Segurança de Sistemas e dados (MSI 2021/2022)

Aula 8

Rolando Martins

DCC – FCUP

Slides Adaptados do Prof. Manuel Eduardo Correia

Future of Malware

- Polymorphic and metamorphic malware
- Fast replication/Warhol worms
- Flash worms, Slow worms, etc.
- Future is bright for malware
  - Good news for the bad guys…
  - …bad news for the good guys
- Future of malware detection?

Polymorphic Malware

- Polymorphic worm (usually) encrypted
- New key is used each time worm propagates
  - The encryption is weak (repeated XOR)
  - Worm body has no fixed signature
  - Worm must include code to decrypt itself
  - Signature detection searches for decrypt code
- Detectable by signature-based method
  - Though more challenging than non-polymorphic…

Metamorphic Malware

- A metamorphic worm mutates before infecting a new system
- Such a worm can avoid signature-based detection systems
- The mutated worm must do the same thing as the original
- And it must be “different enough” to avoid detection
- Detection is currently unsolved problem

Metamorphic Worm

- To replicate, the worm is disassembled
- Worm is stripped to a base form
- Random variations inserted into code
  - Rearrange jumps
  - Insert dead code
  - Many other possibilities
- Assemble the resulting code
- Result is a worm with same functionality as original, but very different signature

Warhol Worm

- "In the future everybody will be world-famous for 15 minutes" — Andy Warhol
- A Warhol Worm is designed to infect the entire Internet in 15 minutes
- Slammer infected 250,000 systems in 10 minutes
  - "Burned out" bandwidth
  - Slammer could **not** have infected all of Internet in 15 minutes — too bandwidth intensive
- Can a worm do "better" than Slammer?

Warhol Worm

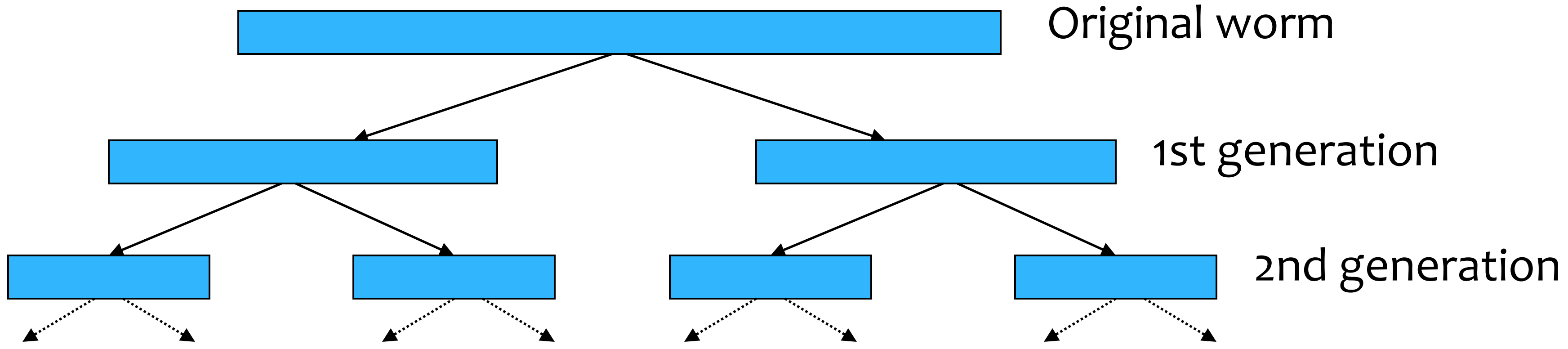
- One approach to a Warhol worm…
- Seed worm with an initial **hit list** containing a set of vulnerable IP addresses
  - Depends on the particular exploit
  - Tools exist for finding vulnerable systems
- Each successful initial infection would attack selected part of IP address space
  - Slammer generated random IP addresses
- Could infect entire Internet in 15 minutes?

Flash Worm

- Possible to do "better" than Warhol worm?
- Can entire Internet be attacked in < 15 min?
- Searching for vulnerable IP addresses is slow part of any worm attack
- Searching might be bandwidth limited
  - Like Slammer
- A "flash worm" is designed to infect entire Internet almost instantly

Flash Worm

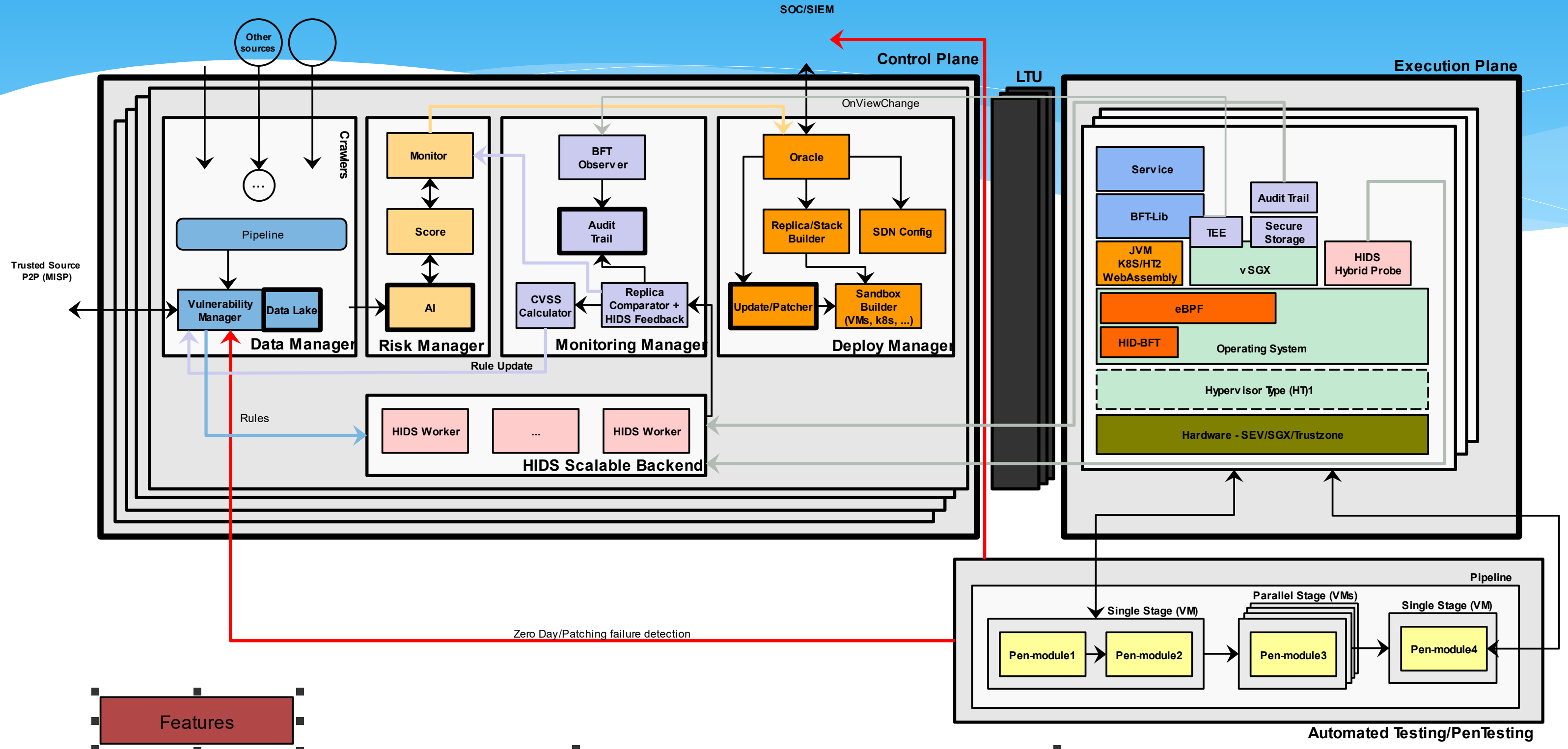
- Predetermine **all** vulnerable IP addresses
  - Depends on the particular exploit
- Embed all known vulnerable addresses in worm
- Result is a huge worm (perhaps 400KB)
- Whenever the worm replicates, it splits the vulnerable address space
- Virtually no wasted time or bandwidth!



Flash Worm

- Estimated that ideal flash worm could infect the entire Internet in **15 seconds!**
- Much faster than humans could respond
- **A conjectured defense against flash worms**
  - Deploy many "personal IDSs"
  - Master IDS watches over the personal IDSs
  - When master IDS detects unusual activity, lets it proceed on a few nodes, blocks it elsewhere
  - If sacrificial nodes adversely affected, attack is prevented almost everywhere

Skynet,
Tadeu Freitas' PhD



Features

- Features Zero Trust architecture
- Proactive Self-Healing/Reconfiguration in the presence of Bizantine Failures
- Possible Online Patching - no downtime
- Attaches corporate SOC
- Additional "Exploits Data Lake" as a service
- Artificial intelligence for detecting malicious activies
- Self-Testing

Briareos: A Modular Framework for Elastic Intrusion Detection and Prevention

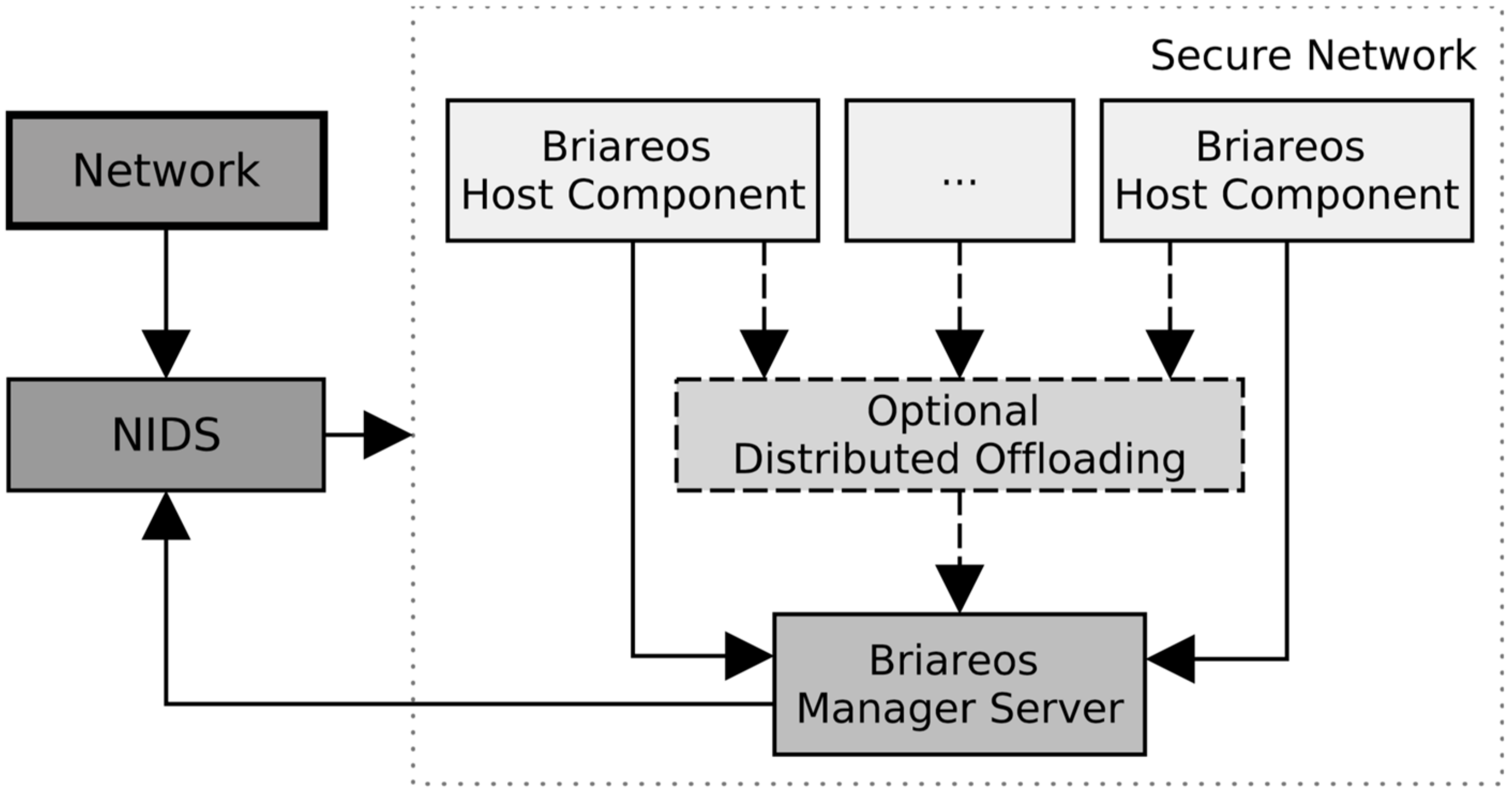
- Current intrusion detection systems are mainly based on signature detection running on top of highly optimized parallel engines
- There is no unified intrusion detection architecture that is able to support them under a unified framework
- Unknown attack vectors are hard to detect
- Given the highly dynamical nature of networked attacks, there is a growing shift to a multi- disciplinary approach as a way to uncover novel algorithms, systems and techniques
- Lack of mechanisms for heavy processing

Briareos: A Modular Framework for Elastic Intrusion Detection and Prevention

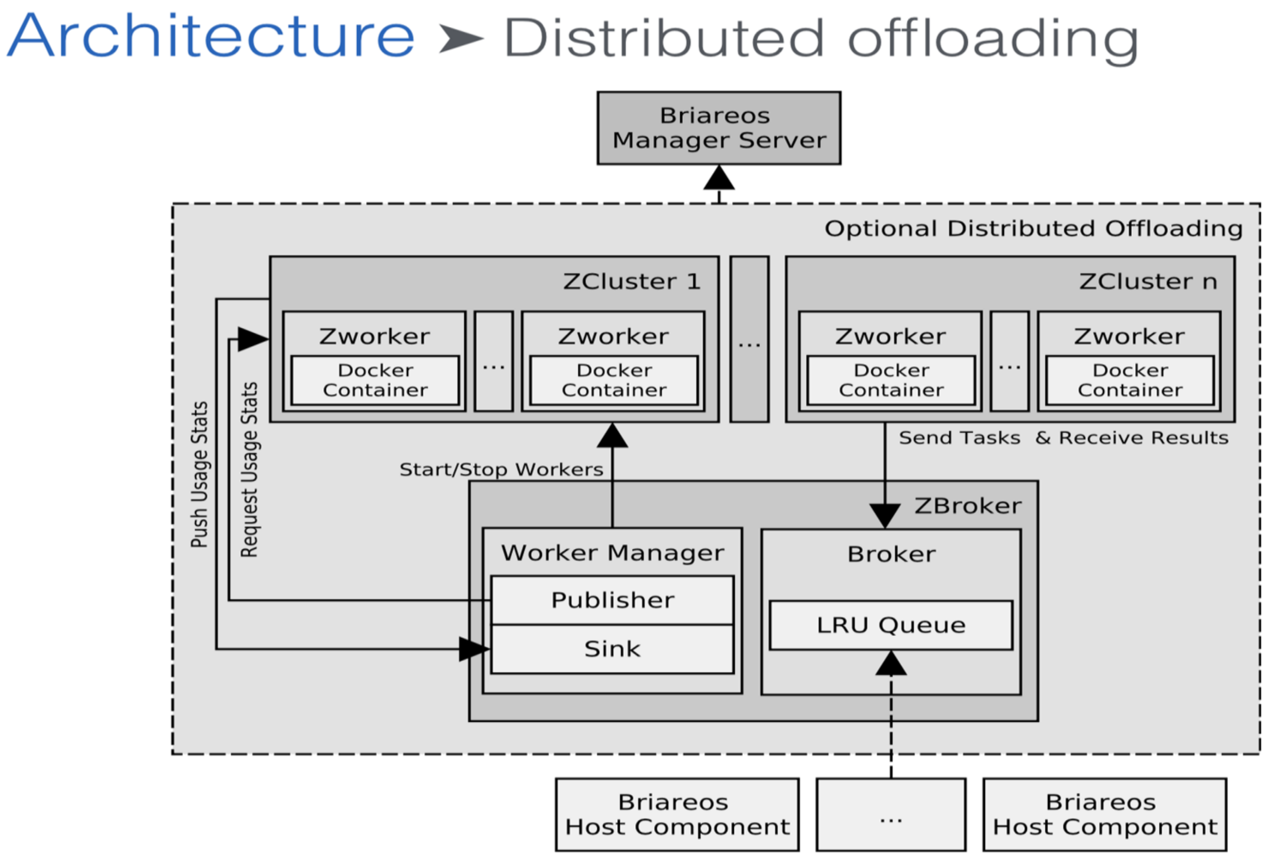
- Easy packet dissection and full control over the processing flow
- Module extensible framework
- Unknown attack prevention and detection
- Correlation: network traffic / operating system
- Distributed system for heavy processing
- Automatic generation of detection patterns
- Intelligence sharing

Briareos: A Modular Framework for Elastic Intrusion Detection and Prevention

Architecture ➤ Overview



Briareos: A Modular Framework for Elastic Intrusion Detection and Prevention

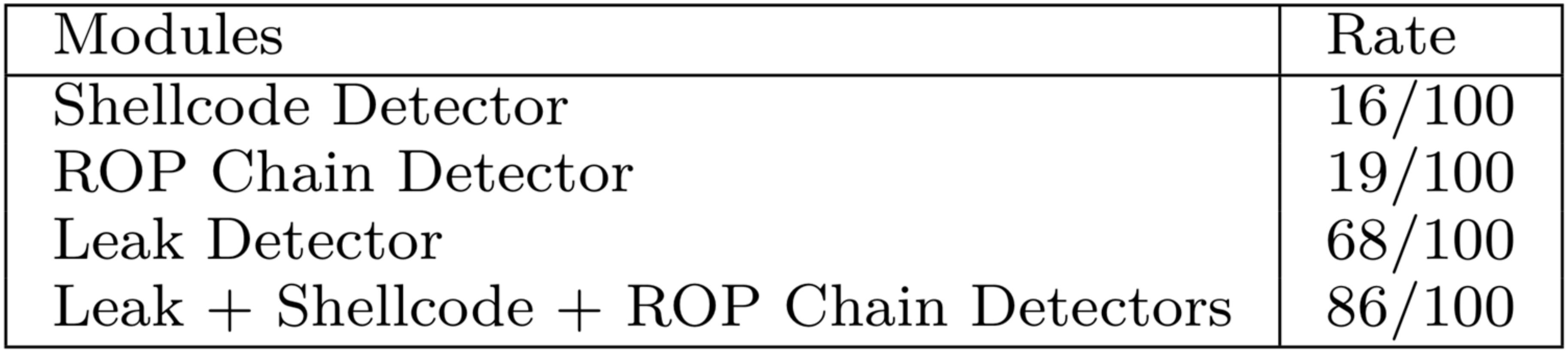


Briareos: A Modular Framework for Elastic Intrusion Detection and Prevention

- **Input pipeline:** PID fetcher, *shellcod*e detector and ROP chain detector

- **Output pipeline:** PID fetcher, memory leak detector, shell detector

Briareos: A Modular Framework for Elastic Intrusion Detection and Prevention



Modules	Rate
Shellcode Detector	16/100
ROP Chain Detector	19/100
Leak Detector	68/100
Leak + Shellcode + ROP Chain Detectors	86/100

Table I: Exploit Detection Results.

Computer Infections

- Analogies are made between computer viruses/worms and biological diseases
- There are differences
  - Computer infections are much quicker
  - Ability to intervene in computer outbreak is more limited (vaccination?)
  - Bio disease models often not applicable
  - "Distance" almost meaningless on Internet
- But there are some similarities…

Computer Infections

- Cyber "diseases" vs biological diseases
- One similarity
  - In nature, too few susceptible individuals and disease will die out
  - In the Internet, too few susceptible systems and worm might fail to take hold
- **One difference**
  - In nature, diseases attack more-or-less at random
  - Cyber attackers select most "desirable" targets
  - Cyber attacks are more focused and damaging

Viruses

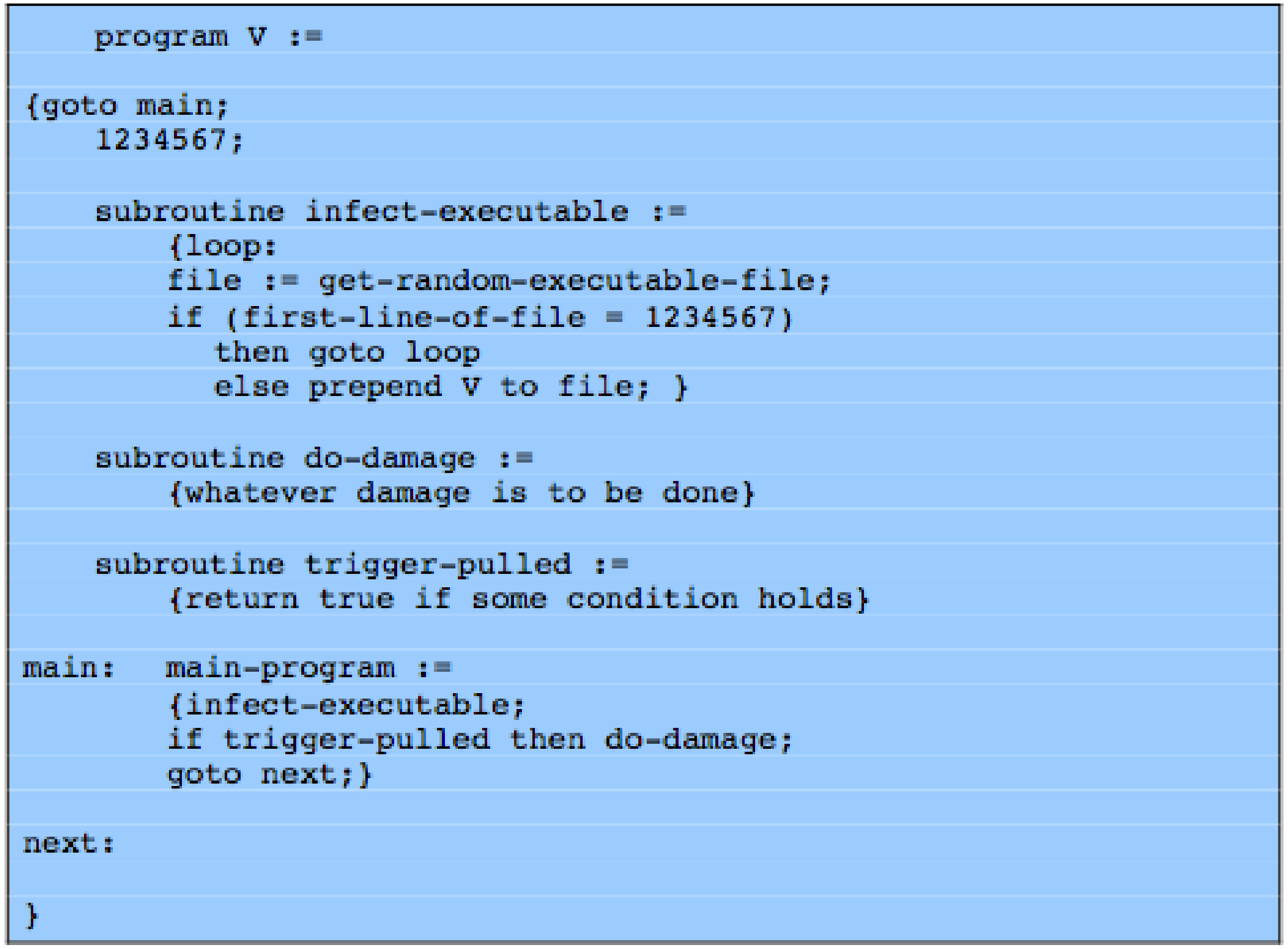
- piece of software that infects programs
  - modifying them to include a copy of the virus
  - so it executes secretly when host program is run
- specific to operating system and hardware
  - taking advantage of their details and weaknesses
- a typical virus goes through phases of:
  - dormant
  - propagation
  - triggering
  - execution

Virus Structure

- components:
  - infection mechanism - enables replication
  - trigger - event that makes payload activate
  - payload - what it does, malicious or benign
- prepended / postpended / embedded
- when infected program invoked, executes virus code then original program code
- can block initial infection (difficult)
- or propagation (with access controls)

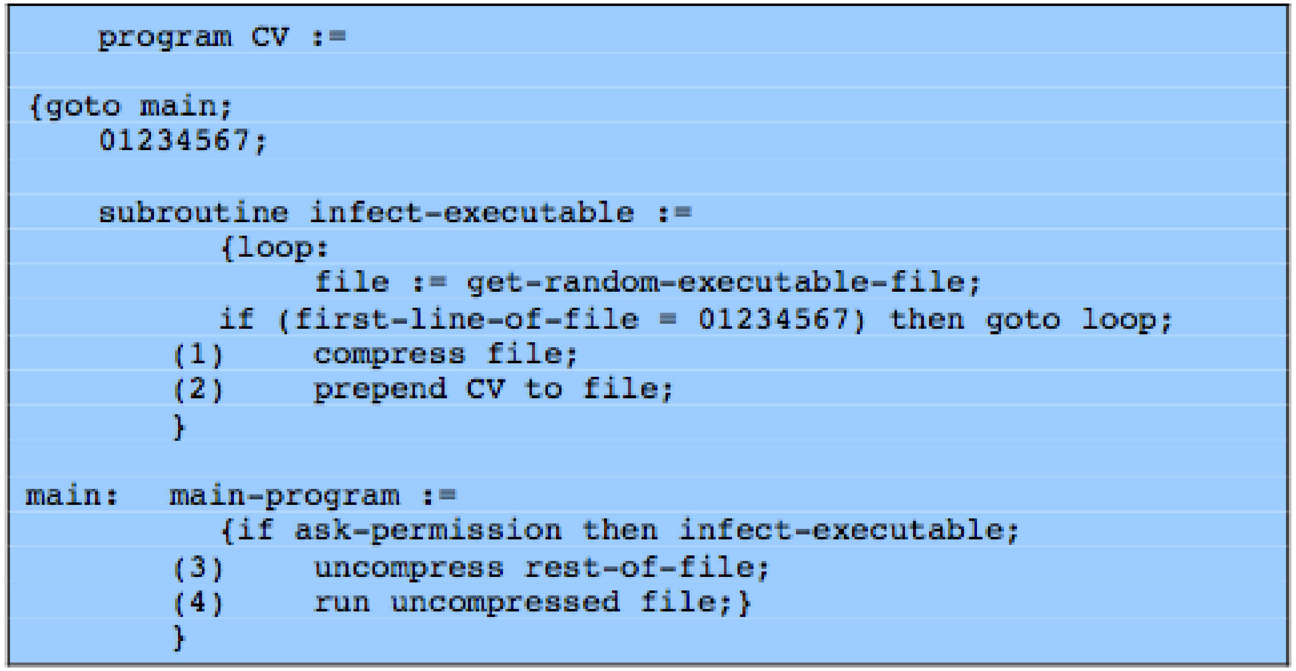
http://struppigel.blogspot.com/2017/11/file-infection-strategies.html

Virus Structure

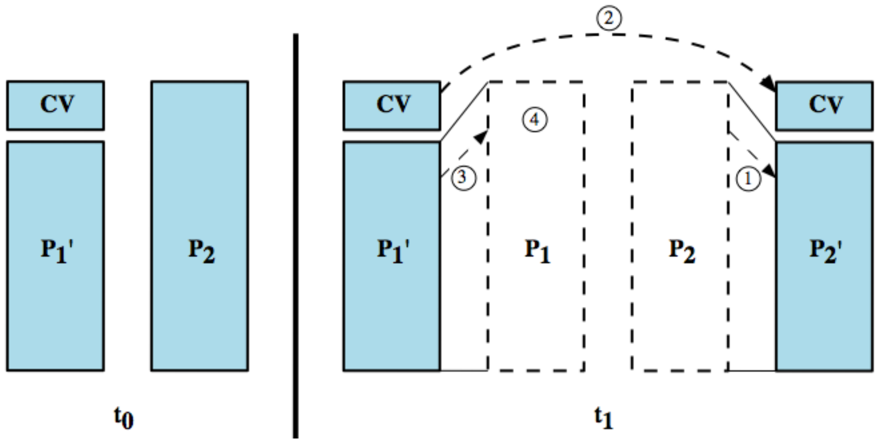


```
    program V :=

{goto main;
    1234567;

    subroutine infect-executable :=
        {loop:
        file := get-random-executable-file;
        if (first-line-of-file = 1234567)
           then goto loop
           else prepend V to file; }

    subroutine do-damage :=
        {whatever damage is to be done}

    subroutine trigger-pulled :=
        {return true if some condition holds}

main:   main-program :=
        {infect-executable;
        if trigger-pulled then do-damage;
        goto next;}

next:

}
```

Compression Virus



```
    program CV :=

{goto main;
    01234567;

    subroutine infect-executable :=
          {loop:
               file := get-random-executable-file;
          if (first-line-of-file = 01234567) then goto loop;
        (1)    compress file;
        (2)    prepend CV to file;
        }

main:   main-program :=
          {if ask-permission then infect-executable;
        (3)    uncompress rest-of-file;
        (4)    run uncompressed file;}
        }
```



Virus Classification

- boot sector
- file infector
- macro virus
- encrypted virus
- stealth virus
- polymorphic virus
- metamorphic virus

Macro Virus

- became very common in mid-1990s since
  - platform independent
  - infect documents
  - spread easily
- exploit macro capability of office apps
  - executable program embedded in office doc
  - often a form of Basic
- more recent releases include protection
- recognized by many anti-virus programs

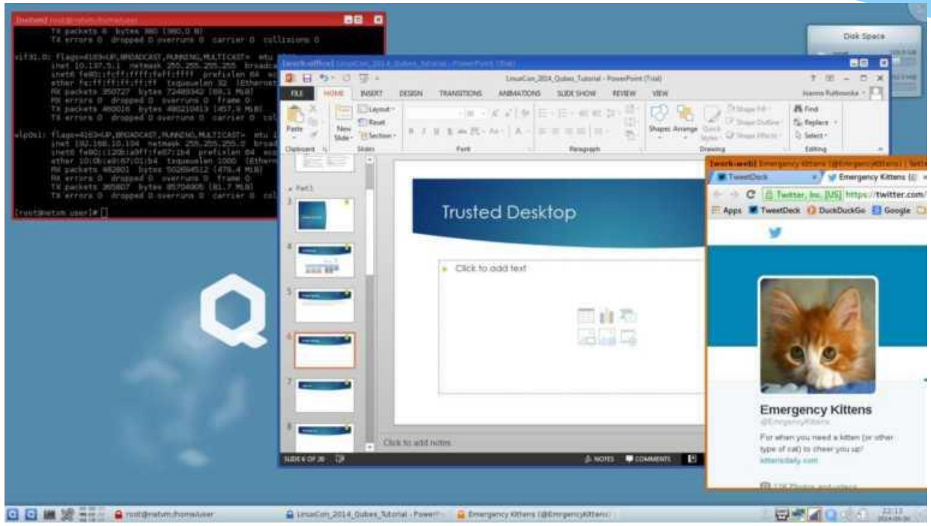
E-Mail Viruses

- Widely WideSpread
- e.g. Melissa
  - exploits MS Word macro in attached doc
  - if attachment opened, macro activates
  - sends email to all on users address list
  - and does local damage
- then saw versions triggered reading email
- hence much faster propagation

Virus Countermeasures

- prevention - ideal solution but difficult
- realistically need:
  - detection
  - identification
  - removal
- if detect but can’t identify or remove, must discard and replace infected program

Qubes
Security oriented OS



```
        TX packets 0  bytes 880 (880.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vif31.0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu
        inet 10.137.5.1  netmask 255.255.255.255  broadca
        inet6 fe80::fcff:ffff:feff:ffff  prefixlen 64
        ether fe:ff:ff:ff:ff:ff  txqueuelen 32  (Ethernet
        RX packets 350727  bytes 72480942 (69.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 480016  bytes 480210413 (457.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  col

wlp0s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1
        inet 192.168.10.104  netmask 255.255.255.0  broad
        inet6 fe80::120b:a9ff:fe67:1b4  prefixlen 64
        ether 10:0b:a9:67:01:b4  txqueuelen 1000  (Ethern
        RX packets 482801  bytes 502694512 (479.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 395807  bytes 85704905 (81.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  col

[root@netvm user]# 
```

Trusted Desktop

Click to add text

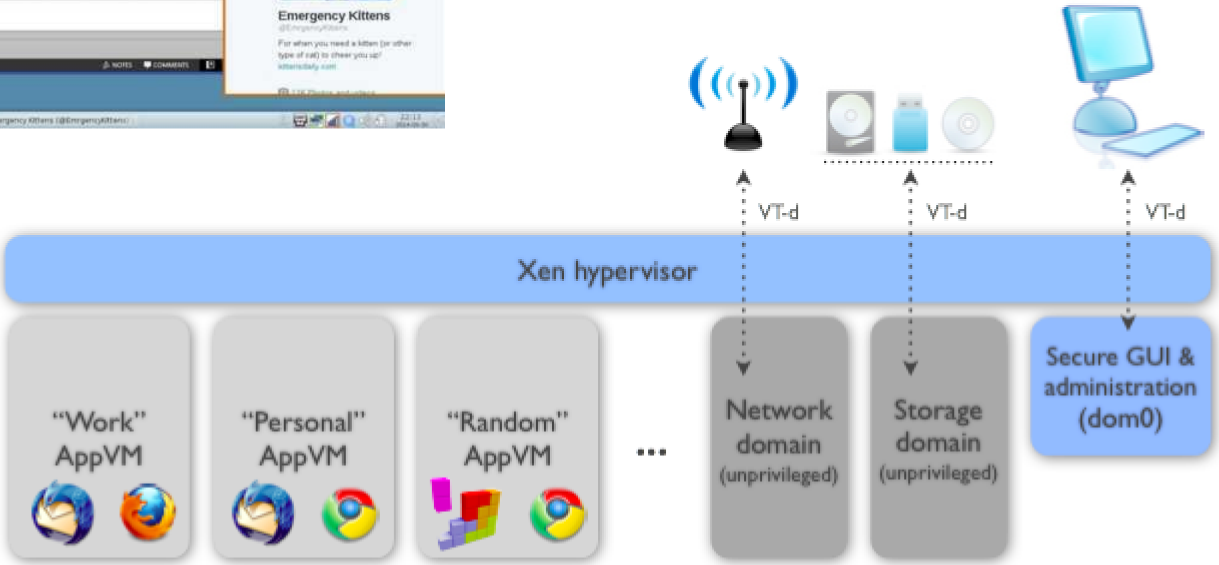
Click to add notes

Emergency Kittens

@EmergencyKittens

For when you need a kitten (or other type of cat) to cheer you up!

kittensdaily.com



VT-d
VT-d
VT-d
Xen hypervisor
"Work" AppVM
"Personal" AppVM
"Random" AppVM
...
Network domain (unprivileged)
Storage domain (unprivileged)
Secure GUI & administration (dom0)

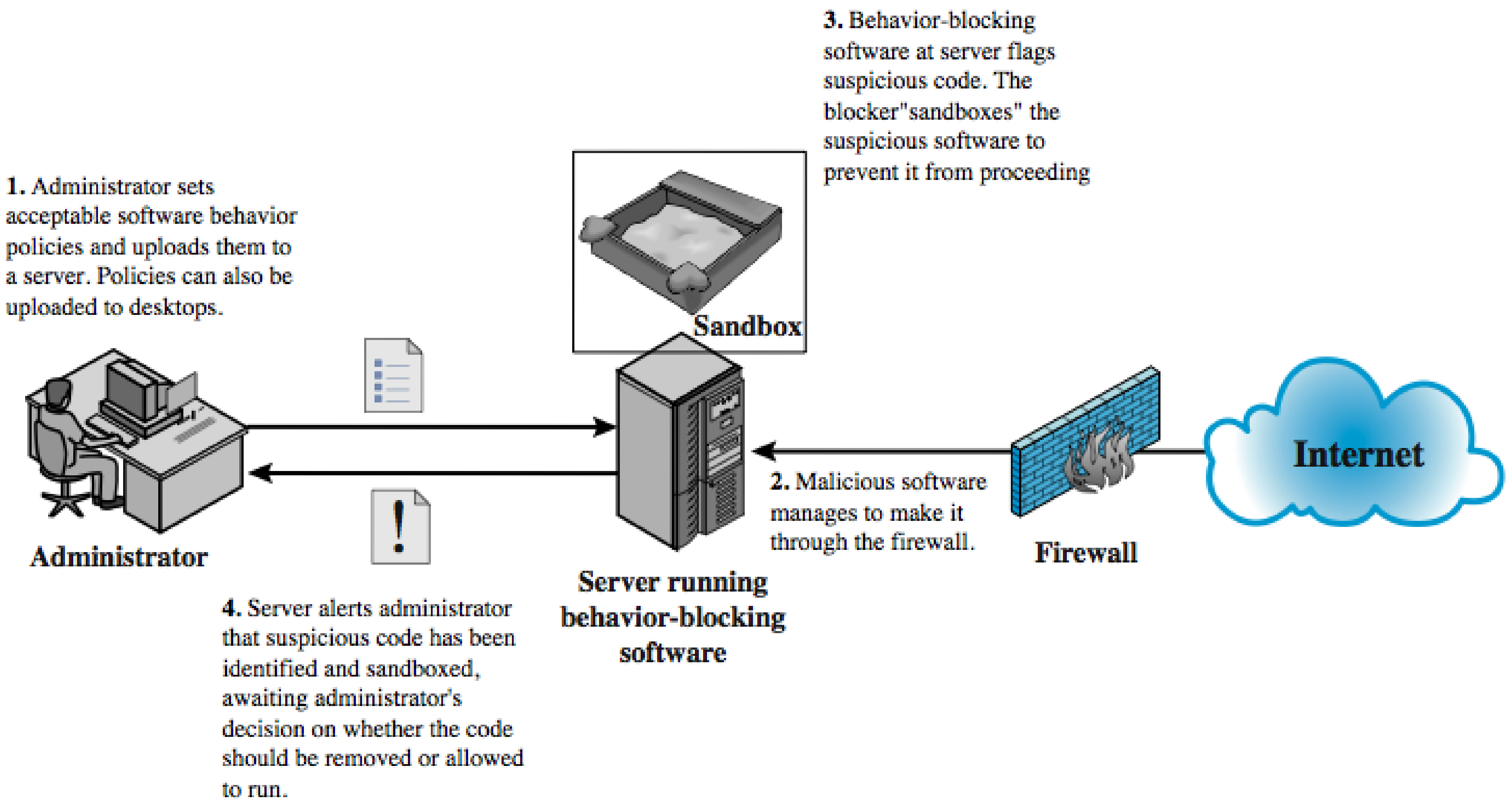
Anti-Virus Evolution

- virus & antivirus tech have both evolved
- early viruses simple code, easily removed
- as become more complex, so must the countermeasures
- generations
  - first - signature scanners
  - second - heuristics
  - third - identify actions
  - fourth - combination packages

Generic Decryption (GD)

- runs executable files through GD scanner:
  - CPU emulator to interpret instructions
  - virus scanner to check known virus signatures
  - emulation control module to manage process
- lets virus decrypt itself in interpreter
- periodically scan for virus signatures
- ISSUE - long time to interpret and scan
  - tradeoff chance of detection vs time delay

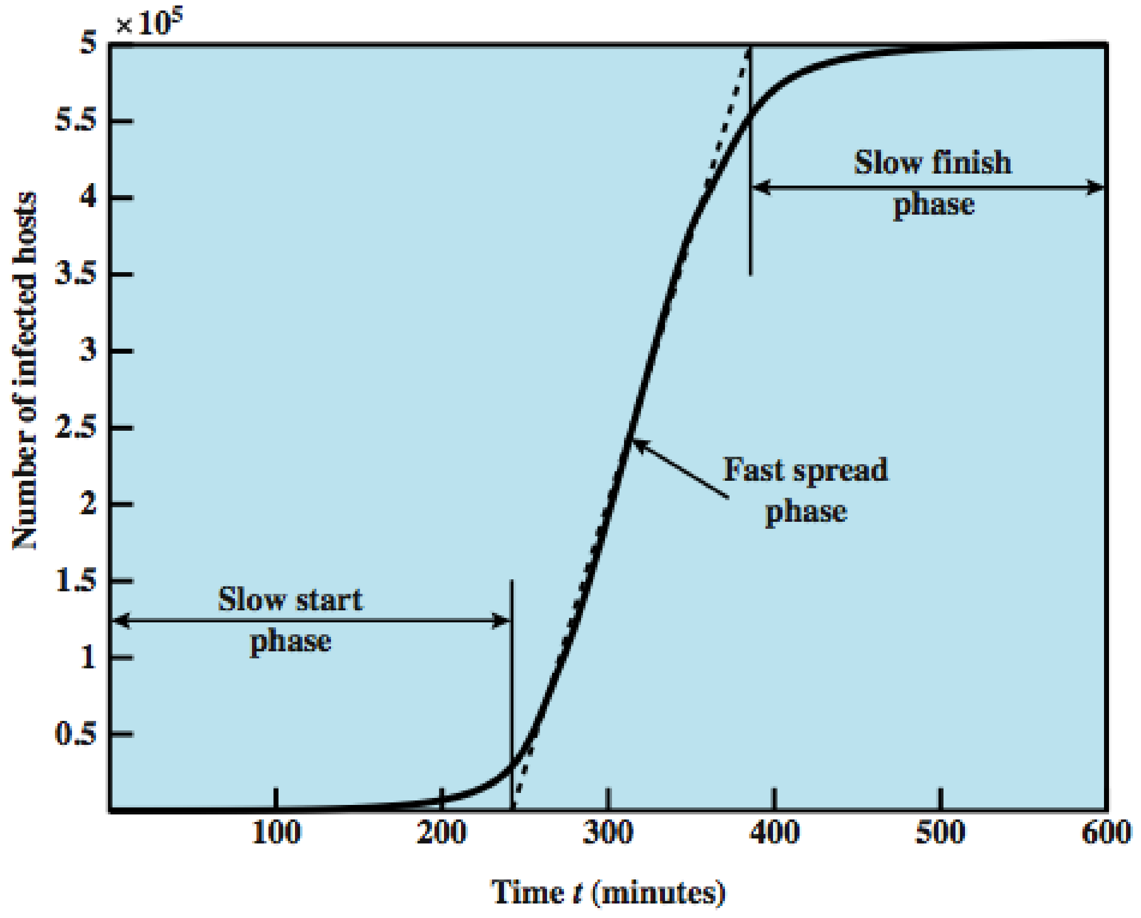
Behavior-Blocking Software


3. Behavior-blocking software at server flags suspicious code. The blocker"sandboxes" the suspicious software to prevent it from proceeding
Sandbox
1. Administrator sets acceptable software behavior policies and uploads them to a server. Policies can also be uploaded to desktops.
Internet
2. Malicious software manages to make it through the firewall.
Firewall
Administrator
Server running behavior-blocking software
4. Server alerts administrator that suspicious code has been identified and sandboxed, awaiting administrator's decision on whether the code should be removed or allowed to run.

Worms

- replicating program that propagates over net
  - using email, remote exec, remote login
- has phases like a virus:
  - dormant, propagation, triggering, execution
  - propagation phase: searches for other systems, connects to it, copies self to it and runs
- may disguise itself as a system process”
- implemented by Xerox Palo Alto labs in 1980’s

Worm Propagation Model



Historic Worm Attacks

- Code Red
  - July 2001 exploiting MS IIS bug
  - probes random IP address, does DDoS attack
  - consumes significant net capacity when active
- Code Red II variant includes backdoor
- SQL Slammer
  - early 2003, attacks MS SQL Server
  - compact and very rapid spread
- Mydoom
  - mass-mailing e-mail worm that appeared in 2004
  - installed remote access backdoor in infected systems

Worm Technology

- multiplatform
- multi-exploit
- ultrafast spreading
- polymorphic
- metamorphic
- transport vehicles
- zero-day exploit

Worm Countermeasures

- overlaps with anti-virus techniques
- Can be detected by A/V
- worms also cause significant net activity (detected by NIDS)

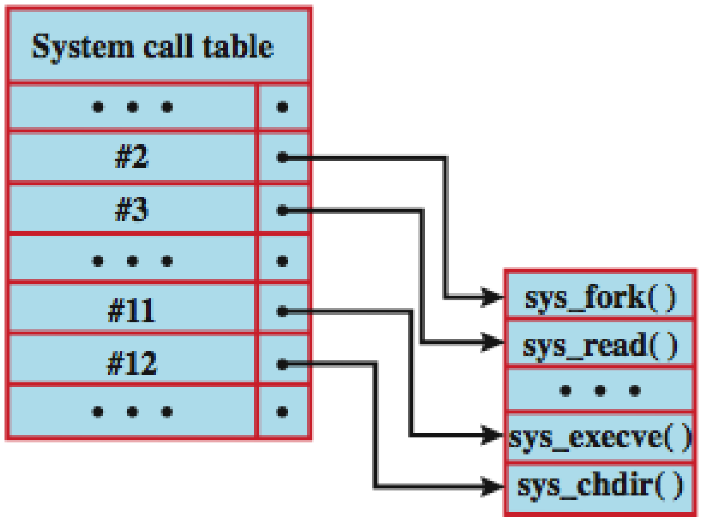
Bots

- program taking over other computers
- to launch hard to trace attacks
- if coordinated form a botnet
- characteristics:
  - remote control facility
    - via IRC/HTTP, DHT (Distributed Hash tables), etc
  - spreading mechanism
    - attack software, vulnerability, scanning strategy
- various counter-measures applicable
  - such as IDSes, honeypots and digital immune systems

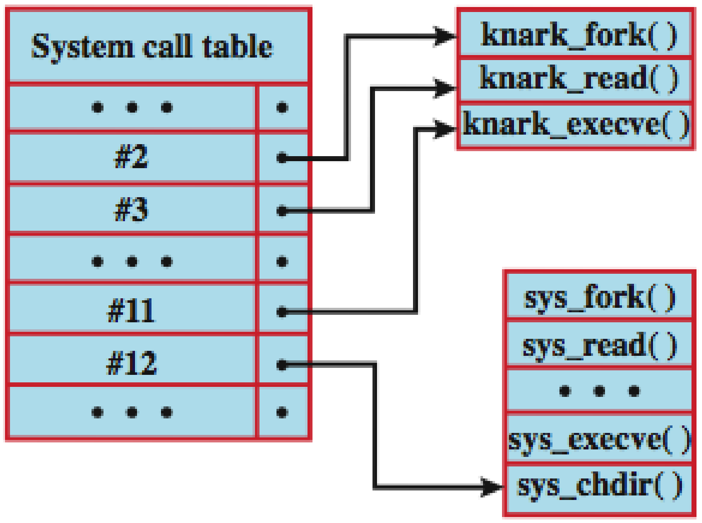
Rootkits

- set of programs installed for admin access
- malicious and stealthy changes to host O/S
- may hide its existence
  - subverting report mechanisms on processes, files, registry entries etc
- may be:
  - persisitent or memory-based
  - user or kernel mode
- installed by user via trojan or intruder on system

Rootkit System Table Mods



(a) Normal kernel memory layout



(b) After nkark install

Miscellaneous Attacks

- Numerous attacks involve software
- We'll discuss a few issues that do not fit in previous categories
  - Salami attack
  - Linearization attack
  - Time bomb
  - Can you ever trust software?

Salami Attack

- What is Salami attack?
  - Programmer "slices off" valuables
  - Slices are hard for victim to detect
- Example
  - Bank calculates interest on accounts
  - Programmer "slices off" any fraction of a cent and puts it in his own account
  - No customer notices missing partial cent
  - Bank may not notice any problem
  - Over time, programmer makes lots of money!

Salami Attack

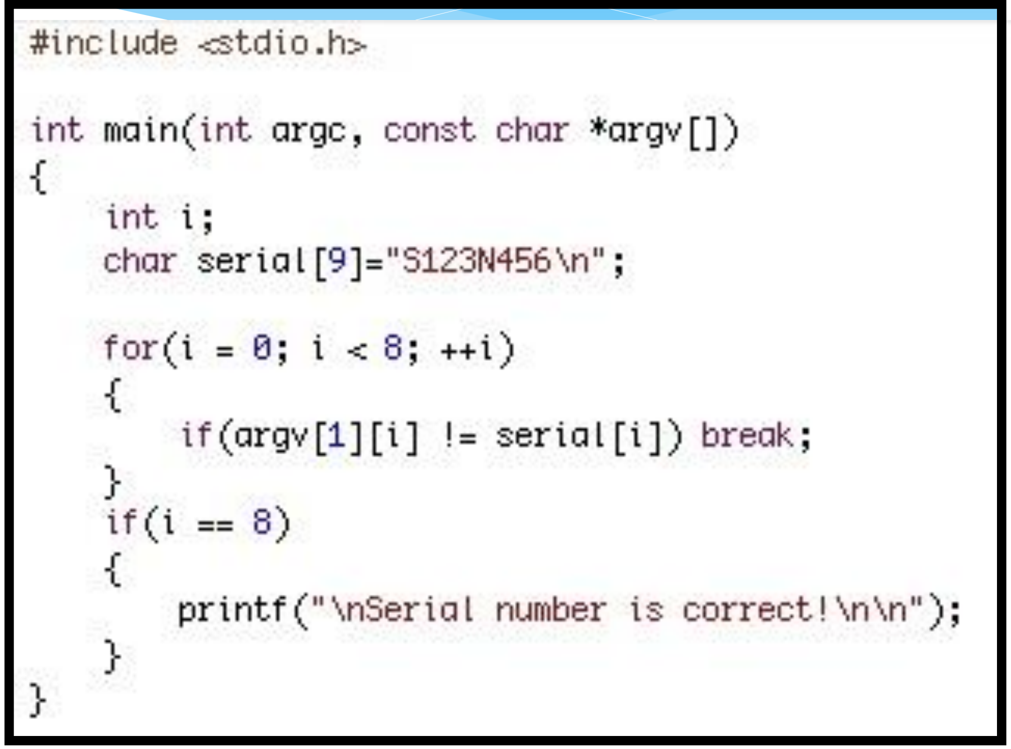
- Such attacks are possible for insiders
- Do salami attacks actually occur?
- Programmer added a few cents to every employee payroll tax withholding
  - But money credited to programmer’s tax
  - Programmer got a big tax refund!
- Rent-a-car franchise in Florida inflated gas tank capacity to overcharge customers

Salami Attacks

- Employee reprogrammed Taco Bell cash register: $2.99 item registered as $0.01
  - Employee pocketed $2.98 on each such item
  - A large “slice” of salami!
- In LA four men installed computer chip that overstated amount of gas pumped
  - Customer complained when they had to pay for more gas than tank could hold!
  - Hard to detect since chip programmed to give correct amount when 5 or 10 gallons purchased
  - Inspector usually asked for 5 or 10 gallons!

Linearization Attack

- Program checks for serial number S123N456
- For efficiency, check made one character at a time
- Can attacker take advantage of this?



```
#include <stdio.h>

int main(int argc, const char *argv[])
{
    int i;
    char serial[9]="S123N456\n";

    for(i = 0; i < 8; ++i)
    {
        if(argv[1][i] != serial[i]) break;
    }
    if(i == 8)
    {
        printf("\nSerial number is correct!\n\n");
    }
}
```

Linearization Attack

- Correct string takes longer than incorrect
- Attacker tries all 1 character strings
  - Finds S takes most time
- Attacker then tries all 2 char strings S*
  - Finds S1 takes most time
- And so on…
- Attacker is able to recover serial number one character at a time!

Linearization Attack

- What is the advantage of attacking serial number one character at a time?
- Suppose serial number is 8 characters and each has 128 possible values
  - Then $128^8 = 2^{56}$ possible serial numbers
  - Attacker would guess the serial number in about 2^{55} tries — a lot of work!
  - Using the linearization attack, the work is about $8*(128/2) = 2^9$ which is trivial!

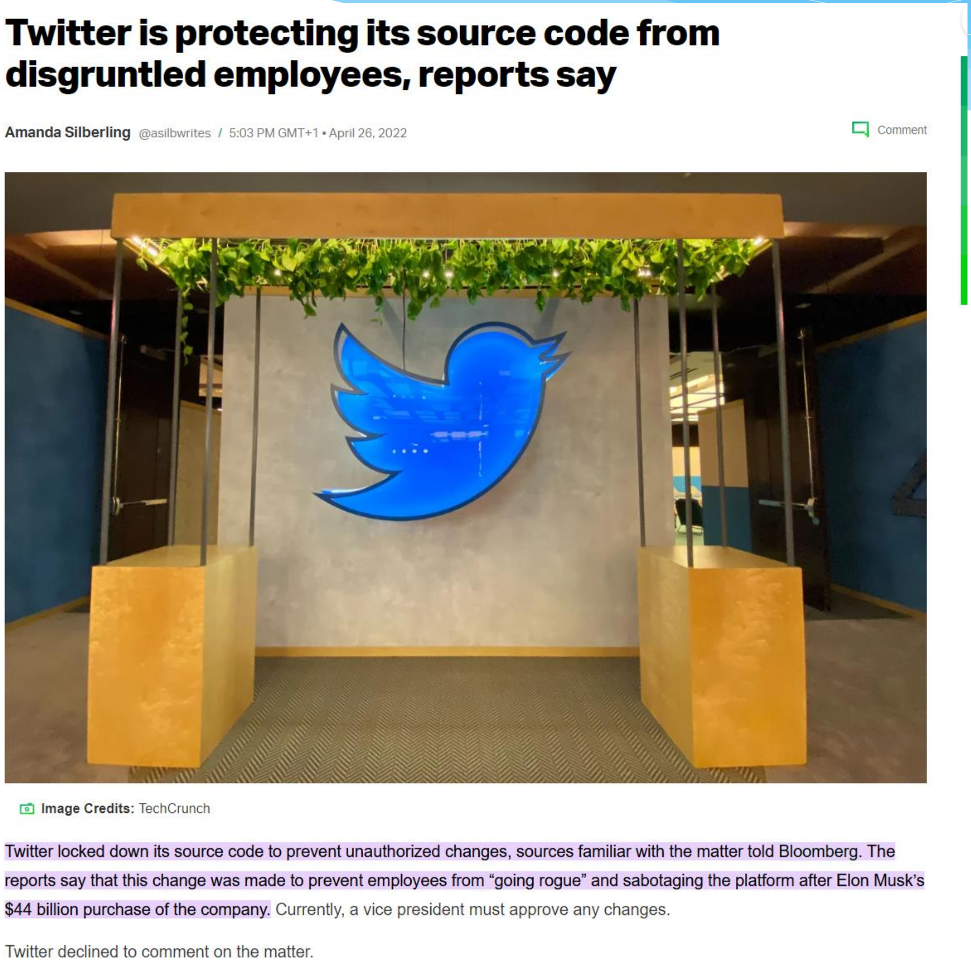
Time Bomb

- In 1986 Donald Gene Burleson told employer to stop withholding taxes from his paycheck
- His company refused
- He planned to sue his company
  - He used company computer to prepare legal docs
  - Company found out and fired him
- Burleson had been working on a malware…
- After being fired, his software “time bomb” deleted important company data

Time Bomb

- Company was reluctant to pursue the case
- So Burleson sued company for back pay!
  - Then company finally sued Burleson
- In 1988 Burleson fined $11,800
  - Took years to prosecute
  - Cost thousands of dollars to prosecute
  - Resulted in a slap on the wrist
- One of the first computer crime cases
- Many cases since follow a similar pattern
  - Companies often reluctant to prosecute

Random thoughts on what to do when people leave



Twitter is protecting its source code from disgruntled employees, reports say

Amanda Silberling @asilbwrites / 5:03 PM GMT+1 • April 26, 2022

Comment

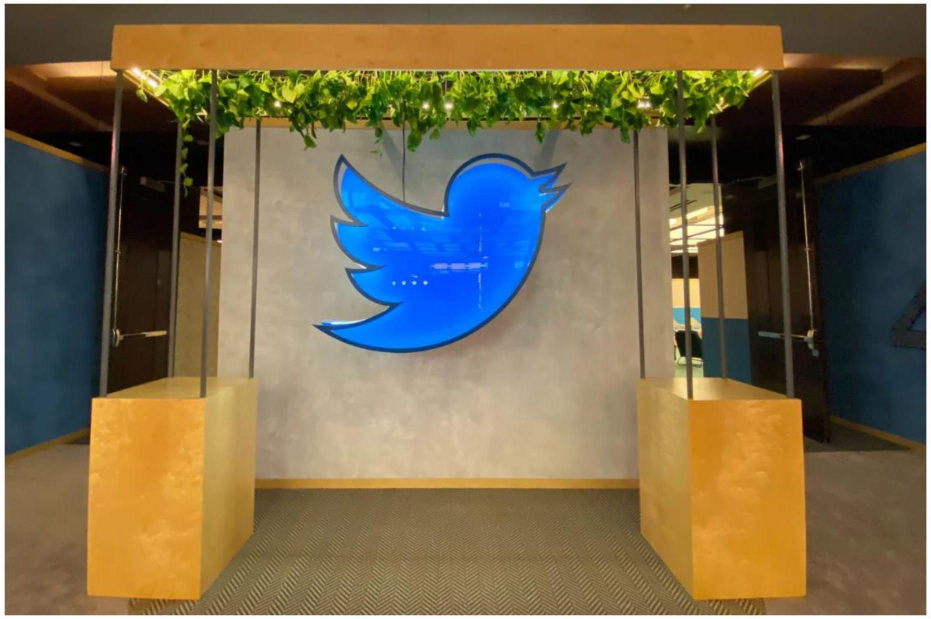


Image Credits: TechCrunch

Twitter locked down its source code to prevent unauthorized changes, sources familiar with the matter told Bloomberg. The reports say that this change was made to prevent employees from "going rogue" and sabotaging the platform after Elon Musk's $44 billion purchase of the company. Currently, a vice president must approve any changes.

Twitter declined to comment on the matter.