# Inoculating SSH Against Address-Harvesting Worms

**Stuart E. Schechter**
**Information Assurance Group**
**MIT Lincoln Laboratory**

**Jaeyeon Jung**
**MIT CSAIL**

This work is sponsored by the Department of Defense under the Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

# A World Without Scanning Worms

- Assume your network is immune to scanning worms
  - Your IP space is sparse
  - Scanning is almost certain to be detected before infection can spread
  - As for Jung, Paxson, Schechter, Staniford, Twycross, Weaver, and Williamson…

Unemployed

# Smart Worms Don’t Scan

## Why scan when infected host has info needed?

- Port 80 (HTTP)
  - Check web browser’s file cache
  - Check addresses in cookie files
  - Perform random google searches
- Port 25 (Mail)
  - Search mail archives

***Services exposed to outside attack anyway.***

***Critical data usually stored/audited elsewhere.***

# *Smart Worms Don’t Scan:* Address-Harvesting Worms Aren’t New

- Morris’ “Internet Worm” found target hosts in
  - `.rhosts`
  - `.forward`
  - `hosts.equiv`
- Exploited buffer overflow (`fingerd`)
- Exploited format string vulnerability (`sendmail`)

# Morris' Worm

- **Could spread without software flaws**
  - **Cracked passwords on local host (dictionary attack)**
  - **Use cracked <user/password> pair to `rsh` to remote hosts**

# Are More Harvesting Worms Coming?

- The good news
  - Morris is out of the business
- The bad news
  - When scan-detection is deployed, worm writers will work harder
  - Dictionary attack worms coming back into vogue
    - Lovgate, Deloader, Gaobot
    - Attack online, without harvesting usernames/passwords
  - `rsh` has been replaced by SSH…

# SSH is Ideal for Address-Harvesting

- For each user, the ssh client keeps a list that
  - contains the name of every host the user has logged into,
  - is kept chronological order of host discovery (most recent are most likely to still be active),
  - and is conveniently titled “known_hosts”
- Config files also may contain hostnames
- Server logs store user/clienthost pairs

***Why scan when targets are on the menu?***

# SSH Introduces Identity Keys

- Instead of using a password…
  - Add public key to hosts you log into
  - Use secret key to authenticate
  - Passwords/agents protect secret key (please!)

- Worms love identity keys
  - One cracked ID key yields many new targets
  - Password-protecting keys is optional
    - » If password, worm can still try dictionary attack
  - Keys can be scooped out of running agents
  - Root not needed if permissions set incorrectly

# SSH Worms: Impact

- Most worms/viruses attack user machines
  - Low/moderate impact
- SSH is used to access & administer…
  - Transaction processing systems
  - Databases & data stores
  - Security devices
  - Just about every other back-office UNIX system
- Often used to tunnel through firewalls
- SSH encryption prevents content inspection

# Fixing SSH: A Strategy

- Prevent worms from harvesting addresses
- Worms can still scan
- We know how to detect scanning worms
  - Weaver, Staniford, Paxson [USENIX Sec 2004]
  - Jung, Schechter, Berger [RAID 2004]

# Hiding Addresses: known_hosts

- The `known_host` file is needed when
  - ssh must check if <key,hostname> pair matches known <key,hostname> pair in file
  - Add new <key,hostname> pairs if needed
- By comparison, `/etc/passwd` needed when
  - Host must check if <username,password> matches known <username,password> pair
- Do we store passwords in plaintext?

# Fixing known_hosts and Config Files

- Hostnames are DNS names or IP addresses
  - host-13.somedomain.com
  - 147.168.9.42
- Don’t store hostname, instead…
  - Generate random salt
  - Store <salt,hash(salt,hostname)> as <s,h>
- Does hostname match known_hosts entry?
  - Read s,h from file entry
  - Check if h=hash(s,hostname)

# Hiding Addresses: Write-Only Logs

- Vulnerable hosts should be
  - Able to write log entries
  - Unable to read log entries
- Use public key cryptography
  - First entry sets session key
  - Encrypt $K_0$ with public key, write to log
  - Encrypt log entry i with key $K_i = \mathrm{hash}(K_{i-1})$
  - Calculate $k_{i+1} = h(k_i)$ and discard $k_i$
- Private key can decrypt $K_0$

For more advanced techniques, see Schneier and Kelsey (1999) and others

# Fixing OpenSSH

- We updated OpenSSH to
  - Hash `known_hosts`
  - Encrypt logs

- Our experience with OpenSSH code
  - Sparsely documented
  - Uses OpenSSL Crypto library
    - » APIs aren't fully documented (code is worse)
    - » Caller must know correct buffer size when calling API (no max length to write parameter)
  - Hard to believe folks are looking at and auditing this code

# Additional Inoculations

## *Diversify to break worm's assumptions*

- Add second password after login
- Use custom shells to limit access
  - Rename key commands
  - Change format of commands
    - » `please rm -f thanks`
- Look for commands that appear to be scripts
  - Key stroke timing

# Concluding remarks

- SSH failed to learn from past
  - Morris worm harvested addresses in 1988
  - Password files encrypted in 1970s
  - SSH released with plaintext `known_hosts` in 1995
- The threat is significant
  - SSH protects mission critical systems
- Fixes are painless
  - Easy to implement
  - Few users will know the difference

# Parallel & Future Work

- Harvest tool
  - Searches disks for all domain names / IPs
  - Protocol guessing heuristics
  - Collects statistics (hashed for privacy)
  - Compares between hosts

- known_host measurement
  - Collect known_host files
  - Analyze topology
  - Model potential spread