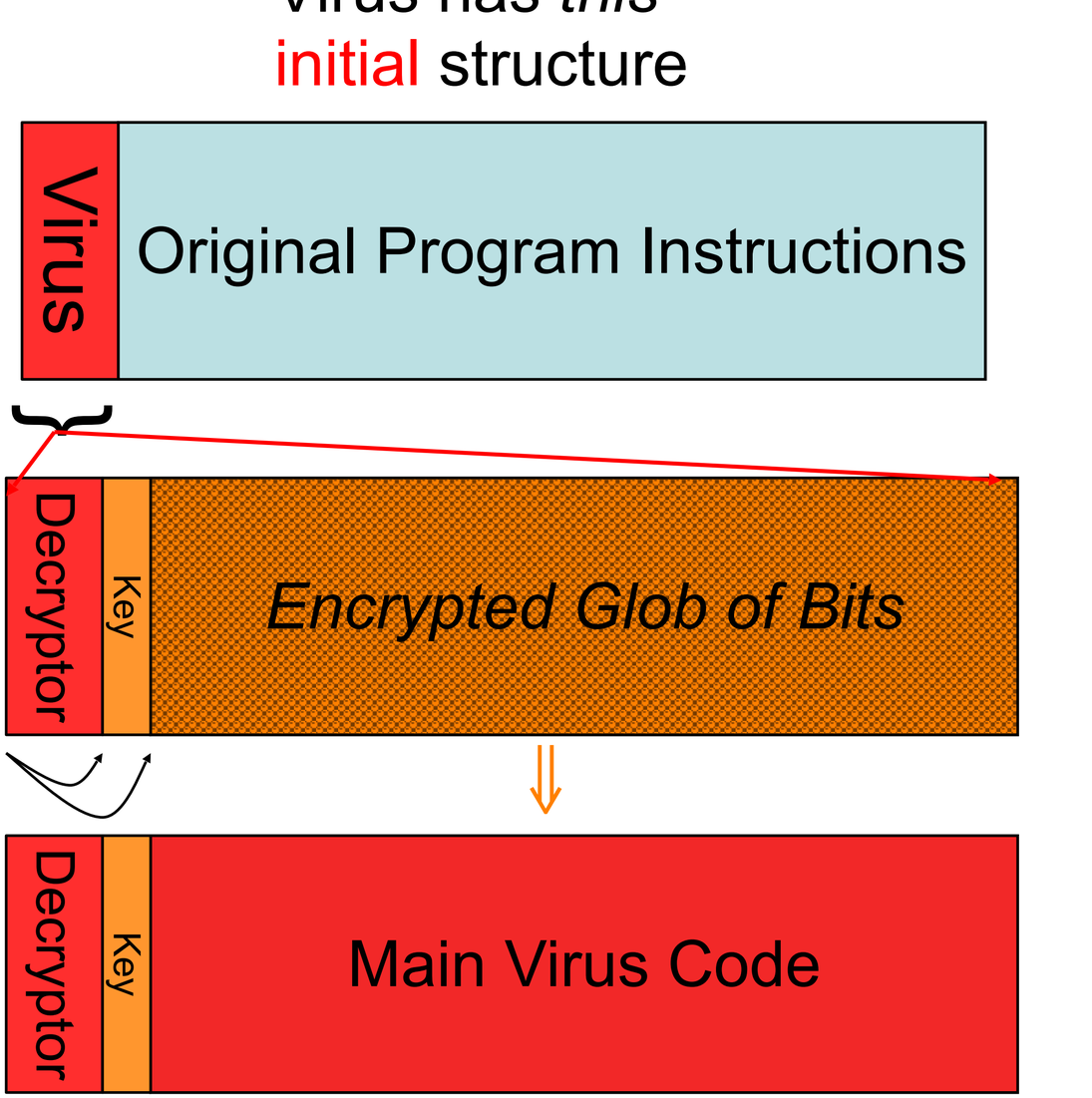
- Application the user runs
- Run-time library loaded in memory
- Disk blocks used to boot OS
- Autorun file on USB device
- …

Many variants are possible! Virus code can be spread out throughout the original program.

Virus Propagation

Additionally, viruses may "mutate" with each infection to avoid easy detection.

- Polymorphism: *encrypt / encode* virus code (e.g., with a different key each time), so it looks different each time. The underlying virus code remains the same. Requires a decryptor that decrypts the virus code when executing.



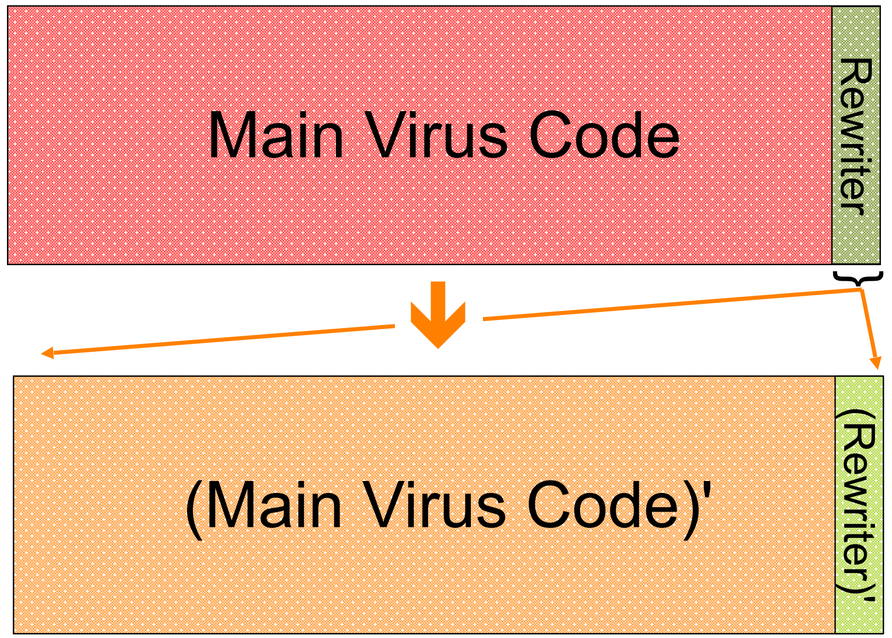
When executed, decryptor applies key to decrypt the glob and run the virus code

Virus Propagation

Additionally, viruses may "mutate" with each infection to avoid easy detection.

- Metamorphism: actually *modify* the virus code to look different each time, while staying semantically equivalent.
  - Examples: Reorder operations. Add/remove meaningless operations. Switch between equivalent algorithms/functions.

When ready to propagate, virus invokes a randomized *rewriter* to construct new but semantically equivalent code (*including the rewriter*)



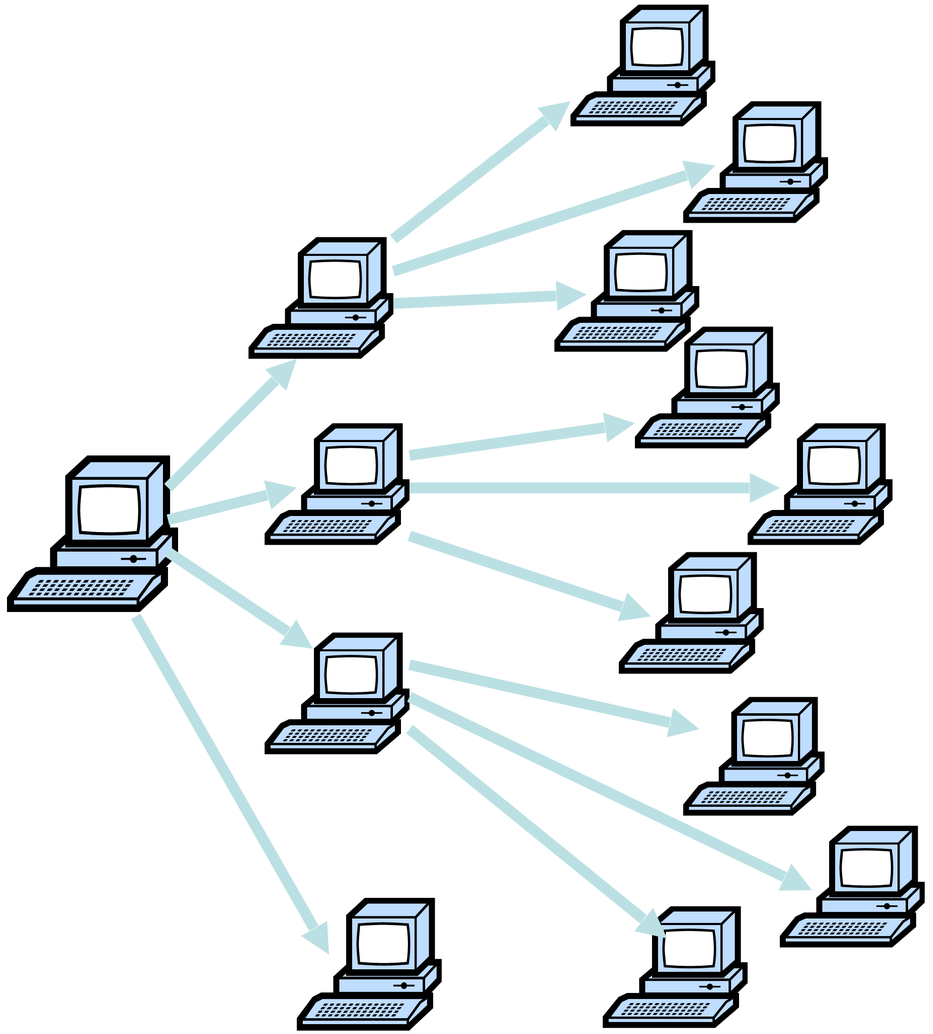
Worms

- Worm = code that self-propagates/replicates across systems by arranging to have itself immediately executed
  - Generally infects by altering running code
  - No user intervention required

Rapid Worm Propagation

Worms can potentially spread quickly because they **parallelize** the process of propagating/ replicating.

Same holds for viruses, but they often spread more slowly since require some sort of user action to trigger each propagation.



Worms

- Worm = code that self-propagates/replicates across systems by arranging to have itself immediately executed
  - Generally infects by altering running code
  - No user intervention required
- Propagation includes notions of *targeting* & *exploit*
  - How does the worm **find** new prospective victims?
    - One common approach: random scanning of 32-bit IP address space
    - But for example “search worms” use Google results to find victims
  - How does worm get code to **automatically run**?
    - One common approach: buffer overflow ⇒ code injection
    - But for example a web worm might propagate using XSS

The Arrival of Internet Worms

- Worms date to Nov 2, 1988 - the *Morris Worm*
- ***Way*** ahead of its time (multiple targeting + exploitation methods developed)
- Created by Robert T. Morris, a grad student at Cornell at the time
  - Not meant to be destructive, but rather highlight the vulnerabilities of many networks.
  - Ended up causing damage (crashing infected hosts)
  - Morris was first person convicted under the US Computer Fraud and Abuse Act (CFAA)
  - Now a CS professor at MIT

Arrival of Internet Worms, con't

- Modern Era began Jul 13, 2001 with release of initial version of Code Red
- Exploited known buffer overflow in Microsoft IIS Web servers
  - *On by default* software on many systems
  - Vulnerability & fix announced previous month
- Defaced websites and launched some DoS attacks

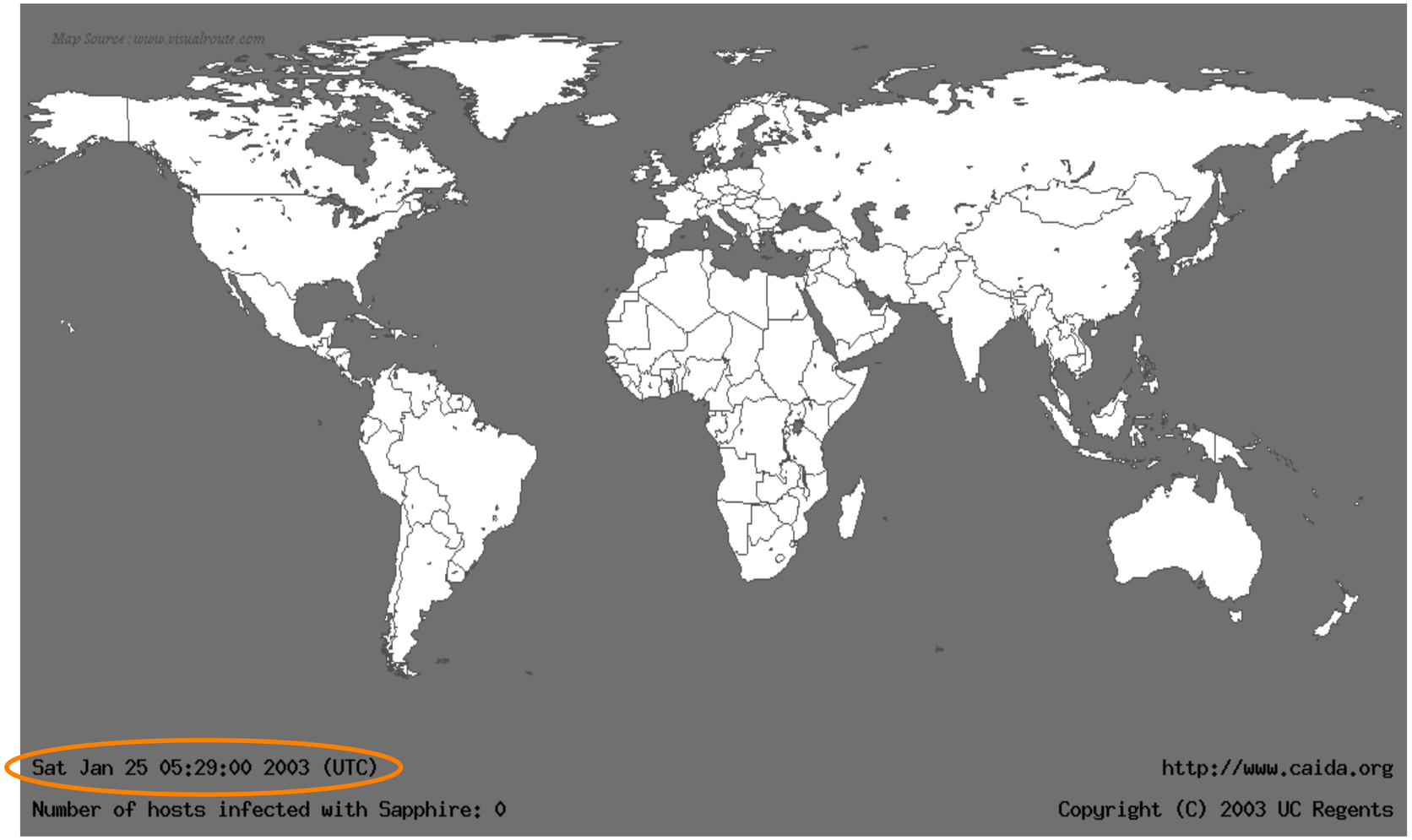
Going Fast: *SQL Slammer*

- 2003: Slammer exploited a connectionless UDP service (related to Microsoft SQL server), rather than connection-oriented TCP
- *Entire worm* fit in a single packet!

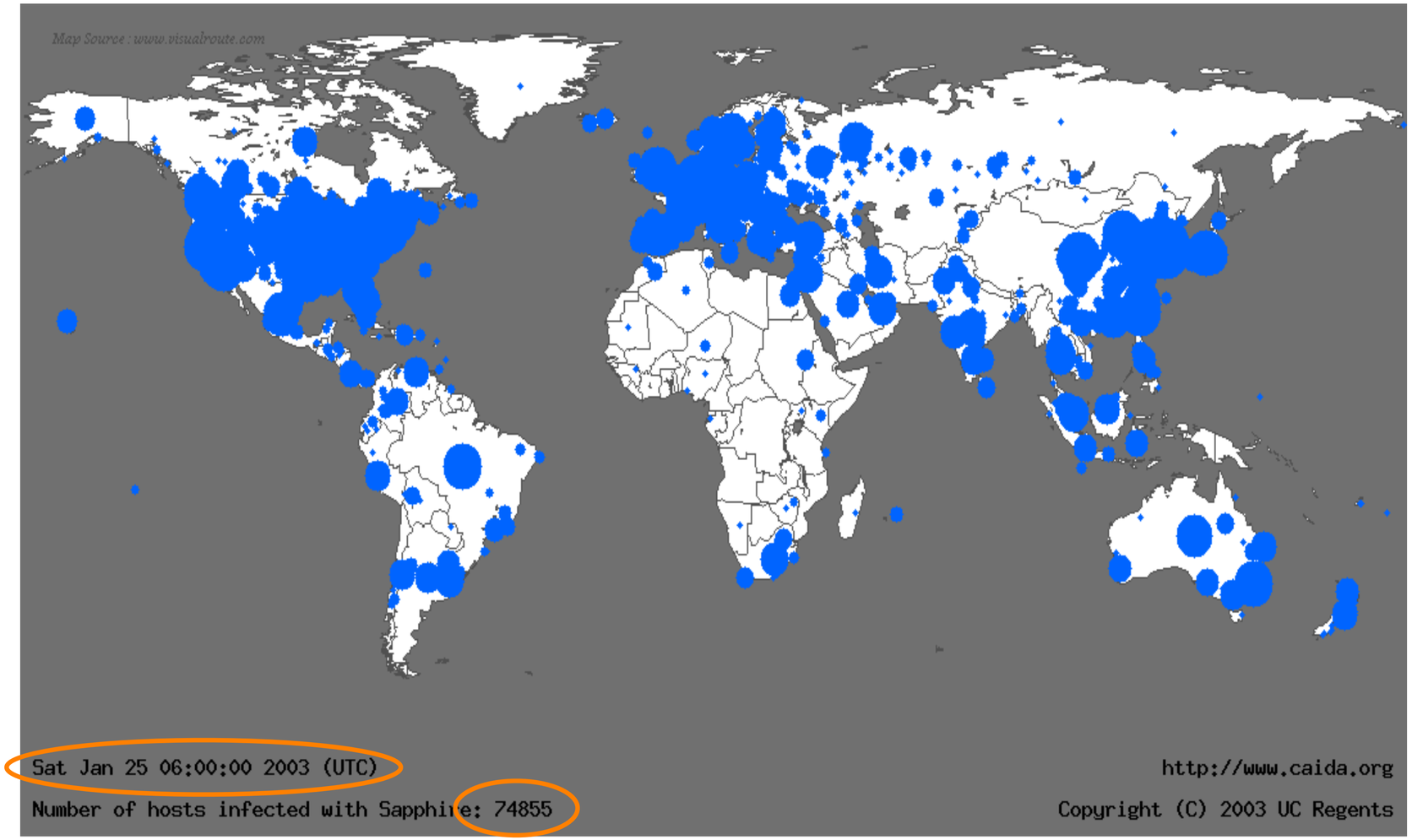
⇒ When scanning, worm could “fire and forget”
Stateless!

- Worm infected 75,000+ hosts in *<< 10 minutes*
- At its peak, **doubled every 8.5 seconds**

Life Just Before Slammer



Life 10 Minutes After Slammer



Worms

- These days, Slammer-style worms are quite rare. Why?
  - More extensive use of stateful firewalls + NATs limited worms' ability to find new targets.
  - Microsoft products were plagued by security problems in early 2000s, many of which were exploited by worms. Microsoft got their security act together.
- Still feasible though (especially as many servers and devices remain publicly accessible)

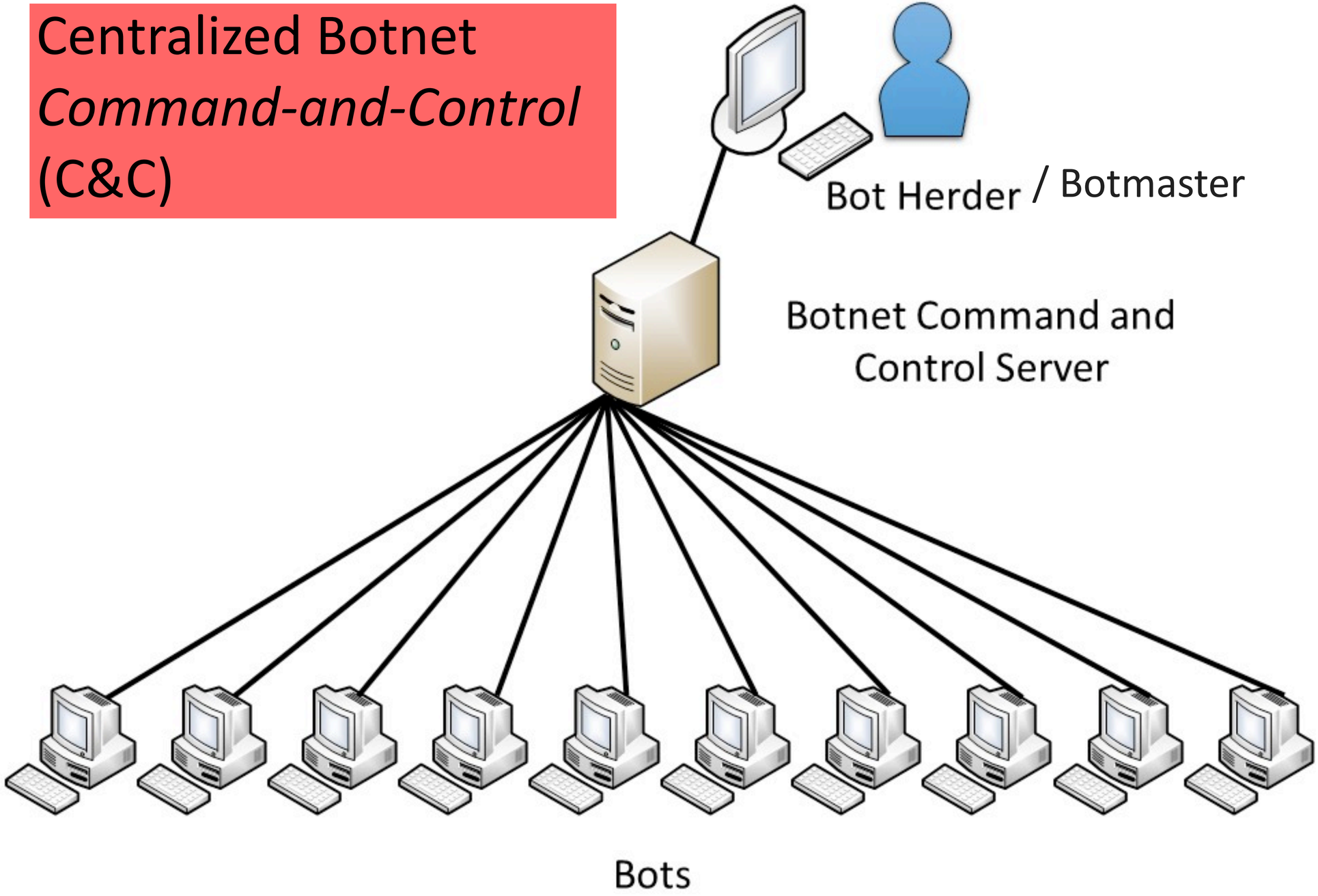
Notable Computer Worms/ Viruses (released by Symantec)

1. **I Love You** (2000)
2. **Conficker** (2009)
3. **Melissa** (1999)
4. **Slammer** (2003)
5. **Nimda** (2001)
6. **Code Red** (2001)
7. **Blaster** (2003)
8. **Sasser** (2004)
9. **Storm** (2007)
10. **Morris** (1988)

Today's Problem: Botnets

- Collection of compromised machines (bots) under (unified) control of an attacker (botmaster)
- Method of compromise decoupled from method of control
  - Launch a worm / virus / drive-by infection / etc.
  - (Or just buy the access – discussed later)
- Upon infection, new bot "*phones home*" to rendezvous w/ botnet *command-and-control* (**C&C**)
- Botmaster uses C&C to push out commands and updates
- Lots of ways to architect C&C:
  - Star topology; hierarchical; peer-to-peer
  - Encrypted/stealthy communication

Centralized Botnet *Command-and-Control* (C&C)



Major Botnets

Storm (2007-2008): email worm, at peak generated about 20% of Internet spam

ZeuS (2007 - present): banking trojan, at one point accounted for ~90% of online bank fraud incidents globally

ZeroAccess (2011 - 2013, 2015 - present): Infected ~9 million machines, conducted extensive ad click fraud and cryptocurrency mining (estimated to have cost online advertisers ~$1M a day at peak)

Emotet (2014 - present): banking trojan + malware loader + ransomware

Mirai (2016 - present): IoT botnet, used for massive DDoS attack that took down major Internet services (e.g., Twitter, Github, Spotify)

Malware Detection

- **Host-based Intrusion Detection Systems (HIDS):** HIDS run on end hosts and aim to detect malware. Anti-virus (AV) is a classic example.

- **Network Intrusion Detection System (NIDS):** Runs in the network, monitoring network traffic (sort of like a firewall, but more complex)

Some Tradeoffs?

- HIDS have full visibility on end host (e.g., syscalls, file accesses, processes, network traffic), while NIDS only have access to network traffic
- HIDS require configuring/managing each machine on the network, whereas NIDS can serve as central monitoring (similar to firewalls)
- If end host is compromised before HIDS detection, HIDS may be compromised.
- Lots of evasion tactics for both (defense in depth, but not perfect itself)

Network-based Malware Detection

- **Signature-Based Detection:** Create a "signature" for malware and its traffic (e.g., unique part of the malware payload, unique part of its generated traffic) that is checked against network traffic to detect malware.
- **Anomaly Detection**: Detect if network traffic is anomalous (in some aspect). Sometimes uses machine learning models (but not always).
- **Specification-based:** Defenders specify exactly what is allowed/disallowed, so block malware-related traffic that violates specifications.
- **Behavioral-based:** Detect and block malicious behavior (e.g., accessing a blacklisted domain)
- **Honeypot:** Set up dummy systems that shouldn't actively be used by real benign users. If network traffic is destined to/from it, most likely malicious.

Fighting Bots / Botnets

- How can we defend against bots / botnets?

- Approach #1: prevent the initial bot infection
  - Equivalent to preventing malware infections in general …. HARD
- Approach #2: Take down the C&C master server
  - Find its IP address, get associated ISP to pull plug

Fighting Bots / Botnets

- How can we defend against bots / botnets?

- Approach #1: prevent the initial bot infection
  - Equivalent to preventing malware infections in general …. HARD
- Approach #2: Take down the C&C master server
  - Find its IP address, get associated ISP to pull plug
- Botmaster countermeasures?
  - Counter #1: keep moving around the master server
    - Bots resolve a domain name to find it (e.g. `c-and-c.evil.com`)
    - Rapidly alter address associated w/ name ("fast flux")
  - Counter #2: buy off the ISP … ("bullet-proof hosting")

BulletProof VPS in Netherlands

fm. $90 USD

If you want a truly authentic European quality connectivity, then our **bulletproof VPS in Netherlands** is the perfect pick for you.

With our promise of 100% uptime, you are getting an unbelievable deal. Because Netherlands have very friendly laws when it comes to content distribution, you can run websites and businesses that may contain sensitive content within Europe.

Simply put – if a certain content is banned to operate in other EU countries, it's probably legal in Netherlands. So if you want a piece of that business, going with a **Bulletproof VPS in Netherlands** is a move you should make.

You can enjoy stellar security, uptime, privacy, and smooth operations from start to finish with our **Netherlands bulletproof VPS service**. Contact us today and feel the difference!

Restrictions

Configurable Options

Processor:	2 core Intel Xeon E3 1230 +$40
Memory:	2048 MiB +$10
Discs:	100 Gb +$20
Network:	unlimited (100Mb/s)
Dedicated IP:	2 +$15
Operating System:	CentOS-6-amd64
Panel:	ISPmanager +$20
Backup size:	5 Gb +$10
Administration:	Optimum +$50

Fighting Bots / Botnets, con't

- Approach #3: seize the domain name used for C&C
- … Botmaster counter-measure?
- Business counter-measure: *bullet-proof domains*

Bulletproof domain registration

fm. 35 USD

Registration of bulletproof domains is conducted by our partners based in China. The reliability of our partners is clearly highlighted by over 5 years of our collaboration and thousands of registered domains.

Bulletproof domains are a must-have for undertaking projects with ample and fierce competition. With **bulletproof domains**, your project will finally be able to function, undeterred by adversaries` attempts to block it through complaints to the domain registrar, while other domains registered from ordinary registrars get blocked in the same circumstances.

Don`t let yourself be pressured or threatened - **register bulletproof domains!**

Type in the domain you wish to register below to check for availability.

www. myhackersite .com GO!

Choose Domains

Domain Name	Status	More Info
myhackersite.com	Available! Order Now	1 Year/s @ $35
myhackersite.net	Available! Order Now	1 Year/s @ $35
myhackersite.org	Available! Order Now	1 Year/s @ $35
myhackersite.biz	Available! Order Now	1 Year/s @ $35
myhackersite.info	Available! Order Now	1 Year/s @ $35
myhackersite.name	Available! Order Now	1 Year/s @ $35

DDoS Protection

fm. $295 USD

Do you need an additional protection for your resource?

Are rivals and ill-wishers trying to disable it?

Our service for **protection against DDoS attacks** will put your mind at ease and help you forget about such problems once and for all!

The most powerful protection will **defeat a DDoS attack** of up to 180 Gbps and 120 million Pps.

Configurable Options

Anti-DDoS: IP protection +$489

- ✓ IP protection +$489
- Domain protection

Billing Cycle

1 mo. 3 mo. 6 mo. yearly

Total Due Today: $784
Total Recurring Monthly: $784

Checkout »

Customer Service

Fighting Bots / Botnets, con't

- Approach #3: seize the domain name used for C&C
- … Botmaster counter-measure?
- Business counter-measure: bullet-proof domains
- Technical counter-measure: DGAs
  - Each day (say), bots generate large list of possible domain names using a Domain Generation Algorithm
    - Large = 50K, in some cases
    - E.g.: `eqxowsn.info, ggegtugh.info, hquterpacw.net, oumaac.com, qfiadxb.net, rwyoehbkhdhb.info, rzziyf.info, vmlbhdvtjrn.org, yeiesmomgeso.org, yeuqik.com, yfewtvnpdk.info, zffezlkgfnox.net`
  - Bots then try a random subset looking for a C&C server
    - Attacker just needs to register & hang onto a small portion of names to retain control over botnet
    - Server **signs** its replies, so bot can't be duped

Fighting Bots / Botnets, con't

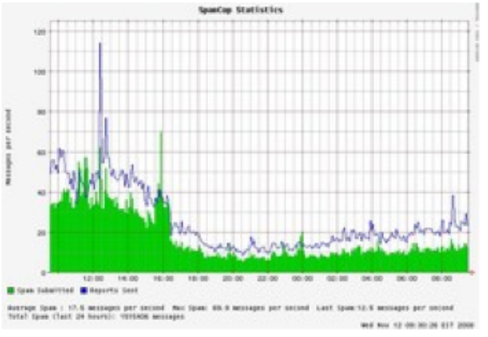
- Approach #4: rally the community to sever bullet-proof hosting service's connectivity

Security Fix

Brian Krebs on Computer Security

Spam Volumes Drop by Two-Thirds After Firm Goes Offline

The volume of junk e-mail sent worldwide plummeted on Tuesday after a Web hosting firm identified by the computer security community as a major host of organizations engaged in spam activity was taken offline. (**Note**: A link to the full story on McColo's demise is available here.)



Experts say the precipitous drop-off in spam comes from Internet providers unplugging **McColo Corp.**, a hosting provider in Northern California that was the home base for machines responsible for coordinating the sending of roughly 75 percent of all spam each day.

In an alert sent out Wednesday morning, e-mail security firm **IronPort** said:

> In the afternoon of Tuesday 11/11, IronPort saw a drop of almost 2/3 of overall spam volume, correlating with a drop in IronPort's SenderBase queries. While we investigated what we thought might be a technical problem, a major spam network, McColo Corp., was shutdown, as reported by The Washington Post on Tuesday evening.

Spamcop.net's graphic shows a similar decline, from about 40 spam e-

Fighting Bots / Botnets, con't

- Approach #4: rally the community to sever bullet-proof hosting service's connectivity
- Lots of effort to conduct though (e.g., legal / political aspects)
- Botmaster countermeasure?
  - Move to another bullet-proof hosting service
  - Peer-to-peer C&C architecture (no single C&C point of failure)
  - Who needs to run a bot when you can buy *just-in-time* bots! Pay-per-install ecosystem.

Installs4Sale.net - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://installs4sale.net/

Most Visited Getting Started Latest Headlines Exchange - GraBBerZ ... GraBBerZ CoM http://www.sysnet.ucs... GraBBerZ CoM Cyber Genome Progra...

Search Sidewiki Bookmarks Translate AutoLink Sign in

Installs4Sale.net

Installs4Sale.net - надежный сервис по загрузкам, достойный доверия

КОНТАКТЫ

560869831

550525933

info [at] installs4sale.net

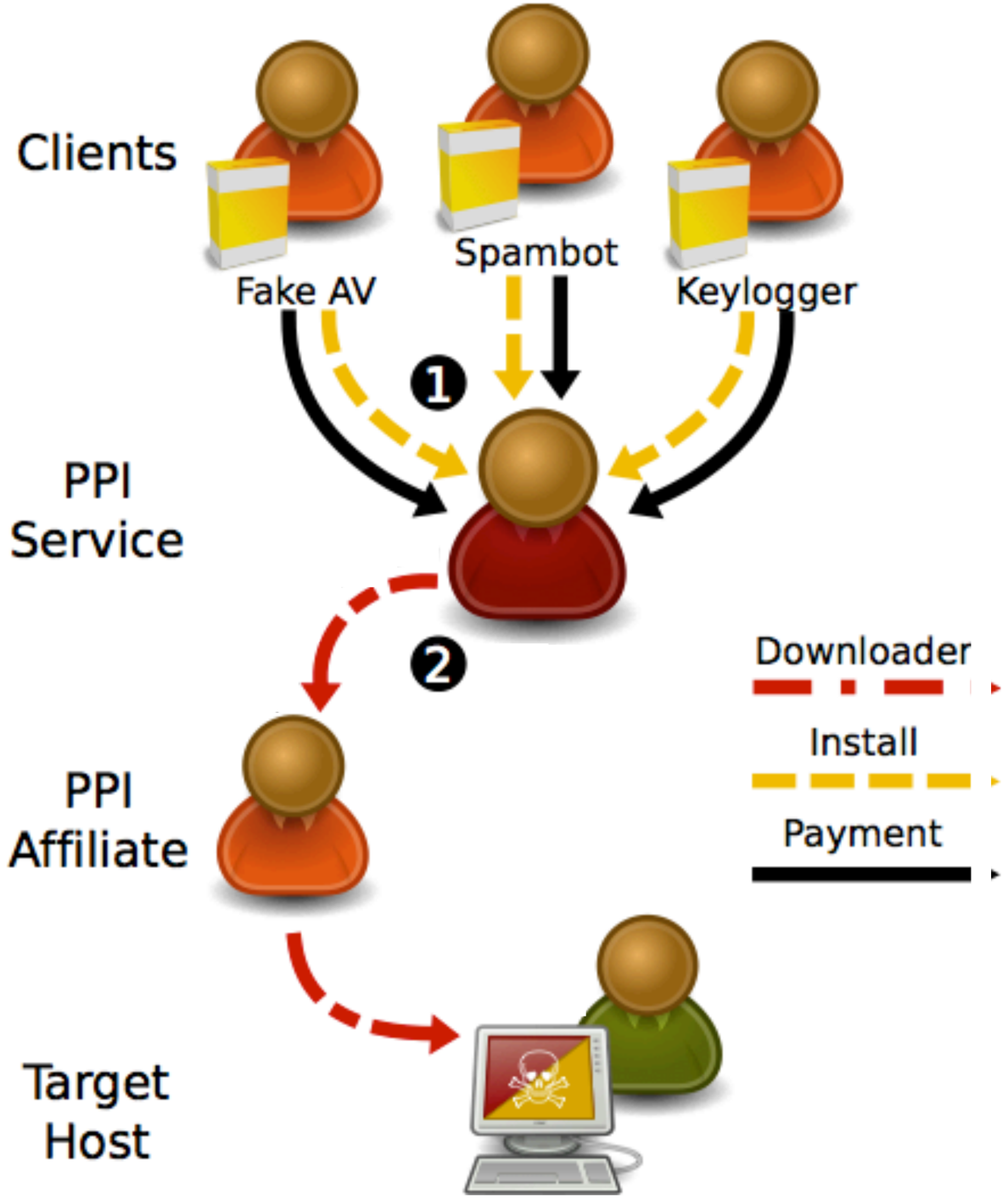
ПРИЕМУЩЕСТВА

- Быстро осуществляем отгрузку практически в любой регион. Принимаем заказы на миксы стран по вашему выбору.
- Для постоянных клиентов действуют скидки и бонусы в виде дополнительного объема загрузок.
- Договоритесь по всем ценам и получите индивидуальные условия вы можете в службе

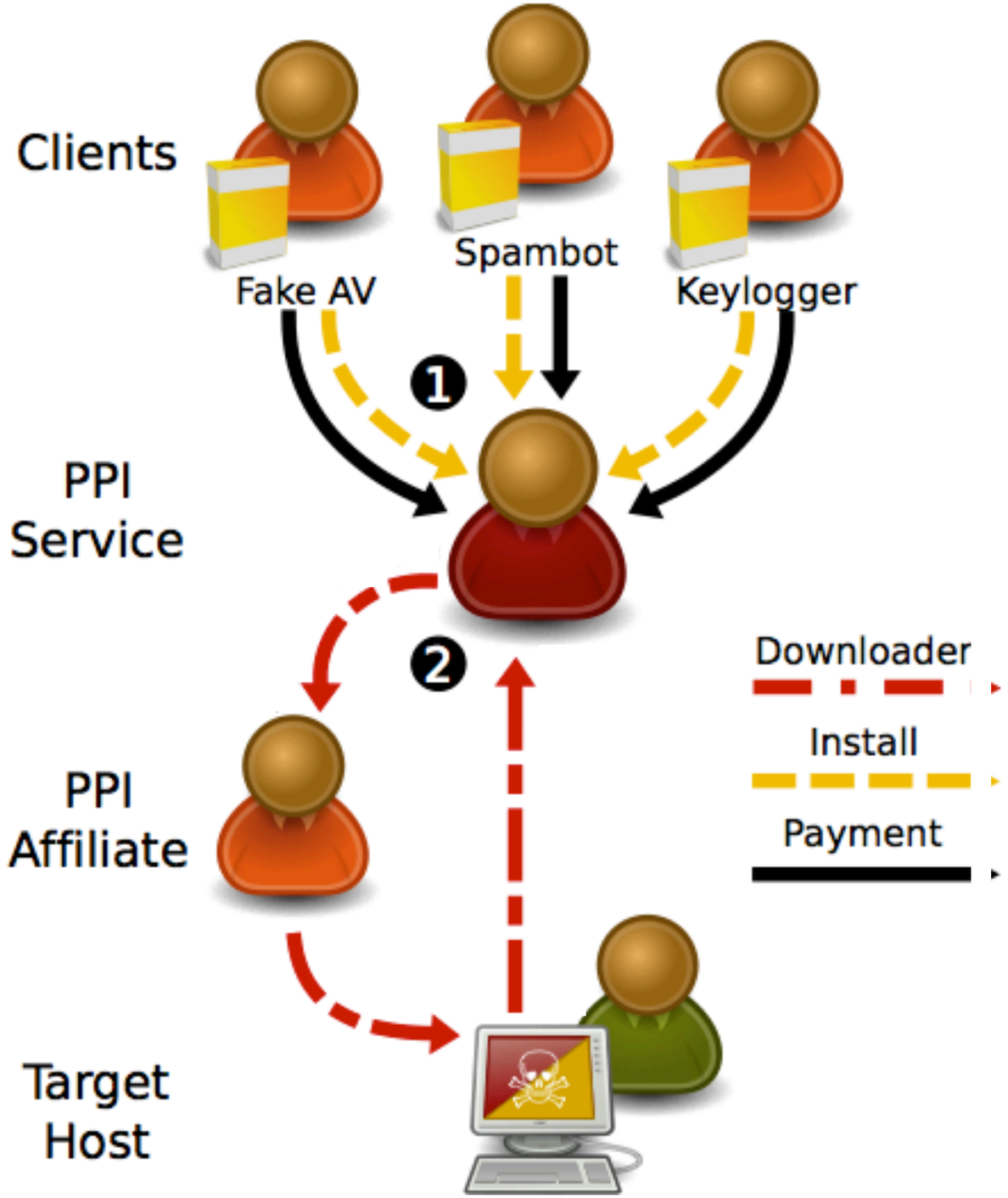
The PPI Eco-system

The PPI Eco-system

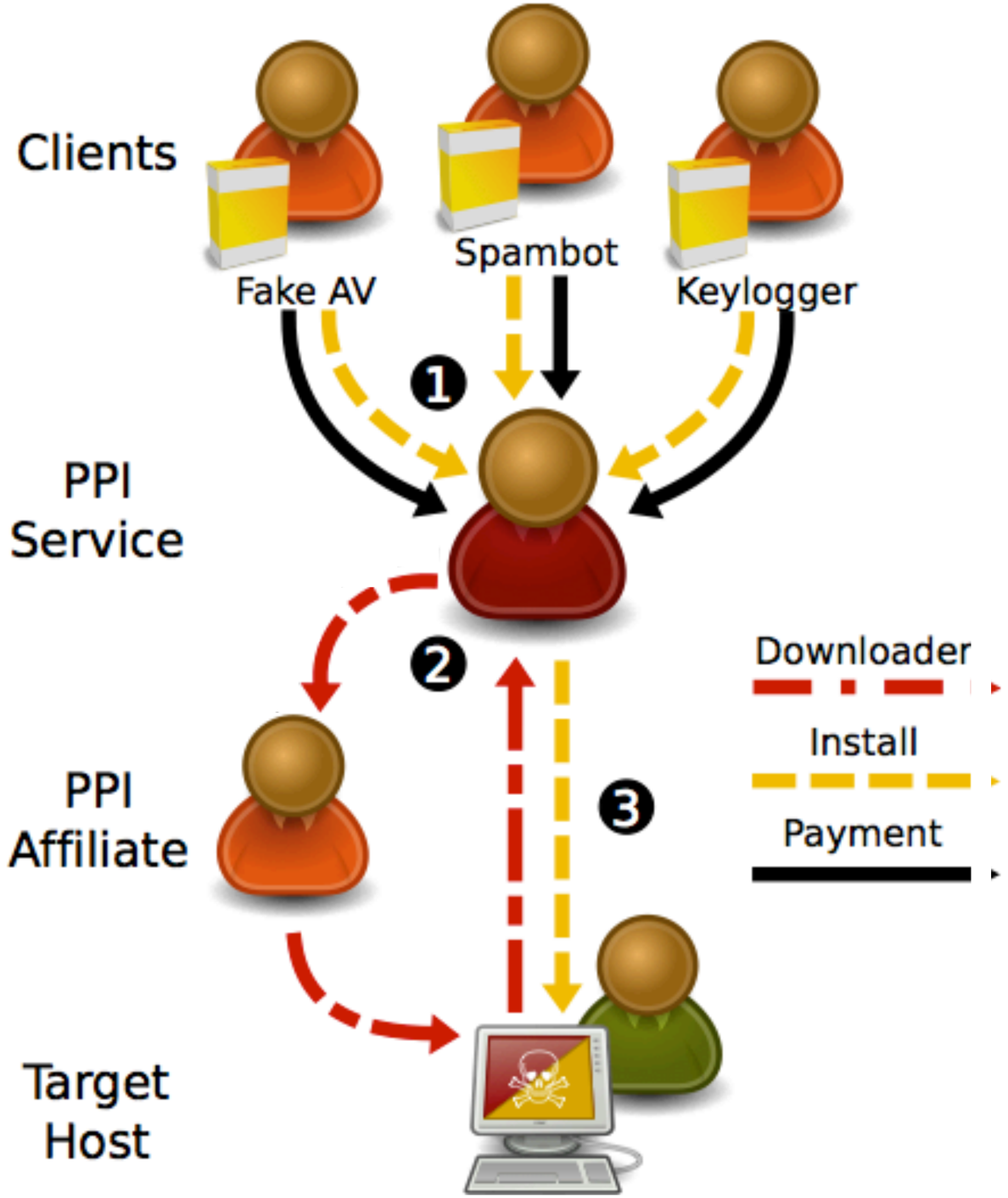
The PPI Eco-system



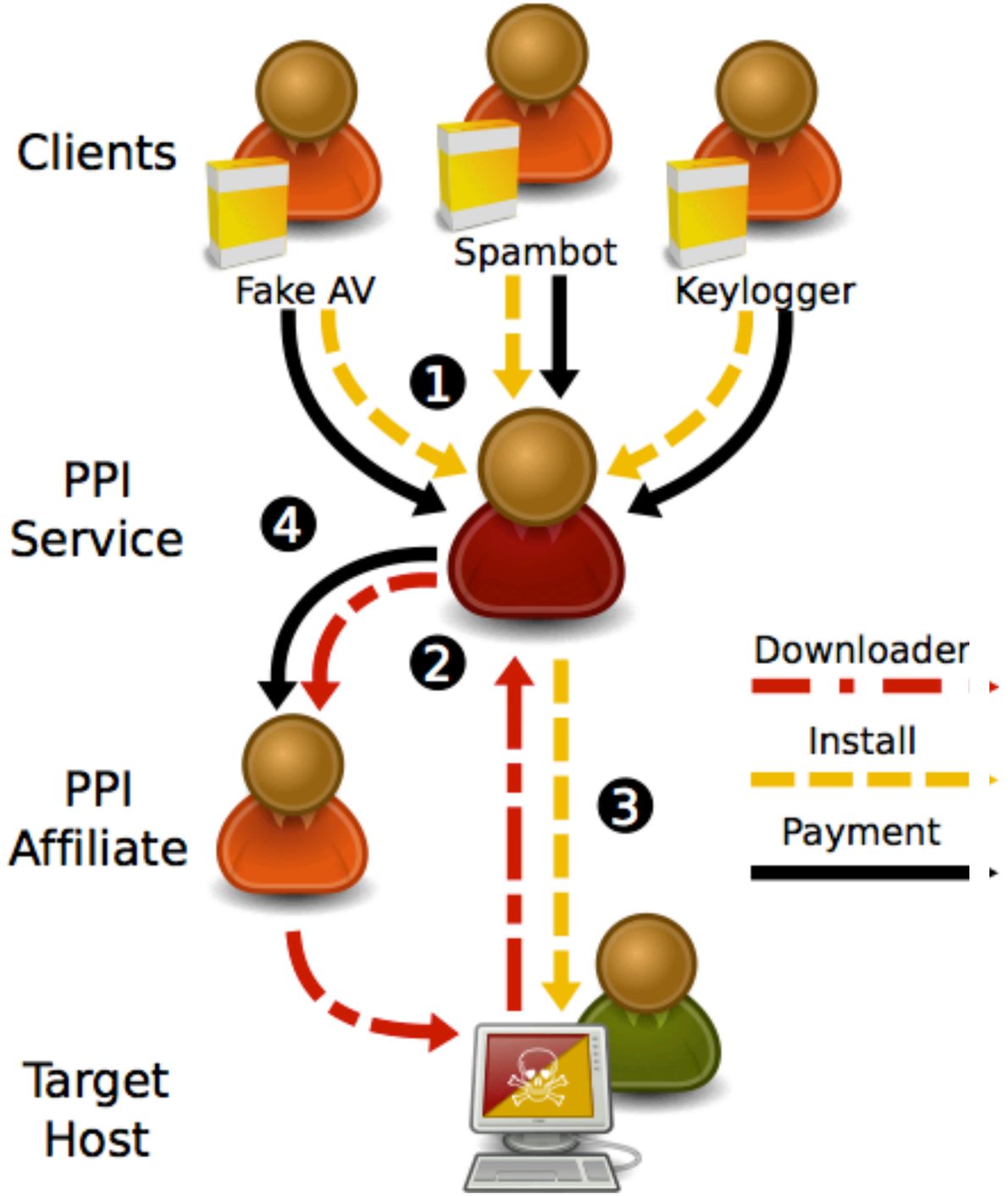
The PPI Eco-system



The PPI Eco-system



The PPI Eco-system



Malware Summary

- Malware is still a major problem today (with notable botnets out there)
  - Different malware propagation methods: virus vs worms
  - Botnets involve centralized control

- Various methods for combating malware, but none are extremely effective.
  - HIDS + NIDS
  - Botnet C&C takedowns
  - Lots of evasion tactics (e.g., polymorphism / metamorphism, bulletproof services, DGAs, PPI)