6.858 Lecture 23
THE ECONOMICS OF SPAM

Administrivia:

- Don't forget to fill out the course evaluation!
- Final project presentations schedule.
    - 3 minutes per group.
    - We tentatively plan to do a 2-hour session until 1pm.
    - Let us know if your group has a hard conflict with 12:30pm--1pm.
- Final project check-off: Sign up for a TA slot.

Up to this point, we've dealt with the *technical* aspects of security (buffer overflows, the same-origin policy, Tor, etc).

- Primary concern: How can an adversary compromise a system? We devise a threat model, and then try to make our system robust against that threat model.
- An alternate perspective: *Why* is the attacker trying to subvert our security policies?
    - Some types of attacks are done for ideological reasons (e.g., political protest by citizens; censorship by governments, Stuxnet). For these kinds of attacks, money is not primary motivator.
        - It's hard to make these attacks more difficult, other than generally "making computers more secure."
        - Economic penalties would likely involve deterrence, i.e., penalties after crime was detected. However, computers and networks currently have poor accountability. Ex: Where did Stuxnet come from? We have some good ideas, but could we win a case in court? And which court would we go to?
    - However, many kinds of computer crime *are* driven by economic motivations (e.g., state-sponsored industrial espionage; spam).
- It takes money to make money!
    - An attacker needs to assemble infrastructure to support an attack. Ex: machines to launch attacks, banks to handle illicit financial transactions.
    - Perhaps we can deter attackers by making their infrastructure costs too high? Ex: Spammers will stop sending spam if it becomes unprofitable!

For today's lecture, we'll focus on attacks that do involve a significant economic component.

- Ex: In China, spammers often hire "text-message cars." The cars intercept communications between cell phones and cell phone towers. The cars discover phone numbers, and then send spam messages to those numbers. One car can send 200,000 messages a day!
    - Cost of interception device: ~$1,600
    - Profit per day: ~$1,600
    - Fines for getting caught: < $5,000

- Since it's rare to get caught, spam cars are a lucrative business.
    - Also, Chinese mobile carriers earn money from each spam message, so the carriers have an incentive to allow the spam to continue.
    - Carriers define special "106-prefix" numbers that are exempt from normal limits on how many messages they can send a day. 106-numbers are supposed to be used for non-commercial reasons (e.g., for companies to contact employees), but 55% of Chinese text message spam comes from 106 numbers.
- Ref: The Economist, "Spam messaging: 106 ways to annoy." November 29, 2014

- There are many companies which trade in "cyber arms." Ex: Endgame.
    - $1.5 million: Endgame will give you the IP addresses and physical locations of millions of unpatched machines.
    - *$2.5 million: Endgame will sell you a "zero-day subscription package" which will give you 25 exploits a year.

    - Who buys exploits from cyber arms dealers? Governments? Companies (e.g., for "hack-back" schemes)? ...?

There's a marketplace for buying and selling all kinds of resources that attackers can use for evil purposes.

- Compromised systems
    - Entire compromised machine.
    - Access to a compromised web site (e.g., post spam, links, redirectors, malware).
    - Compromised email accounts (e.g., Gmail).
    - Running a service on an existing botnet (spam, DoS).
- -Tools
    - Malware kits
    - Bugs, exploits
- -Stolen information
    - SSNs, credit cards numbers, email addresses, etc.

This paper focuses on the spam ecosystem (in particular, the sales of drugs, knock-off goods, and software). There are three main steps:

1) Advertising: Somehow getting a user to click on a link.
2) Click support: Presenting a web site that will be the target of a click.
3) Realization: Allowing the user to buy something, send money, and then receive a product.

Ultimately, money comes from the last part in this chain, when the user buys something.

- Many components are outsourced or supported via affiliate programs: spammers work as the advertisers, but the affiliates handle most/all of the backend stuff (e.g., working with banks).

- Spammers typically work on a commission, getting 30%--50% of the money that they bring in.

Next, we'll discuss these three steps in detail, and look at possible ways to disrupt them.

Advertising: How do you get a user to click on a link?

- Typical approach: send email spam. [Other methods also work: blog/comment spam, spam in social networks, ...]
- Cost of sending spam: $60 per million spam messages, at retail.
    - Actual costs are much lower for direct operators of a spam botnet.
    - Delivery and click rates for spam emails are quite low, so sending spam has to be really cheap in order to be profitable.
    - Earlier study by some of the same guys:
        - ~350 million spams sent
        - ~10k clicks, 28 purchase attempts
- How can we make sending spam more expensive?
    - IP-level blacklists: Used to work for a while, but only if adversary has few IPs.
    - Charging for sending email?
        - Old idea, in various forms: money, computation, CAPTCHAs.
        - Can this work? How could we get everyone to adopt this at once?
        - Will this work even if everyone adopts at once? What if user devices are compromised? [But even with compromised desktops, charging per message may be high enough of a bar to greatly reduce spam, since generating spam needs to be very cheap to be profitable!]
- Three workarounds for adversary:
    - Large-scale botnets give access to many IP addresses.
    - Compromised webmail accounts give access to special IP addresses.
        - Yahoo, Gmail, Hotmail cannot be blacklisted.
    - Hijack IP addresses (using BGP announcements).
- Still, workarounds are not free, and they incur some costs for the spammer.
    - Cost of sending spam used to be even lower before IP-level blacklists.

Botnets are often used to send spam. [Draw picture]

- Typical architecture
    - Many compromised end-user machines that run the botnet software.
    - Command & control (C&C) server/infrastructure for sending commands to bots.
    - Bots periodically get new tasks from C&C infrastructure.
- Individual bot machines have a variety of useful resources:
    - Physical: IP address (good for sending spam), network bandwidth, CPU cycles.
    - Data: email contacts (good for sending spam), credit card numbers, . . .

- It's difficult to prevent bot machines from sending spam --- there may be millions of bot IPs!

How much does it cost to get your malware installed on end-hosts?

- Price per host: ~$0.10 for US hosts, ~$0.01 for hosts in Asia.
- Seems hard to prevent; many users will happily run arbitrary executables.

What does the command and control architecture look like?

- Centralized C&C infrastructure: adversary needs
- "bullet-proof" hosting (i.e., a host that will refuse takedown requests from banks, legal authorities).
    - Bullet-proof hosts charge a risk premium.
- What to do if hosting service is taken down?
    - Adversary can use DNS to redirect. Also, using "fast flux" techniques, the attacker can rapidly change the IP address that is associated with a hostname.
        - Attacker creates a list of server IP addresses (there may be hundreds or thousands of IPs); attacker binds each one to the hostname for a short period of time (e.g., 300 seconds).
    - How hard is it to take down botnet's DNS domain name?
        - Can take down either domain's registration, or the domain's DNS server.
        - Adversary can use domain flux, span many separate registrars.
            - Harder to take down: requires coordination between registrars!
            - Happened for Conficker: it was significant/important enough…
- Decentralized C&C infrastructure: peer-to-peer networks.
    - Allows bot master to operate fewer or no servers; hard to take down.

Compromised webmail accounts can also be used to send spam.

- Very effective delivery mechanism: everyone accepts email from Yahoo, Gmail, etc.
- Webmail providers are motivated to prevent accounts from being compromised.
    - If the provider doesn't prevent spam, then *all* mail from that provider may be marked as spam!
    - The provider monetizes the service using ads, so the provider needs real users to click on ads.
- How do providers detect spam?
    - Monitor messages being sent by each account, detect suspicious patterns.
    - For suspicious messages and initial signup/first few msgs, use a CAPTCHA: present the user with an image/sound, ask user to transcribe it---this should be easy for a human, but hard for a computer.
- How hard is it to get a compromised webmail account?
    - Price per account: ~$0.01-0.05 per account on Yahoo, Gmail, Hotmail, etc.

Why are webmail accounts so cheap? What happened to CAPTCHAs?

- Adversaries build services to solve CAPTCHAs; it's just a matter of money.
    - Turns out to be quite cheap: ~$0.001 per CAPTCHA, with low latency.
    - Surprisingly, it's mostly done by humans: attacker can outsource the work to any country with cheap labor. [Could also use Amazon's Mechanical Turk: It's a crowd-sourced web service that allows humans to work on tasks that are difficult for computers to perform.]
- Instead of hiring someone to solve the attacker, the attacker can reuse the CAPTCHA on another site, and ask a normal visitor to solve it.
- Providers can implement more frequent checks for spam senders, but regular users may get annoyed if the checks are too frequent.
    - Ex: Gmail lets you enable two-factor authentication. In this scheme, when you open Gmail from a previously unknown machine, Google will send you a verification code via SMS.

Click support: The user contacts DNS to translate a hostname into an IP address; then, the user contacts the associated web server. So, the spammer needs to:

1) register a domain name.
2) run a DNS server.
3) run a web server.

Q: Why do spammers bother with domain names? Why not just use raw IP addresses to serve content?

- A1: Users might be less likely to click on a link that has a raw IP address in it?
- A2: A stronger reason is that using a layer of indirection makes it easier to keep the content server alive.
    - If law enforcement deregisters the domain name or disables the DNS server, but the server is still alive, the spammer can just register a new domain name and create a new DNS server.

- Spam URLs often point to redirection sites.
    - Free redirectors like bit.ly or other URL shorteners.
    - A compromised site can also perform a redirect to the spam server.
- Redirection sites are useful because spam filtering systems may blacklist URLs.
    - A popular site is extremely useful as a redirection platform: to stop the spam, filtering software would have to blacklist a popular website!
- Spammers sometimes use botnets as web servers or proxies.
    - This hides the IP address of the real web server; indirection once again!

In some cases, a single affiliate provider will run some or all of these services.

- Q: Can't law enforcement just take down the affiliate program?
- A: In theory, yes, but it can be hard to take down the entire organization. Also, there are a non-trivial number of affiliate programs.

- Q: How difficult is it to take down individual domain names or web servers?
- A: Depends on the registrar or hosting provider [see Figures 3, 4, 5 in the paper].
    - Only a few number of registrars host domains for many affiliates; similarly, only a few number of ASes host web servers for many affiliates.
    - Only a few affiliates distribute their domain, name server, and web server across many registrars and ASes.
    - Bullet-proof hosting providers are more expensive, but plentiful; even if they're taken down, they're relatively easy to replace.

What happens during the realization phase?
1) User pays for goods.
2) User receives goods in the mail (or downloads software).

Payment protocol: almost invariably credit cards.
- Credit card info travels along this flow:

```
Customer
|---->Merchant
     |----> Payment processor (helps the
          |  merchant deal with the
          |  payment protocol)
          |
          |-->Acquiring bank (merchant's)
          |-->Association network
               | (e.g., Visa)
               |
               |---> Issuing bank
               (customer's)
```

- The issuing bank decides whether the transaction looks legit, and if so, sends an approval back.
- PharmaLeaks paper: Some programs have over $10M/yr revenue!

For physical goods, the supplier typically ships the goods directly to purchaser (this is called "drop shipping").
- Drop shipping means that the affiliate program does not need to stockpile physical products themselves.
- Authors speculate that there are plenty of suppliers, so there's no supply-side bottleneck.

Q: Why do spammers properly classify their credit card transactions?
A: The association network (e.g., Visa or Mastercard) charges high fines for miscoded transactions! Presumably, association networks don't want to get in trouble for obscuring the true purposes of financial transactions.

Q: Why do spammers actually ship the goods?

A: Credit card companies track the number of "chargeback" requests (i.e., the number of times that customers ask their credit card company to return funds involved in a broken transaction).

  - Credit card companies issue penalties if the number of chargeback transactions is too high (>1%).
  - So, it's not sustainable for a spammer to frequently charge customers but not ship goods, particularly if . . .

Only a few banks are willing to interact with spammers! [Table V, Figure 5 in the paper]

- CCS'12 paper: Only 30 acquiring banks seen over 2 years!
- So, an effective spam prevention technique is to focus on those small number of banks. Why?
  - High cost to switch banks.
  - Financial risk in switching.
- But what can we actually do?
  - Convince issuing banks to blacklist these acquiring banks?
  - Try to convince these banks to stop dealing with spammers? This may be tricky: "Due to incongruities in intellectual property protection, it is not even clear that the sale of such goods [like pharmaceuticals] is illegal in the countries in which such banks are located." [Section IV.D]
    - -Spamming is distasteful, but it's not always criminal.
    - -For example, some affiliate customers might not come from spam---they might come from legitimate Google searches!

Since this paper was published, credit card networks have taken some action.

- After the paper came out, some pharmacy and software vendors lodged complaints about Visa [the authors used Visa cards to make their spam purchases].
- In response, Visa made some policy changes:
  - All pharmaceuticals sales are now labeled high-risk; if a bank acts as an acquirer for high-risk merchants, the bank is more strictly regulated (e.g., the bank needs to engage in a risk-management program).
  - Visa's operating guidelines now explicitly enumerate and forbid illegal sales of drugs and trademark-infringing goods.
  - The new language allows Visa to aggressively issue fines against acquiring banks.
  - Some affiliate programs responded by requiring customers to submit a photo ID (the goal was to filter out test buys from security researchers). However, this hurts sales, since customers are reluctant to give their ID info.

Does this paper raise ethical concerns? Are the authors supporting the spammers by purchasing their goods?

Some companies have launched "hack-back" campaigns to retaliate against theft of intellectual property, or to stop botnets involving their machines.

- Ex: In 2013, Microsoft, American Express, PayPal, and a bunch of other companies took down a large botnet. Microsoft then told the affected users that they should patch their machines.
- Microsoft's legal reasoning: Botnets were violating Microsoft trademarks. Increasingly, companies are using novel legal arguments to take action against botnets . . . is this a good idea?

Reference:

- http://css.csail.mit.edu/6.858/2013/readings/captcha-econ.pdf
- http://css.csail.mit.edu/6.858/2013/readings/priceless.pdf [CCS'12]
- http://css.csail.mit.edu/6.858/2013/readings/pharmaleaks.pdf
- http://www.usenix.org/media/events/atc11/tech/videos/savage.mp4
- http://research.microsoft.com/pubs/167719/whyfromnigeria.pdf

6.858 Computer Systems Security
Fall 2014