Malware, con’t

CS 161: Computer Security

Prof. Vern Paxson

TAs: Jethro Beekman, Mobin Javed, Antonio Lupher, Paul Pearce & Matthias Vallentin

http://inst.eecs.berkeley.edu/~cs161/

April 18, 2013
Large-Scale Malware

• **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
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.
Large-Scale Malware

• **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?
  – How does worm get code to **automatically** run?

• **Botnet** = set of compromised machines (“bots”) under a common *command-and-control* (**C&C**)
  – Attacker might use a worm to get the bots, or other techniques; orthogonal to bot’s use in botnet
The Arrival of Internet Worms

• Worms date to Nov 2, 1988 - the *Morris Worm*
• *Way* ahead of its time
• Employed whole suite of tricks to infect systems …
  – *Multiple* buffer overflows
  – Guessable passwords
  – “Debug” configuration option that provided shell access
  – Common user accounts across multiple machines
• … and of tricks to find victims
  – Scan local subnet
  – Machines listed in system’s network config
  – Look through user files for mention of remote hosts
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* in many systems
  – Vulnerability & fix announced previous month
• Payload part 1: web site defacement
  – HELLO! Welcome to http://www.worm.com!
    Hacked By Chinese!
  – Only done if language setting = English
Payload part 2: check day-of-the-month and …
  – … 1\textsuperscript{st} through 20\textsuperscript{th} of each month: spread
  – … 20\textsuperscript{th} through end of each month: attack
    • Flooding attack against 198.137.240.91 …
    • … i.e., \texttt{www.whitehouse.gov}

Spread: via \textit{random scanning} of 32-bit IP address space
  – Generate pseudo-random 32-bit number; try connecting to it; if successful, try infecting it; repeat
  – Very common (but not fundamental) worm technique

Each instance used same random number seed
  – How well does the worm spread?

\textbf{Linear growth rate}
Code Red, con’t

• White House responds to threat of flooding attack by changing the address of www.whitehouse.gov
• Causes Code Red to die for date \( \geq 20^{th} \) of the month due to failure of TCP connection to establish.
  – Author didn’t carefully test their code - buggy!

• But: this time random number generator correctly seeded. Bingo!
The worm dies off globally!

Number of new hosts probing 80/tcp as seen at LBNL monitor of 130K Internet addresses

Measurement artifacts

The worm dies off globally!
Modeling Worm Spread

- Worm-spread often well described as *infectious epidemic*
  - Classic SI model: homogeneous random contacts
    - SI = Susceptible-Infectible
  - Model parameters:
    - N: population size
    - S(t): susceptible hosts at time t.
    - I(t): infected hosts at time t.
    - $\beta$: contact rate
      - How many population members each infected host communicates with per unit time
      - E.g., if each infected host scans 10 Internet addresses per unit time, and 2% of Internet addresses run a vulnerable server \( \Rightarrow \beta = 0.2 \)
  - Normalized versions reflecting relative proportion of infected/susceptible hosts
    - \( s(t) = \frac{S(t)}{N} \quad i(t) = \frac{I(t)}{N} \quad s(t) + i(t) = 1 \)
Computing How An Epidemic Progresses

• In continuous time:

\[ \frac{dI}{dt} = \beta \cdot I \cdot \frac{S}{N} \]

Increase in # infectibles per unit time
Total attempted contacts per unit time
Proportion of contacts expected to succeed

• Rewriting by using \( i(t) = \frac{I(t)}{N} \), \( S = N - I \):

\[ \frac{di}{dt} = \beta i(1 - i) \quad \Rightarrow \quad i(t) = \frac{e^{\beta t}}{1 + e^{\beta t}} \]

Fraction infected grows as a logistic
Fitting the Model to Code Red

Exponential initial growth

Growth slows as it becomes harder to find new victims!
Spread of Code Red, con’t

- Recall that # of new infections scales with contact rate $\beta$.
  \[
  \frac{dI}{dt} = \beta \cdot I \cdot \frac{S}{N}
  \]

- For a scanning worm, $\beta$ **increases** with $N$:
  - Larger populations infected more quickly!
    - More likely that a given scan finds a population member

- Large-scale monitoring finds 360K systems infected with Code Red on July 19:
  - Worm got them in 13 hours

- That night ($\Rightarrow 20^{th}$), worm dies due to DoS bug

- Worm actually managed to **restart itself** Aug. 1:
  - ... and each successive month for years to come!

*Emergent behavior*
Life Just Before Slammer
Life Just After Slammer

Sat Jan 25 06:00:00 2003 (UTC)
Number of hosts infected with Sapphire: 74866
Going Fast: *Slammer*

- Slammer exploited *connectionless* UDP service, 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*
The Usual Logistic Growth

Probes Recorded During Code Red's Reoutbreak

Number Seen in an hour

Hour of the day

# of scans

Predicted # of scans
Slammer’s Growth

What could have caused growth to deviate from the model?

Hint: at this point the worm is generating 55,000,000 scans/sec

Answer: the Internet ran out of carrying capacity! (Thus, $\beta$ decreased.) Access links used by worm completely clogged. Caused major collateral damage.
Big Worms: Conficker

Yearly Conficker A+B Population

2009 - 2010
Big Worms: Conficker

Yearly Conficker A+B Population

Unique IPs

Month

May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr

2012 - 2013
Stuxnet

- Discovered July 2010. (Released: Mar 2010?)
  - Multi-mode spreading:
    - Initially spreads via USB (virus-like)
    - Once inside a network, quickly spreads internally using Windows RPC
  - Kill switch: programmed to die June 24, 2012
  - Targeted SCADA systems
    - Used for industrial control systems, like manufacturing, power plants
  - Symantec: infections geographically clustered
    - Iran: 59%; Indonesia: 18%; India: 8%
Stuxnet, con’t

• Used four *Zero Days*
  – Unprecedented expense on the part of the author
• “Rootkit” for hiding infection based on installing Windows drivers with *valid digital signatures*
  – Attacker *stole* private keys for certificates from two companies in Taiwan
• Payload: *do nothing* …
  – … *unless* attached to particular models of frequency converter drives operating at 807-1210Hz
  – … like those made in Iran (and Finland) …
  – … and used to operate centrifuges for producing *enriched uranium for nuclear weapons*
Stuxnet, con’t

• Payload: do nothing …
  – … unless attached to particular models of frequency converter drives operating at 807-1210Hz
  – … like those made in Iran (and Finland) …
  – … and used to operate centrifuges for producing enriched uranium for nuclear weapons

• For these, worm would slowly increase drive frequency to 1410Hz …
  – … enough to cause centrifuge to fly apart …
  – … while sending out fake readings from control system indicating everything was okay …

• … and then drop it back to normal range
Israel Tests on Worm Called Crucial in Iran Nuclear Delay

By WILLIAM J. BROAD, JOHN MARKOFF and DAVID E. SANGER
Published: January 15, 2011

This article is by William J. Broad, John Markoff and David E. Sanger.

The Dimona complex in the Negev desert is famous as the heavily guarded heart of Israel’s never-acknowledged nuclear arms program, where neat rows of factories make atomic fuel for the arsenal.

Over the past two years, according to intelligence and military experts familiar with its operations, Dimona has taken on a new, equally secret role — as a critical testing ground in a joint American and Israeli effort to undermine Iran’s efforts to make a bomb of its own.

Behind Dimona’s barbed wire, the experts say, Israel has spun nuclear centrifuges virtually identical to Iran’s at Natanz, where Iranian scientists are struggling to enrich uranium. They say Dimona tested the effectiveness of the Stuxnet computer worm, a destructive program that appears to have wiped out roughly a fifth of Iran’s nuclear
Worm Take-Aways

• Potentially enormous reach/damage
  ⇒ Weapon

• Hard to get right

• Emergent behavior / surprising dynamics

• Remanence: worms stick around
  – E.g. Slammer still seen in 2013!

• Propagation faster than human response
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.
• Upon infection, new bot “phones home” to rendezvous w/ botnet command-and-control (C&C)
• Lots of ways to architect C&C:
  – Star topology; hierarchical; peer-to-peer
  – Encrypted/stealthy communication
• Botmaster uses C&C to push out commands and updates
Example of C&C Messages

1. Activation (report from bot to botmaster)
2. Email address harvests
3. Spamming instructions
4. Delivery reports
5. DDoS instructions
6. *FastFlux* instructions (rapidly changing DNS)
7. HTTP proxy instructions
8. Sniffed passwords report
9. IFRAME injection/report

From the “Storm” botnet circa 2008
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
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 crop 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

• 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 ...
Termed \textit{Bullet-proof hosting}

We offer a complaint-resistant hosting to host your sites, which are specified in mass mailings.

We decided to bring visitors to your web site through unsolicited mass emails? Wonderful idea! You certainly expect a boom visits. But! As in any ointment and then not pass without a spoon of tar ... Alas, but your wonderful site, shortly after the start of spam mail, will be closed due to flood of complaints from postal services. Is there a way to avoid these problems? Of course! Our complaint-resistant hosting simply ignores any complaints, all postal services, and you can be rest assured about the performance of their sites - they will not be closed. And you get new customers, expand their business and increase their sales and revenue, thanks to spam mailing lists.
**Obzoustownchivy hosting** is more expensive than usual, but you will have the full guarantee that your site no one ever closes, it will always be available to your customers!

<table>
<thead>
<tr>
<th>MINI PLAN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume disc</td>
<td>400 MB</td>
</tr>
<tr>
<td>Domains</td>
<td>1</td>
</tr>
<tr>
<td>Traffic *</td>
<td>Unlimited</td>
</tr>
<tr>
<td>FTP-access</td>
<td>there is</td>
</tr>
<tr>
<td>MySQL database</td>
<td>there is</td>
</tr>
<tr>
<td>Control panel</td>
<td>there is</td>
</tr>
<tr>
<td>COST</td>
<td>4 000 rub. / 1 month.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STARTER PLAN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume disc</td>
<td>500 mb</td>
</tr>
<tr>
<td>Domains</td>
<td>3</td>
</tr>
<tr>
<td>Traffic *</td>
<td>Unlimited</td>
</tr>
<tr>
<td>FTP-access</td>
<td>there is</td>
</tr>
<tr>
<td>MySQL database</td>
<td>there is</td>
</tr>
<tr>
<td>Control panel</td>
<td>there is</td>
</tr>
<tr>
<td>COST</td>
<td>5 000 rub. / 1 month.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BUSINESS PLAN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume disc</td>
<td>1000 mb</td>
</tr>
<tr>
<td>Domains</td>
<td>7</td>
</tr>
<tr>
<td>Traffic *</td>
<td>Unlimited</td>
</tr>
<tr>
<td>FTP-access</td>
<td>there is</td>
</tr>
<tr>
<td>MySQL database</td>
<td>there is</td>
</tr>
<tr>
<td>Control panel</td>
<td>there is</td>
</tr>
<tr>
<td>COST</td>
<td>7 000 rub. / 1 month.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREMIUM PLAN</th>
<th></th>
</tr>
</thead>
</table>
Fighting Bots / Botnets, con’t

• Approach #3: seize the domain name used for C&C
  – This is what’s currently often used, often to good effect …
• … Botmaster counter-measure?
  – Each day (say), bots generate large list of possible domain names using a Domain Generation Algorithm
    • Large = 50K, in some cases
  – Bots then try a random subset looking for a C&C server
    • Server signs its replies, so bot can’t be duped
    • Attacker just needs to hang on to a small portion of names to retain control over botnet
• This is becoming state-of-the-art …
• Counter-counter measure?
  – Behavioral signature: look for hosts that make a lot of failed DNS lookups (research)
How FBI, police busted massive botnet
12m zombie machines run by 3 admins

By John Leyden

Posted in Malware, 3rd March 2010 15:56 GMT

Analysis More details have emerged about a cybercrime investigation that led to the takedown of a botnet containing 12m zombie PCs and the arrest of three alleged kingpins who built and ran it.

As previously reported, the Mariposa botnet was principally geared towards stealing online login credentials for banks, email services and the like from compromised Windows PCs. The malware infected an estimated 12.7 million computers in more than 190 countries.

The Mariposa Working Group infiltrated the command-and-control structure of Mariposa to monitor the communication channels that relayed information from compromised systems back to the hackers who run the botnet. Analysis of the command system laid the groundwork for the December 2009 shutdown of the botnet, as well as shedding light on how the malware operated and provided a snapshot of the current state of the underground economy.
The botmasters made money by selling parts of the botnet to other cyercrooks,

Netkairo finally regained control of Mariposa and launched a denial of service attack against Defence Intelligence using all the bots in his control. This attack seriously impacted an ISP, leaving numerous clients without an Internet connection for several hours, including several Canadian universities and government institutions.
The botmasters made money by selling parts of the botnet to other cybercrooks, laundering stolen bank login credentials and credit card details via an international network of money mules. Search engine manipulation and serving pop-up ads was also part of the illegal business model behind the botnet.

The criminal gang behind Mariposa called themselves the DDP (Días de Pesadilla or Nightmare Days) Team. They nearly always connected to the Mariposa controlled servers from anonymous VPN (Virtual Private Network) services, preventing investigators from identifying their real IP addresses.

However when the December shutdown operation happened, the gang's leader, alias Netkairo, panicked in his efforts to regain control of the botnet. Netkairo made the fatal error of connecting directly from his home computer instead of using the VPN, leaving a trail of digital fingerprints that led to a series of arrests two months later.

Netkairo finally regained control of Mariposa and launched a denial of service attack against Defence Intelligence using all the bots in his control. This attack seriously impacted an ISP, leaving numerous clients without an Internet connection for several hours, including several Canadian universities and government institutions.
Once again, the Mariposa Working Group managed to prevent the DDP Team from accessing Mariposa. We changed the DNS records, so the bots could not connect to the C&C servers and receive instructions, and at that moment we saw exactly how many bots were reporting. We were shocked to find that more than 12 million IP addresses were connecting and sending information to the C&C servers, making Mariposa one of the largest botnets in history.

alleged lieutenants “Ostiator” and “Johnyloleante” have been charged with cybercrime offences. More arrests are expected to follow.

Under Spanish law suspects are not named at this stage of proceedings. Pedro Bustamante, senior research advisor at Panda Security, said: “Our preliminary analysis indicates that the botmasters did not have advanced hacking skills.

"This is very alarming because it proves how sophisticated and effective malware distribution software has become, empowering relatively unskilled cyber criminals to inflict major damage and financial loss." ©
Addressing The Botnet Problem

• What are our prospects for securing the Internet from the threat of botnets? What angles can we pursue?
• Angle #1: detection/cleanup
  – Detecting infection of individual bots hard as it’s the *defend-against-general-malware* problem
  – Detecting bot doing C&C likely a losing battle as attackers improve their sneakiness & crypto
  – Cleanup today lacks oomph:
    • *Who’s responsible?* … and do they *care?* (*externalities*)
    • Landscape could greatly change with different model of *liability*
• Angle #2: go after the C&C systems / botmasters
  – Difficult due to ease of Internet anonymity & complexities of international law
    • But: a number of recent successes in this regard
    • Including some via peer pressure rather than law enforcement (*McColo*)
  – One promising angle: policing domain name registrations
Addressing The Problem, con’t

• Angle #3: prevention
  – Bots require installing new executables or modifying existing ones
  – Perhaps via infection …
    • … or perhaps just via user being fooled / imprudent

• Better models?
• We could lock down systems so OS prohibits user from changing configuration
  – Sacrifices flexibility
  – How does this work for home users?
  – Can we leverage trusted kernels + white lists / code signing?
• Or: structure OS/browser so code runs with Least Privilege
  – Does this solve the problem?
  – Depends on how granular the privileges are … and how the decision is made regarding just what privileges are “least”
    • E.g., iTunes App Store model (vetting), Android model (user confirmation)