Malware and Cybercrime
Today

- We’ve talked about ways that machines can be compromised
- But what happens afterwards?
  - Malware & anti-malware
  - Botnets
  - Cybercrime
Viruses & worms

- Replicating malicious programs
  - Viruses replicate by attaching to a host program (or document)
    - Copying themselves into new programs/documents they encounter
    - Traditionally driven by human action (i.e., opening document)
  - Worms replicate via the network
    - Each compromised host tries to infect other hosts; parallelism
    - Self-spreading

- Goals:
  - Evade detection
  - Spread
  - Accomplish their goal (payload)… whatever that is
Quick aside: why is self-replication interesting?

- Because it allows massive compromise for low investment in resources

- Some worms have taken over hundreds of thousands of hosts in a day; others have covered the entire Internet in 10 minutes
How to think about malware outbreaks

- Well described as infectious epidemics
  - Simplest model: Homogeneous random contacts

- Classic SI model
  - $N$: population size
  - $S(t)$: susceptible hosts at time $t$
  - $I(t)$: infected hosts at time $t$
  - $\beta$: contact rate
  - $i(t) = I(t)/N$, $s(t) = S(t)/N$

\[
\begin{align*}
\frac{dI}{dt} &= \beta \frac{IS}{N} \\
\frac{dS}{dt} &= -\beta \frac{IS}{N}
\end{align*}
\]

\[di \over dt = \beta i(1 - i)\]

\[i(t) = e^{\beta(t-T)} \over 1 + e^{\beta(t-T)}\]
Two things matter when considering the scope of a malware outbreak

- How likely is it that a given infection attempt is successful?
  - Target selection (random, biased, hitlist, topological, ...)
  - Vulnerability distribution (e.g. density – $S(0)/N$)

- How frequently are infections attempted?
  - $\beta$: Contact rate
Back to Malware:
Virus design history

- Boot Sector viruses
  - Historically important, but less common today

- Memory resident viruses
  - Standard infected executable

- Encrypted viruses

- Polymorphic viruses

- Each new advancement is the result of co-evolution – Darwinian requirement that malware authors improve to survive
Boot sector Viruses
(old school)

- Vector: floppy disks

- Bootstrap Process:
  - Firmware (ROM) copies MBR (master boot record) to memory, jumps to that program

- MBR (or Boot Sector)
  - Fixed position on disk
  - “Chained” boot sectors permit longer Bootstrap Loaders
Boot sector Viruses

- Virus breaks the chain
- Inserts virus code
- Reconnects chain afterwards
Why attack the Bootstrap?

- Automatically executed *before* OS is running
  - Any thus, before detection tools are running

- OS hides boot sector information from users
  - Hard to discover that the virus is there
  - Harder to fix

- Any good malware scanning software scans the boot sectors
  - But good bootsector viruses may restore good bootsector during normal operation (replace it when you logout or when anti-virus software isn’t running)
  - Bootsector malware has re-emerged (Meebroot/Sinowal) and now versions that rewrite device firmware (e.g., on the disk drive, the BIOS, network card, etc)
Virus Attachment to Host Code

- Simplest case: insert copy at the beginning of an executable file
  - Runs before other code of the program
  - Historically most common program virus

- Runs before & after original program
  - Virus can clean up after itself

- Virus could modify code in place
  - Doesn’t change size, but could change behavior
  - Maybe harder to detect?
Detecting Malware

▪ **Scanning (signatures)**
  
  ▪ Integrity checking (check if file has changed)
    - Keep “known good” hash of existing executables (whitelist);
      validate programs on computer against whitelist
  
  ▪ Behavior (heuristic) detection
    - E.g. does software use system features atypical of an application program; make anomalous network access; try to read sensitive files, etc...
Virus Signatures

- Viruses can’t be completely invisible:
  - Code must be stored somewhere
  - Virus must do something when it runs
  - Identify existing viruses and extract “signature” byte sequences unique to them
  - Idea: look in files these signatures

- Issues
  - Where to scan (beginning of file, whole file, registry settings, etc)
  - How to scan (just look for string, or actually execute program)
  - How long to scan (tradeoffs in performance/coverage)
  - How to distinguish polymorphs (research issue)
1. User runs an infected program.
2. Program transfers control to the virus.
The Simple Virus

Infected Program

3. Virus locates a new program.
4. Virus appends its logic to the end of the new file.
5. Virus updates the new program so the virus gets control when the program is launched.
Head/Tail Scanners

Most of these application-infecting viruses attached themselves to either the top or bottom of the host file:

So anti-virus engineers built head/tail scanners.

The scanner loads the head and tail regions of the file into a buffer and then scans with a multi-string search algorithm.
So what do the bad guys do?

- Move the virus to the middle of the file
- Becomes prohibitively expensive to scan
  - Must scan whole file

- Solution: scalpel scanning
  - Idea: limit scanning to likely entry-points for viruses
  - If you have more time you can also scan for more than just strings (regular expressions)
Scalpel Scanning

1. Locate the main program entry-point.

2. While the current instruction is a JUMP or a CALL instruction, trace it.

3. If the current instruction is not a JUMP or CALL instruction, search for all fingerprints in this region of the file.
The Encrypted Virus

Soon after the first generation of executable viruses, virus authors began writing self-encrypting strains.

These viruses carry a small decryption loop that runs first, decrypts the virus body and then launches the virus.

Each time the virus infects a new file, it changes the encryption key so the virus body looks different.
The Encrypted Virus

1. MOV DI, 120h
2. MOV AX, [DI]
3. XOR AX, 5132h
4. MOV [DI], AX
5. ADD DI, 2h
6. CMP DI, 2500h
7. JNE 3
8. WJSVTBPMZPL
9. NAADJGNANW
...

The decryption routine stays the same. Only the key(s) change.

The encrypted body changes.

1. MOV DI, 120h
2. MOV AX, [DI]
3. XOR AX, 0030h
4. MOV [DI], AX
5. ADD DI, 2h
6. CMP DI, 2500h
7. JNE 3
8. PKEPAJHENZAW
9. MNANTPOOTIZN
...

Still easy to detect because the decryption loop stays the same.
The Polymorphic Virus

- Polymorphic viruses are self-encrypting viruses with a changing decryption algorithm

- When infecting a new file, such a virus:
  - Generates brand-new decryption code from scratch
  - Encrypts a copy of itself using a complementary encryption algorithm
  - Inserts both the new decryption code and the encrypted body of the virus into target file
The Polymorphic Virus

1. User Executes Program

Host Program

Decryption Loop

Virus

Mutation Engine

RAM
The Polymorphic Virus

1. User Executes Program
2. Virus Decrypts Itself
The Polymorphic Virus

1. Host Program
2. Decryption Loop
3. Virus
4. Mutation Engine
5. RAM

3. Virus finds new prog.
**The Polymorphic Virus**

3. Virus finds new prog.
4. Mutation engine creates new decryptor.
5. Virus makes a new copy of itself and encrypts this copy.
The Polymorphic Virus

5. Virus makes a new copy of itself and encrypts this copy.
6. Virus appends the new decryptor and encrypted virus body to new file.
The Polymorphic Virus

Host Program

Decryption Loop

Virus

Mutation Engine

Decryption Loop'

Host Program (New)

Decryption Loop'

Virus

Mutation Engine

Mutation Engine
The Polymorphic Virus

And we have a new infection!
Polymorphic malware: Extremely difficult to detect...

- May be no shared code between two malware samples of the same strain

- Heuristics
  - Xray scanning: guess the key
  - Try to analyze semantics of what code does (blows up quickly)

- Generic decryption
  - Assumptions
    - Virus gains control of the host immediately
    - Virus decrypts itself deterministically
    - Virus has a some static body that can be detected with traditional signatures
  - Key idea:
    - Emulate code execution until the virus decrypts itself
      - Typically use some sort of virtual machine (VM) environment
    - Search for signatures in memory
“Generic” Decryption

- Invented by Alan Solomon (a.k.a. Dr. Solomon)
  - Chose name to obscure how it worked

- Assumptions
  - Virus gains control of the host immediately
  - Virus decrypts itself deterministically
  - Virus has a some static body that can be detected with traditional signatures

- Key idea:
  - Emulate code execution until the virus decrypts itself
    - Typically use some sort of virtual machine (VM) environment
  - Search for signatures in memory
1. Load suspected program into VM.
2. Allow the program to execute normally.
1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.

1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1
Generic Decryption

1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.

- 1. Fetch Byte
- 2. Decrypt Byte
- 3. Store Byte
- 4. Loop to 1

Virtual Machine

Simulated OS and other data structures

Host Program

Decryption Loop

Virus

Mutation Engine

Modified Memory
1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.

1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1

Simulated OS and other data structures
Host Program
Decryption Loop
Virus
Mutation Engine
Virtual Machine
Modified Memory
Generic Decryption

1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.
Generic Decryption

1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.

And on it goes...

1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1

39
Generic Decryption

1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.

And on it goes...

1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1

Virtual Machine

Simulated OS and other data structures

Host Program

Decryption Loop

Virus

Mutation Engine

Modified Memory
**Generic Decryption**

1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.

And on it goes...

```
1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1
```

Simulated OS and other data structures

Host Program

Decryption Loop

Virus

Mutation Engine

Virtual Machine

Modified Memory
## Generic Decryption

1. **Load suspected program into VM.**
2. **Allow the program to execute normally.**
3. **“Tag” all modified memory as the program executes.**

### Diagram:

1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1

And on it goes...

<table>
<thead>
<tr>
<th>Simulated OS and other data structures</th>
<th>1. Load suspected program into VM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Program</td>
<td>2. Allow the program to execute normally.</td>
</tr>
<tr>
<td>Decryption Loop</td>
<td>3. “Tag” all modified memory as the program executes.</td>
</tr>
<tr>
<td>Virus</td>
<td></td>
</tr>
<tr>
<td>Mutation Engine</td>
<td></td>
</tr>
<tr>
<td>Virtual Machine</td>
<td></td>
</tr>
<tr>
<td>Modified Memory</td>
<td></td>
</tr>
</tbody>
</table>

### Algorithm:

1. **Fetch Byte**
2. **Decrypt Byte**
3. **Store Byte**
4. **Loop to 1**

And on it goes...
Generic Decryption

1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.
4. Scan all modified areas of virtual memory for virus signatures.

And on it goes...

1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1

Simulated OS
and other data structures

Host Program

Decryption Loop

Virus

Mutation Engine

Virtual Machine

Modified Memory
Generic Decryption

1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.
4. Scan all modified areas of virtual memory for virus signatures.
But you’ve just kicked the can down the road

- How long to emulate program?
  - Emulate too long and the system slows to a crawl
  - Don’t emulate enough and you might miss the virus

- What if malware can tell its running inside a VM?
  - Don’t decrypt if you are

- What about malware that only activates with some specific input?

- This wheel turns many more times (and none of it is unique to viruses... true of all malware)

- Bottom line: detection is complex and malware authors constantly work to make it harder to do host-based malware detection

- Key assumptions of anti-malware software:
  - Malware is known a priori (i.e., there are good signatures that can be extracted)
  - Malware is used again (i.e., that discovering new malware instance is useful)
So you’ve taken over 100,000 machines...

- Then what?

- Use machines *together* for some purpose
What’s a botnet?

- A network of compromised computers with a common command & control system (C2)
  - Each host called a bot
- The bot controller sends controls the network en masse
  - Spam, phishing
  - Denial-of-service [e.g., dirtjumper]
  - Click fraud
  - Stealing local data (e.g. credit cards, passwords, bank account #’s, etc) [e.g., zeus, spyeye]
  - Botnets typically self-updating
Botnet Architecture

- **Bot network**
  - Centralized:
    - IRC server (Internet relay chat) or Web server
    - Multiple servers for robustness (e.g., try round-robin)
  - Peer-to-peer: self organizing
    - Each host can be a worker or a proxy; decided dynamically
    - Multi-level hierarchy forwards traffic back to controller

- **Push vs pull designs**
  - Attacker sends out message to tell bots what to do (push)
  - Worker bots “ask” for work to do (pull)

- **Recovery (resilience in case of “takedown”)**
  - Most common “Domain Generation Algorithms”
  - If you can’t reach C2, then contact domain X.com (where X is generated based on date)
Example:
Storm peer-to-peer botnet
Dealing with botnets

- Detection
  - Detecting the malware
  - Detecting network access to known C2s or using known botnet C2 protocols
  - Allow “honeypot” computers to be infected and see what they do
  - Infiltrate or takeover C2 and see which hosts connect to it (the infected ones)

- What to do?
  - Get users to cleanup infected hosts (e.g., Microsoft Malware Removal Tool)
  - Quarantine from network (some ISPs do this)
  - Shutdown C2 (carefully) or disrupt its behavior
  - Takeover C2 and command bots to deinstall themselves (legal issues)
So... what do people do with botnets?
Economic Drivers

- Starting in 2005, emergence of profit-making malware
  - Anti-spam efforts force spammers to launder e-mail through compromised machines (starts with MyDoom.A, SoBig)
  - “Virtuous” economic cycle transforms nature of threat

- Commoditization of compromised hosts
  - Fluid third-party exchange market (millions of hosts)
    - Raw bots (range from pennies to dollars)
    - Value added tier: SPAM proxying (more expensive)

- Innovation in both host substrate and its uses
  - Botnets: sophisticated command/control networks: platform
  - SPAM, piracy, phishing, identity theft, DDoS are all applications
Installs4Sale.net - надежный сервис по загрузкам, достойный доверия

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.
- Бесплатная доставка по всему миру.
Договоримся по всем ценам и получить индивидуальные условия вы можете в службе поддержки.

Мы отслеживаем уникальность инсталов и их чистоту перед продажей.

УСЛОВИЯ

- Мы работаем своего продукта. Допускается частичная оплата постоянным клиентам на большие объемы
- Мы не несем ответственности за то, что у вас по каким-то причинам отсутствует загрузка. Если вы не видите инсталов с первых минут, мы можем произвести отгрузку до выполнения обстоятельств.

ТАРИФЫ

<table>
<thead>
<tr>
<th>Страна</th>
<th>Цена</th>
</tr>
</thead>
<tbody>
<tr>
<td>GB (Англия)</td>
<td>150$</td>
</tr>
<tr>
<td>DE (Германия)</td>
<td>150$</td>
</tr>
<tr>
<td>USA (США)</td>
<td>130$</td>
</tr>
<tr>
<td>IT (Италия)</td>
<td>120$</td>
</tr>
<tr>
<td>Микс (US, CA, AU, GB)</td>
<td>100$</td>
</tr>
<tr>
<td>CA (Канада)</td>
<td>100$</td>
</tr>
<tr>
<td>Микс (Европа)</td>
<td>40$</td>
</tr>
<tr>
<td>Азия</td>
<td>10$</td>
</tr>
</tbody>
</table>

Все цены указаны за 1000 уникальных загрузок.

Все права защищены install4sale.net, 2009.
Last news

<table>
<thead>
<tr>
<th>Date</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.12.2006</td>
<td>From today our price for Asia grows up to 1.55 for 1k and the price for Italy - to 300$ for 1k.</td>
</tr>
<tr>
<td>26.11.2006</td>
<td>For the reason of bad price for Antilles region we have to lower our price for it to 125. We're waiting for your understanding. We'll work on this problem as soon as possible.</td>
</tr>
<tr>
<td>11.07.2006</td>
<td>Now, we accept axis looks!</td>
</tr>
<tr>
<td>11.06.2006</td>
<td>We resolve our problem with hosting! And we have a special bonus: you'll get +30% more to your money!</td>
</tr>
<tr>
<td>31.05.2005</td>
<td>From the 31th of May the new system of anti-influenza is started.</td>
</tr>
<tr>
<td>07.11.2005</td>
<td>Problems with BackURL solved, use it!</td>
</tr>
<tr>
<td>11.10.2005</td>
<td>Now you can send not unique traffic to your resources with help of BackURL</td>
</tr>
<tr>
<td>10.10.2005</td>
<td>From the 10th of October the new system of tracking IS STARTED. From this moment we pay different $$$ for different countries</td>
</tr>
<tr>
<td>19.09.2005</td>
<td>From the 19th of September the price for 1000 loads will rise to 80$</td>
</tr>
<tr>
<td>3.08.2005</td>
<td>New system of statistics and new design are started!</td>
</tr>
<tr>
<td>11.07.2005</td>
<td>From the 11th of July the price for 1000 loads will rise to 70$</td>
</tr>
</tbody>
</table>

Adverts link

<table>
<thead>
<tr>
<th>HTML Link:</th>
<th><img src="http://wpjnddgpq.biz/dl/adv622.php" alt="HTML Link" /> width=1 height=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden HTML link:</td>
<td><img src="" alt="Hidden HTML Link" /> width=1 height=1</td>
</tr>
<tr>
<td>EXE Link (last update 68 hours ago):</td>
<td><img src="http://wpjnddgpq.biz/dl/leadadv622.exe" alt="EXE Link" /></td>
</tr>
</tbody>
</table>
Making money...

- Monetize platform of compromised host
  - **Generic resources**: CPU, IP address, bandwidth, storage
  - **Unique resources**: e-mail accounts, credit card numbers, bank accounts, intellectual property

- Ultimately, must find a way to “cash out”...
Two core criminal value creation strategies...
Click fraud

- **Assumption:**
  - Click on ad is a customer

- **Attack**
  - Deplete other ad budgets
  - Click on own ads for revenue

- **What is done**
  - Identify fraudulent patterns (e.g., many clicks from IP, no sales)
  - Refund money from those
Infostealers

- Infected machines gather information from the disk or as it is typed and send it back
  - Either via command & control channel
  - Or to “dead drop” (e.g., Web site that anyone can read)

- Commercial use (e.g., Zeus/Spyeye)
  - Gathering credentials for online services, banks, credit cards, etc

- Espionage use (e.g., Ghostnet/Flame)
  - Gathering documents of value
Zeus example

```sql
Zeus :: Statistics

Information:
- Profile: [value]
- GMT date: 11.03.2009
- GMT time: 14:13:27

Statistics:
- Summary

Botnet:
- Online bots
- Remote commands

Logs:
- Search
- Search with template
- Uploaded files

System:
- Profiles
- Profile
- Options
- Logout

Table:
<table>
<thead>
<tr>
<th>Botnet: Any</th>
<th>Installs (157)</th>
<th>Online bots (378)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GB</td>
<td>32</td>
<td>TH</td>
</tr>
<tr>
<td>--</td>
<td>23</td>
<td>--</td>
</tr>
<tr>
<td>RU</td>
<td>19</td>
<td>RU</td>
</tr>
<tr>
<td>US</td>
<td>10</td>
<td>GB</td>
</tr>
<tr>
<td>TH</td>
<td>14</td>
<td>US</td>
</tr>
<tr>
<td>DE</td>
<td>6</td>
<td>TR</td>
</tr>
<tr>
<td>IN</td>
<td>0</td>
<td>IN</td>
</tr>
<tr>
<td>SE</td>
<td>3</td>
<td>PE</td>
</tr>
<tr>
<td>IE</td>
<td>2</td>
<td>PE</td>
</tr>
<tr>
<td>CH</td>
<td>2</td>
<td>HU</td>
</tr>
<tr>
<td>MY</td>
<td>1</td>
<td>SA</td>
</tr>
<tr>
<td>SA</td>
<td>1</td>
<td>EG</td>
</tr>
</tbody>
</table>

```

```sql
Zeus :: Statistics

Information:
- Total logs in database: 3677358
- Total bots: 3985
- Total active bots in 24 hours: 678

```
Zeus example
Infostealers

- Best infostealers can defeat two-factor authentication

- In-browser malware
  - Allow user to authenticate normally to bank
  - Piggyback theft transaction (wire transfer) on this login
  - **Rewrite bank javascript** so it balance is “fixed up” and theft transaction is invisible

- Requires custom malware for each bank (typically target one bank at a time)
Cashout

- So... you’ve stolen a bunch of credit cards, or back account credentials.... Now what?

- Direct monetization
  - “White plastic”: burn new cards and do cash withdrawals (usually outsourced for 50% commission)
  - Wire transfer (to other US bank), then “money mules” withdraw money & transfer via Western Union

- Reshipping fraud
  - Purchase goods online (dense value per pound) with stolen credit cards and send to US address
  - Reshipping mules receive item and reship to overseas location
Fraud: FakeAV

- Two vectors
  - Infected machine pops up warning
  - Compromised Web site creates fake warning for visitors
    - Aside: search engine optimization (SEO) and abuse another big use for botnets (i.e., poisoning Google search results)

- Warning indicates that machine is infected

- Looks like a real AV system

- Offers to clean you machine if you subscribe (e.g., $50)
Fraud: FakeAV
Extortion: Ransomware

- Malware encrypts all files and requires machine’s owner to pay to unlock
  - Typically uses non-standard payment instruments: e.g., paysafecard, Bitcoin
  - Will unlock data with payment

- Two kinds of lures:
  - Fraudulent:
    - We are the FBI/BKA/RIAA/etc.... You have copyrighted material, child pornography, etc... on your machine... you will be brought to court unless you settle
  - Straight out extortion:
    - Pay us or you’ll never see your files again
ATTENTION!

IP: [redacted]
Location: [redacted]
IPS: [redacted]

Your PC is blocked due to at least one of the reasons specified below.

You have been violating Copyright and Related Rights Law (Video, Music, Software) and illegally using or distributing copyrighted content; thus infringing Article 1, Section 8, Clause 8, also known as the Copyright of the Criminal Code of the United States of America.

Article 1, Section 8, Clause 8 of the Criminal Code provides for a fine of two to five hundred minimal wages or a deprivation of liberty for two to eight years.

You have been viewing or distributing prohibited Pornographic content (Child Porn/ Zoophilia and etc.). Thus violating article 202 of the Criminal Code of United States of America. Article 202 of the Criminal Code provides for a deprivation of liberty for four to twelve years.

Illegal access has been initiated from your PC without your knowledge or consent, your PC may be infected by malware, thus you are violating the law On Neglectful Use of Personal Computer. Article 210 of the Criminal Code provides for a fine of up to $100,000 and/or a deprivation of liberty for four to nine years.
Ransomware
Largest botnet application: spam

- Overview of spam and anti-spam
- Local research in spam economics
What is spam?

- To you:
  - Mail you didn't want
- To e-mail providers:
  - Bulk e-mail their customers didn't want
- To the law:
  - Bulk commercial e-mail that doesn't abide by a set of rules
    - CAN SPAM Act
Spam “applications”

- Marketing
  - Selling goods/services
  - Stock spam
  - Advanced Fee Fraud (419 scams)
- Attraction (taking you to a site)
  - Phishing/spear phishing
  - XSS, CSRF attacks
  - Drive-by malware
- Infection via attachments
Gathering targets

- **Harvesting**
  - Web crawling (home pages, myspace, etc)
  - News, Mailing list crawling
  - Malware harvesting
  - Blind addressing

- **Stealing lists from enterprises/providers**

- **Purchasing mailing lists**
  - Legal: opt-in
  - Other...
How Email Works: Quick Overview

helo test
250 mx1.mindspring.com Hello
abc.sample.com
[220.57.69.37], pleased to meet you
mail from: test@sample.com
250 2.1.0 test@sample.com... Sender ok
rcpt to: jsmith@mindspring.com
250 2.1.5 jsmith... Recipient ok
data
354 Enter mail, end with "." on a line by itself
from: test@sample.com
to: jsmith@mindspring.com
subject: testing
John, I am testing...
.
250 2.0.0 e1NMajH24604 Message accepted for delivery
quit
221 2.0.0 mx1.mindspring.com closing
Connection
Connection closed by foreign host.
More on e-mail

- A particular domain (e.g. ucsd.edu) has a small number of mail servers for all outbound mail (e.g., smtp.ucsd.edu)

- However, it is possible for each machine to send mail directly
Sending spam

- Base message composition
- Mass mailing program
  - Interface with target lists
  - Add polymorphism/specialization/personalization
  - Connect to delivery infrastructure
- Delivery infrastructure
  - Send from own machine
    - Can have many RCPT TO: addresses in one e-mail
  - Launder origin via open relays/proxies
  - Launder origin via Email service provider
  - Launder origin via botnet
Example: Send-Safe
What to do about it?

- Block reception
  - Blacklisting
  - Sender authentication
  - Content filtering

- Change economic model
  - Charge sender per message

- Change addressing model

- Legal remedy
  - CAN-SPAM act
Blacklisting

- Detect spam
  - Honeyclients (dummy e-mail accounts)
  - User reports
  - Anomaly detectors plus inspection

- Save the IP address that sent you the spam

- Report to Blacklisting service

- Configure mail servers to validate each IP address against blacklisting service before accepting e-mail

- Issues?
Sender authentication

- Validate that purported origin domain could have generated the message
  - From: obama@whitehouse.gov [132.239.1.2]

- SPF
  - Do DNS lookup on domain, get list of IPs that are allowed to send mail for that domain; validate

- DomainKeys
  - Mail header includes digital signature
  - Recipient does DNS lookup on domain to get public key and verifies signature with it
    - Yahoo, Gmail, AOL

- Issues?
Content filtering

- Phrase filtering
  - Known suspect keywords (e.g. Viagra, Cialis)

- Heuristics
  - All CAPITAL letters, embedded images, came from estonia, spoofed header, IP address space is dynamic, etc

- Learning approaches
  - E.g., Bayesian filtering – train algorithm on known spam, known ham – certain words happen more in spam (e.g. Viagra). Use word appearance as filter
How to evaluate anti-spam?

- It's easy to catch 100% of all spam!
  - Reject all messages

- It's easy to never misclassify good mail
  - Accept all messages

- Need to know false positives and false negatives
  - False positives are a big deal!

- Tricky because most algorithms can be tuned... no single number
Remainder of today: 
Spam economics (UCSD/ICSI)

• We tend to focus on the **costs** of spam
  – > 100 Billion spam emails sent **every day** [Ironport]
  – > $1B in direct costs – anti-spam products/services [IDC]
  – Estimates of indirect costs (e.g., *productivity*) 10-100x more

• But spam exists *only* because it is **profitable**
  • **Someone is buying!**

• Alternative
  – Attack underlying **economic support** for spam
History of the spam business model

- Direct Mail: origins in 19th century catalog business
  - Idea: send unsolicited advertisements to potential customers
  - Rough value proposition:
    Delivery cost < (Conversion rate * Marginal revenue)

- Modern direct mail (> $60B in US)
  - Response rate: ~2.5% (mean per DMA)
  - CPM (cost per thousand) = $250 - $1000

- Spam is qualitatively the same... just quantitatively cheaper.
First: how spam-based advertising works
Affiliate program structure

- **Division of labor**
  - **Affiliates** handle advertising (e.g., spam, SEO)
    - Independent contractors
    - Paid 25-60% commission depending on program
  - **Affiliate programs** handle backend
    - Payment processing, customer service, fulfillment
    - Sometimes hosting and domain registration

- **Why?**
  - Transfer of risk: innovation risk vs investment risk
  - Specialization lowers cost structure
Many affiliate programs...
“Leaked” ground truth data
Glavmed, Spamit, Rx-Promo

- 185M in gross revenue, 1+ million customers, 1.5+ million purchases, 2600+ affiliates

<table>
<thead>
<tr>
<th>Program</th>
<th>Period</th>
<th>Affiliates</th>
<th>Customers</th>
<th>Billed orders</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlavMed</td>
<td>Jan 2007 – Apr 2010</td>
<td>1,759</td>
<td>584,199</td>
<td>699,516</td>
<td>$81M</td>
</tr>
<tr>
<td>SpamIt</td>
<td>Jun 2007 – Apr 2010</td>
<td>484</td>
<td>535,365</td>
<td>704,169</td>
<td>$92M</td>
</tr>
<tr>
<td>RX-Promotion</td>
<td>Oct 2009 – Dec 2010</td>
<td>415</td>
<td>59,769 – 69,446</td>
<td>71,294</td>
<td>$12M</td>
</tr>
</tbody>
</table>
Heavy-tailed revenue

10% of affiliates account for ~80% of total program revenue
Where does the money come from?
Demand

- Purchasers are Western
  - US (75%), US + Europe/Canada/Australia (91%)

- Demand is primarily for ED drugs
  - 75-80% of orders
  - Long tail of drugs for chronic conditions
  - Abuse drugs high revenue
    (opiates, benzos, stimulants)

- Demand not even vaguely saturated... new customers joining at constant rate
Click Trajectories

- Click Trajectory project
  - Find “bottlenecks” in the spam “value chain” (i.e., what lets them make money)
  - Place where intervention could be most effective
    - Resources with largest impact on profitability
    - Highest switching cost for adversary

- Measure empirically
  - Resources needed to monetize each piece of spam
  - By playing the role of customer; at scale
    - Three domains: pharma, replica, software

Levchenko, Pitsillidis and an amazing cast of 13 others...

*Click Trajectories: End-to-end analysis of the Spam value chain, IEEE S &P, 2011*
Aug 1 -- Oct 31 2010
7 URL/Spam feeds + 5 botnet feeds
- 968M URLs
- 17M domains

Crawled domains for 98% of URLs in
- 1000s of browser instances
- Large IP address diversity

Hundreds of purchases
- Unique card # per order
- Full transaction data
Vpxl
#1 Dietary Supplement for Men
100% Natural Dietary Supplement
Contains 60 capsules
600+ orders later...
Result

- Most resources (domains, hosts, botnets, etc) have very **low replacement cost**

- Exception: merchant banks
  - Slow to establish, up front capital, holdback forfeiture; (loss can be larger than profit)
  - Handful of banks monetize all such abuse

- Research > Policy
  - Worked with brands, EOP IP Czar, Visa/MC
  - Undercover purchase/takedown regime

  Highly effective:
  - Microsoft effectively shuts down counterfeit software sales for > 18mos
  - Counterfeit pharma cut down (> 50% orgs close)

- European banks depart “risky” market
  - (now dominated by China, Azerbaijan, etc)
Qualitative Timeline

11/2011: Microsoft starts merchant complaint actions

11/20/2011: ATTENTION Dear advertisers, we are having problems with our bank accounts, which were suddenly frozen. We are forced to temporarily stop accepting OEM traffic.

12/23/2011: Remark by leading affiliate: "The sun is setting on the OEM era" (McCoy, Kreibich, Voelker and Savage, Priceless: the role of Payments in Abuse-advertised Goods, CCS 2012.)
"Right now most affiliate programs have a mass of declines, cancels and pendings, and it doesn't depend much on the program imho, there is a general sad picture, fucking Visa is burning us with napalm (for problematic countries, it's totally fucked, on a couple of programs you're lucky if you get 50% through)."
Summary

- Malware detection is complex
  - No foolproof way to tell if software is benign or not
  - Arms race where malware authors innovate to stay undetected

- Botnets have become a staple of e-crime (spam, phishing, identity theft, etc)
  - Couple large numbers of compromised machines with central command and control
  - Creates platform economy

- Cybercrime
  - Lots of ways to monetize access to someone’s computer (information, access, bandwidth, etc)
  - Click fraud, info stealers, ransomware, ddos, etc...

- Spam
  - Direct marketing meets botnets -> 100B spam/day
  - Significant profit center for criminals

- Sometimes most effective solutions aren’t technical
Next time

- Hardware security: Meltdown, Spectre and Rowhammer