Foundations of Network and Computer Security

John Black

Lecture #12
Oct 5th 2004

CSCI 6268/TLEN 5831, Fall 2004

Announcements

- Project #0 due today
- Please sign a disclaimer if you have not already done so
  - I have them with me
- Project #1 has been assigned
  - See web page

Certs in My Browser

- Let's see how many certificates we have on this machine, just for fun…
  - This may not make it through to Tegrity, but we'll live dangerously for just a minute

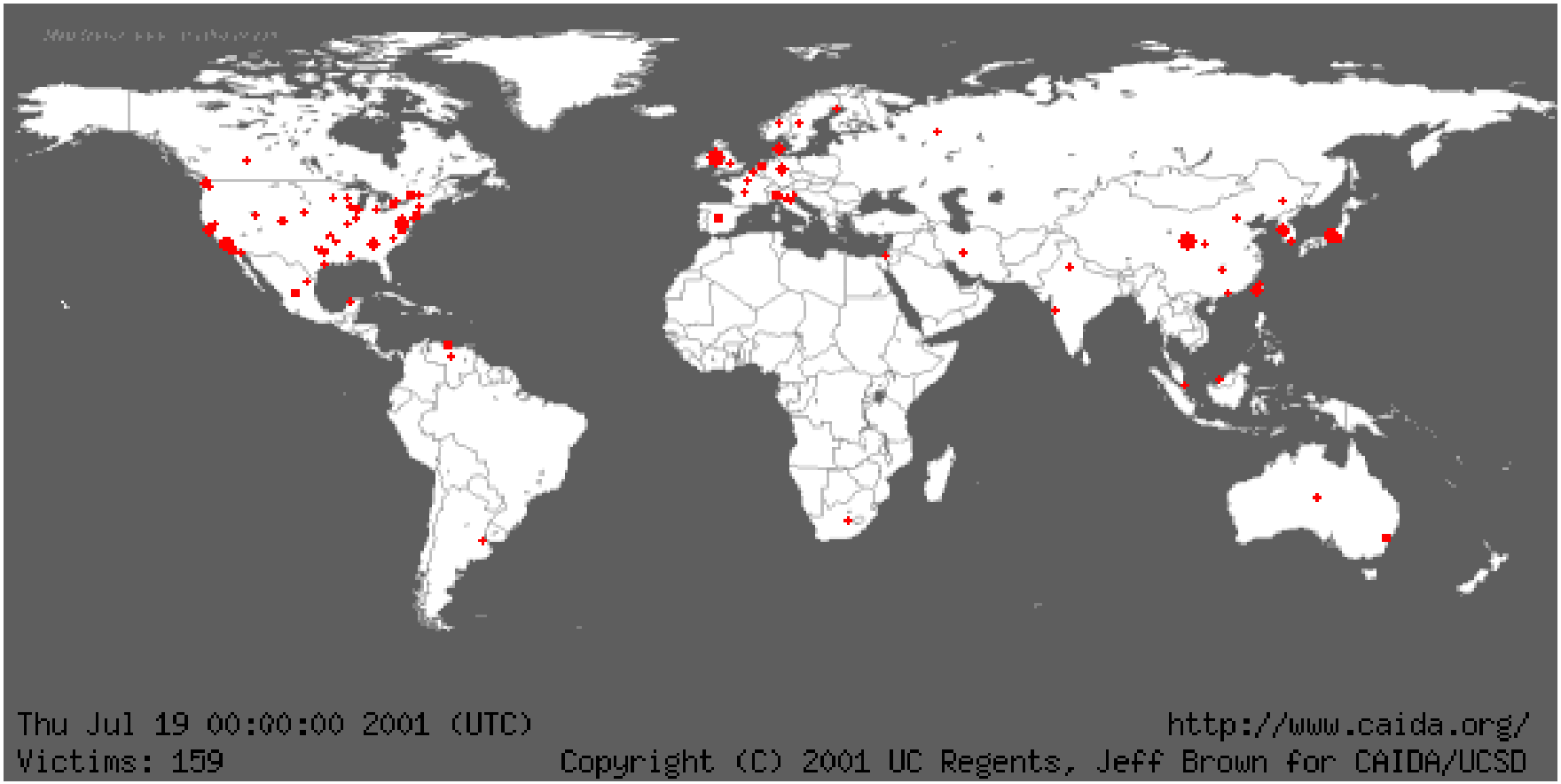
ILoveYou (2000)

- Clever technology, great social engineering
  - Subject: I love you
  - Body: Kindly check attached love letter from me
    - And message was from sender you know!
  - Attachment: LOVE-LETTER-FOR-YOU.TXT.vbs
    - Note the double-extension – VBS script
    - If you didn't have your OS set to show extensions, you'd just see LOVE-LETTER-FOR-YOU.TXT

It Gets Worse

- SirCam, Nimda, CodeRed, BadTrans
  - Nimda: very complex
    - Mostly spread via unpatched IIS servers, but also
      - Via email (attached EXE)
      - Browsing dubious web sites with unsecured browser
      - Using backdoors from other viruses (CodeRed II, eg)
      - Payload: back door access
  - Code Red (2001)
    - Tons of variants still around

Code Red Spread
(14 hrs, 350,000 hosts)



Code Red Payload

- Coordinated attack against www1.whitehouse.gov
  - Used hardcoded IP address
  - Checked to ensure port 80 was active first
  - Easy to stop this, and indeed the IP was moved before Code Red launched its payload, so no direct damage done
- windowsupdate.microsoft.com was infected too
  - Users got infected while trying to patch!
- First version used static seed for random()
  - Limited the number of IPs it generated
- Five days later this was fixed

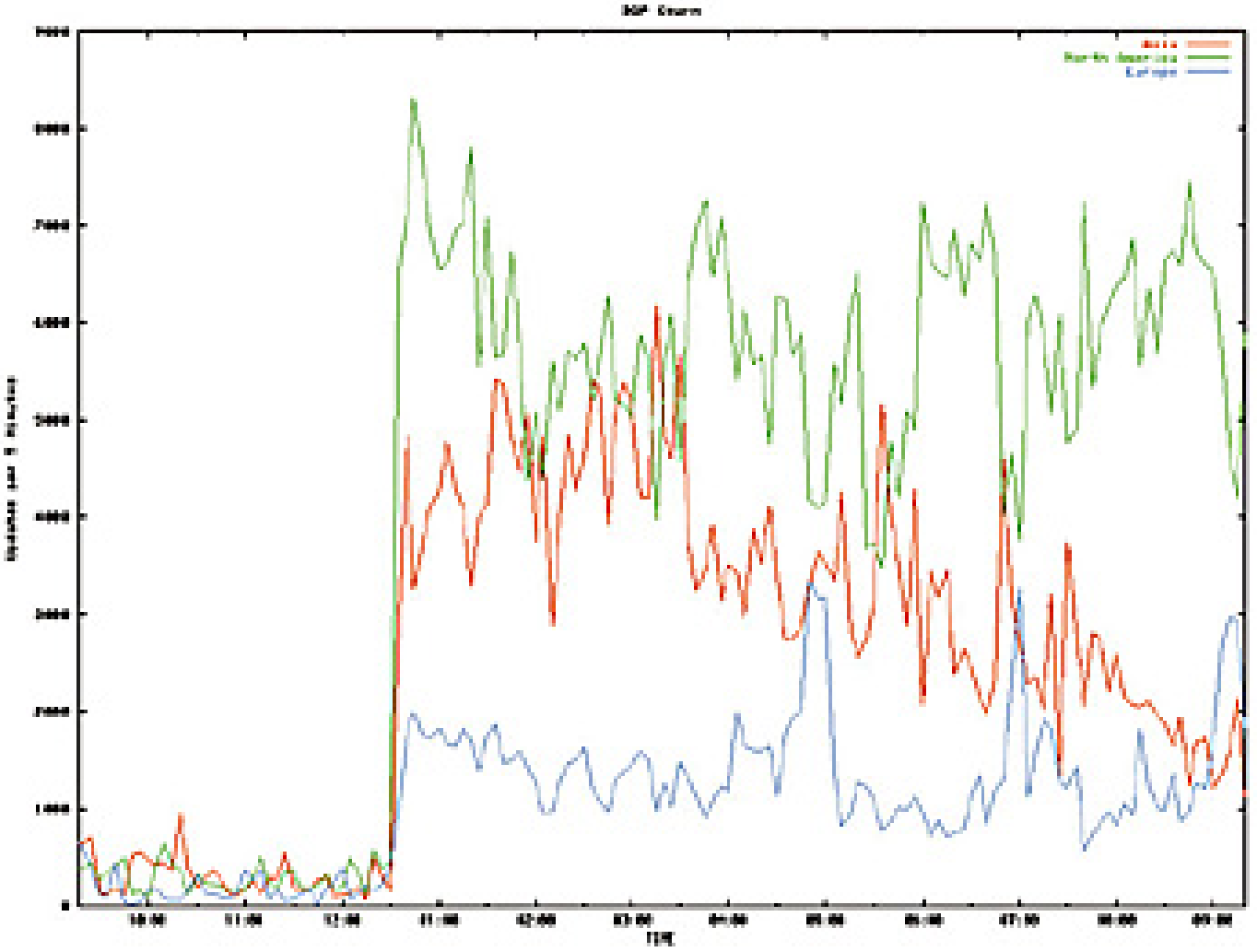
Code Red Details

- Spreads as a bad HTTP request.
- The IIS system mishandles the request, and instead executes the included packet with full permissions.
- The infected server then creates 99 threads which each attack random IP addresses
  - Random number generator works properly now
- This continues for the 1-19 of the month. On the 20-27 of the month, all the threads attack a specific IP at www.whitehouse.gov
  - Still see network traffic surges *today* from this worm
  - People don't patch!
- Defaces current pages on the server
  - Welcome to http://www.worm.com!
    Hacked by Chinese!

SQL/Slammer (2003)

- Exploits buffer overflow in MS SQL server
  - UDP traffic to port 1434
- Side-effect was DoS
  - Worm propagated so fast that it shut down many sites
  - Launched 12:30am EST victim numbers doubled every 8.5 seconds
  - By 12:45am, large pieces of the Internet were basically gone
    - 300,000 cable modem users in Portugal down
    - South Korea off the map (no cell phones or computer access)
  - Seattle 911 resorted to paper
  - Continental cancelled flights from Newark hub

SQL/Slammer
BGP Churn



Witty Worm (March 2004)

- Attacked a *security* product!
  - Internet Security Systems (ISS)
    - ISS RealSecure Network, RealSecure Server Sensor, RealSecure Desktop, and BlackICE
  - You can't even trust your security systems?!
- Vulnerability revealed by eEye Digital Security
  - Witty released *10 hours* after vulnerability was released
  - Destructive payload (deletes pieces of hard drive)

Flash Viruses

- Viruses can spread very fast
  - SQL/Slammer had only a 376 byte code size
  - No pause between propagation attempts
- Reading assignment
  - Read “How to 0wn the Internet in your Spare Time”
- A real problem
  - If you reinstall an old OS and attempt to download patches, you may be infected before you can patch!

Prevention

- Stay patched
  - windowsupdate.com
  - Linux patches
- Reduce network services to those needed
  - “Best block is not be there” – Mr. Miagi
  - Windows still comes with a ton of stuff turned on
  - SQL Slammer victims didn’t even know they were running an SQL server!
  - netstat –a
    - Might surprise you

Prevention (cont)

- Don’t open attachments unless you’re sure
  - Always run a virus scanner
  - Even Word docs are dangerous
- Don’t visit questionable web sites
  - Esp if your browser is set to low security levels
  - Javascript is evil

Trojans

- Malicious code hidden within another object
  - Email attachments can contain trojans
  - This is how many viruses spread
- Backdoor is usually considered as a synonym
  - Putting a backdoor into login.c qualifies

Thompson's Turing Award Lecture (1995)

- Thompson and Ritchie won the Turing award for creating Unix
- Thompson's is my favorite Turing award lecture
  - "Reflections on Trusting Trust"
  - Please read it (it's short)
- His lecture has three stages
  - Stage I: a "Quine"
  - A Quine is a program which outputs its own source code

A Quine in C

```
char*f="char*f=%c%s%c;main() {printf(f,34,f,34,10);}%c";
    main(){printf(f,34,f,34,10);}
```

- We printf the string f, inserting f into itself as a parameter
  - Yow!
- We could attach any extra code we like here
- File this away in your head for now: we can write a program which outputs its own source code

Thompson, Stage II

- Note that a C compiler is often written in C
  - Kind of strange chicken-and-egg problem
  - How to bootstrap
- Interesting “learning behavior”
  - You add a feature, compile compiler with itself, then it “knows” the feature
- Once you get a rudimentary compiler written, it can be arbitrarily extended

Thompson, Stage III

- Add a backdoor to login.c
  - Allow valid passwords *plus* some “master” password
  - Note that this would be caught soon enough because it exists in the login.c source code
- Ok, so be sneakier
  - Add code in cc.c (the C compiler) to add the backdoor to login.c whenever compiling login.c
  - Add self-replicating code to the C compiler to reproduce itself plus the login.c backdoor!

Implementing the Trojan

- Now compile login.c
  - Compiler adds the backdoor
- Compile cc.c
  - Compiler sees that it's compiling itself and self-replicating code runs to ensure login.c trojan and cc.c trojan are compiled into cc binary
- Now remove all this new code from cc.c
  - Back door exists only in binary!
  - login.c and cc.c will continue to have trojan even after infinite recompiles

Moral of the Story

- The amount of cleverness we haven't even thought of yet is scary
  - We're probably never going to have completely secure computers and networks
  - The most we can hope for is "best effort" from those we trust and from ourselves
  - It's going to be an eternal battle between us and the criminals