Intrusioni e Virus

Tecniche di Intrusione

- Obiettivo: Accedere al Sistema, spesso ottenendo la passwd dell'utente
- Nella maggior parte dei sistemi esiste un file che associa ogni utente alla rispettiva passwd
- Password file protection:
  - one-way encryption
  - access control

Three Classes of Intruders

- Masquerader – utente non autorizzato che accede al Sistema usando l'account *di un utente legittimo* (*outside*)
- Misfeasor – utente legittimo che accede a servizi per cui non è aitorizzato, oppure che fa cattivo uso dei suoi privilegi (*inside*)
- Clandestine user – elude il controllo per evadere le analisi sulle sue attività *(inside|outside)*

Intruders

Intruder attacks range from benign to serious:

- Benign intruders tolerable but consume resources
- Difficult to know in advance the type of intruder
- Really growing problem
  - globalization
  - the move to Client/Server architectures
  - hacker's steep learning curve

Types Of Hackers

- Old School – Capt Crunch – no malicious intent – believe in open system
- Script Kiddies – 12-30 yrs old, mostly males – limited knowledge – too much time on their hands – also called Cyber Punks – brag and get caught

Types Of Hackers

- Professional Criminals – Crackers – careers built on criminal hacking – break into secure areas and sell information – often involved in espionage and organized crime

Cyber Punk

Kevin Poulsen
1990

Notare la foto: anche se “cattivo” è un “bel ragazzo”. E’ Americano!

- Took over all the telephone lines of Los Angeles KISS-FM radio station - he then made himself the 102nd caller and won a $50,000 944 S2 Porche
- Indicted for 19 counts of conspiracy, fraud, wiretapping and money laundering - spent 3 years in prison

Crackers

Vladimir Levin
1994

Notare la foto: “cattivo” anche se matematico, comunque brutto. E’ Russo! Chiedetevi ora perché Trump vince

- Russian mathematician – led group that hacked into Citibank computers and extorted 10 million dollars.
- Caught in 1995 by Interpol - sentenced to three years in prison and forced to give up his share of the money.

Types Of Hackers

- Coders – Virus Writers - see themselves as an elite group - they have a lot of programming background and write code, but won't use it themselves
- They have their own networks to experiment with, which they call **Zoos**
- They leave it to others to introduce their codes into **The Wild**, or the Internet.

Tecniche per Acquisire le Passwd

guess attack

1. Try default passwords used with standard accounts shipped with the system
2. Exhaustive try of all short passwords
3. Try words in system's dictionary or list of likely passwords (hacker bulletin boards)
4. Collect information about users (full names, names of spouses and children, pictures and books in their office, related hobbies)
5. Try users' phone numbers, social security numbers, room numbers
6. Try all legitimate license plate numbers
7. Use a trojan horse
8. Tap the line between a remote user and the system

Intrusion Detection

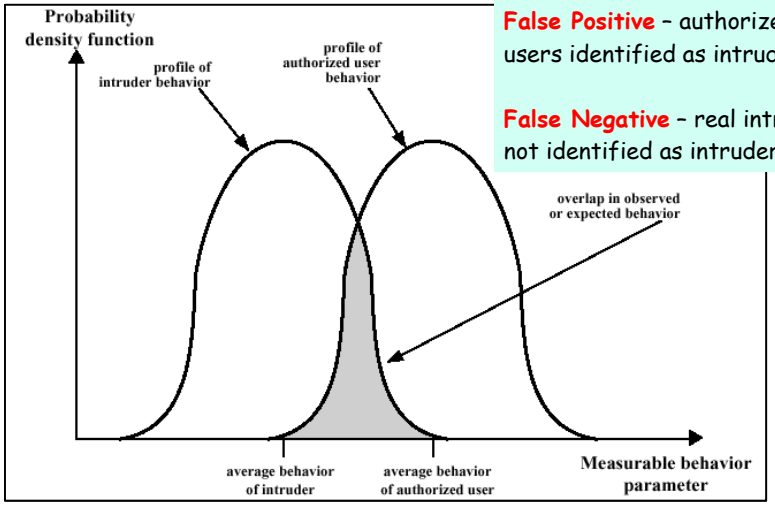
Second line of defense (firewall is 1st)

- Quick detection – minimizza i danni e permette un più veloce ripristino
- Deterrent - un Sistema di intrusion detection efficace per prevenire le intrusioni
- Collection of techniques - information about intrusion techniques leads to stronger prevention facility

Intrusion Detection

- Basic Assumption:
  Behavior of the intruder *differs* from legitimate user in quantifiable ways
- There is an element of compromise and art in the practice of intrusion detection

Intruder & Authorized User Behavior



False Positive – authorized users identified as intruders

False Negative – real intruders not identified as intruders

Finding The Bad Guy

- Necessità di distinguere un masquerader da un legitimate user
- Osservare la storia passata (Bayes Theorem)
- Stabilire un pattern of behavior
- Osservare deviazioni significative

Two Approaches:

Statistical Anomaly Detection

- Collection of data over a period of time about legitimate user behavior
- Statistical tests to observe behavior and confidently determine non-legitimate use
  - Threshold detection: for frequency of occurrence of certain events
  - Profile-based: profile of user activity and change detection
- Successful against masqueraders but not against misfeasors

Two Approaches:

Rule-based Detection

- Attempt to define set of rules that determine intruder's behavior
  - Anomaly detection: detect deviation from previous usage patterns
  - Penetration identification: expert system that searches for suspicious behavior
- Better approach for detecting penetration

Audit Record

Basic Tool of Intrusion Detection

- Native audit records
  - Information collected for accounting
  - No extra cost but not necessary or conveniently formed information
- Detection-specific audit records
  - Only info required by IDS
  - Extra overhead
  - Vendor independent
  - Subject, action, object, exception condition, resource usage, timestamp (Denning)

Statistical Anomaly Categories

- Threshold detection
  - Counting the *number of occurrences* of a specific event type over an *interval of time*
  - Generate either a lot of false positives or a lot of false negatives
- Profile-based systems
  - Characterizing the *past behavior* of individual users or related groups of users and then *detecting significant deviations*
  - A profile is a *set of parameters*
  - *Foundation* of this approach is an analysis of *audit records*
  - *Records over time* define typical behavior. *Current audit records* are used to detect intrusion

Statistical Anomaly Detection

- Various tests determine whether current activity fits within acceptable limits
  - Mean & standard deviation – crude for intrusion detection
  - Multivariate – correlation determines intruder behavior
  - Markov process – establish transition probabilities among various states
  - Time series – focus on time intervals
  - Operational model – exceeding fixed limits
- Prior knowledge of security flaws is not required

Measures Used For Intrusion Detection

Measure	Model	Type of Intrusion Detected
Login and Session Activity		
Login frequency by day and time	Mean and standard deviation	Intruders may be likely to log in during off-hours.
Frequency of login at different locations	Mean and standard deviation	Intruders may log in from a location that a particular user rarely or never uses.
Time since last login	Operational	Break-in on a "dead" account.
Elapsed time per session	Mean and standard deviation	Significant deviations might indicate masquerader.
Quantity of output to location	Mean and standard deviation	Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data.
Session resource utilization	Mean and standard deviation	Unusual processor or I/O levels could signal an intruder.
Password failures at login	Operational	Attempted break-in by password guessing.
Failures to login from specified	Operational	Attempted break-in.
Command or Program Execution Activity		
Execution frequency	Mean and standard deviation	May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.
Program resource utilization	Mean and standard deviation	An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization.
Execution denials	Operational model	May detect penetration attempt by individual user who seeks higher privileges.
File access activity		
Read, write, create, delete frequency	Mean and standard deviation	Abnormalities for read and write access for individual users may signify masquerading or browsing.
Records read, written	Mean and standard deviation	Abnormality could signify an attempt to obtain sensitive data by inference and aggregation.
Failure count for read, write, create, delete	Operational	May detect users who persistently attempt to access unauthorized files.
File resource exhaustion counter	Operational	

Rule-Based Detection

- Observe events in the system and apply a set of rules that decide if activity is suspicious or not
- Approaches focus on either:
  - Anomaly detection
  - Penetration identification

Rule-Based Anomaly Detection

- Similar in terms of approach and strengths to statistical anomaly detection
- Automatically generate rules by analyzing historical audit records to identify usage patterns
- Assume the future will look like the past and apply rules to current behavior
- Does not require a knowledge of security vulnerabilities
- Requires a rather large database of rules (10^4 to 10^6)

Rule-Based Penetration Identification

- Based on expert system technology
- Uses rules for identifying known penetrations or ones that exploit known weaknesses – suspicion rating
- Rules generated by experts and system specific
- Strength is a function of the skills of the rule makers – hire a hacker
- Early systems: NIDX, IDES, Haystack – late 80's
- Best approach is a high level model that is independent of specific audit records
- USTAT, a state transition model, deals with general actions and reduces the number of rules

Base-Rate Fallacy

- IDS system must meet the standard of high rate of detections with a low rate of false alarms
- False alarm rate is the limiting factor for the performance of an IDS
- This is due to the Base-Rate Fallacy - the belief that probability rates are false – i.e., failure to take base rates into account when judging probability

Base-Rate Fallacy

A cab was involved in a hit-and-run accident at night. Two cab companies, the Green and the Blue, operate in the city.

You are given the following data:

- 85% of the cabs in the city are Green and 15% are Blue.
- A witness identified the cab as a Blue cab.

The court tested his ability to identify cabs under the appropriate visibility conditions. When presented with a sample of cabs (half of which were Blue and half of which were Green) the witness made correct identifications in 80% of the cases and erred in 20% of the cases.

Question: What is the probability that the cab involved in the accident was Blue rather than Green?"

Base-Rate Fallacy

When people answer this, they tend to say that the probability it was Blue (the rare case) is about 80%, but the real probability is 41%, because this takes into account the fact that there are may more green cabs than blue ones.

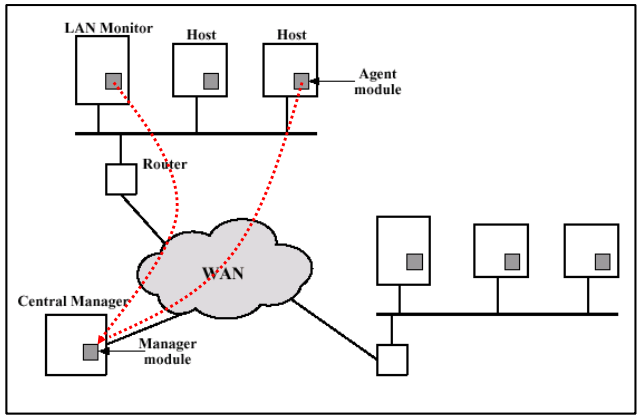
The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection - Stefan Axelsson

Bottom Line: IDS systems have a long way to go!

Distributed Intrusion Detection Scalability Issues

- Too much overhead for standalone IDS on each host
- Heterogeneous environment – different audit records
- Need IDS across the network
- Centralized vs decentralized issues

Distributed Intrusion Detection

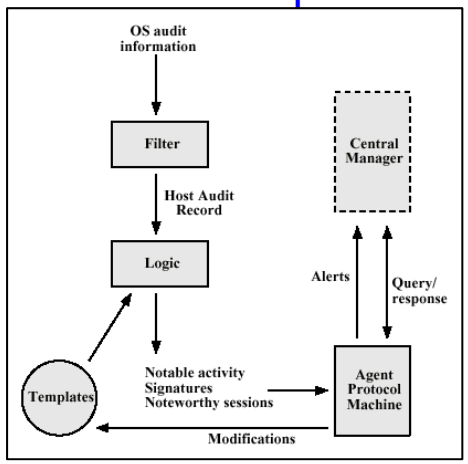


Distributed Intrusion Detection

- Host agent module – background process collects data and sends results to the central manager
- LAN monitor agent module – analyzes LAN traffic and sends results to the central manager
- Central manager module – processes and correlates received reports to detect intrusion

Agent Architecture

Machine Independent



Honeypots

- Decoy systems
- Lure attacker from critical systems
- Collect information about the attacker
- Keep attacker around long enough to respond
- Jury is still out on this!

Password Management

Password Protection

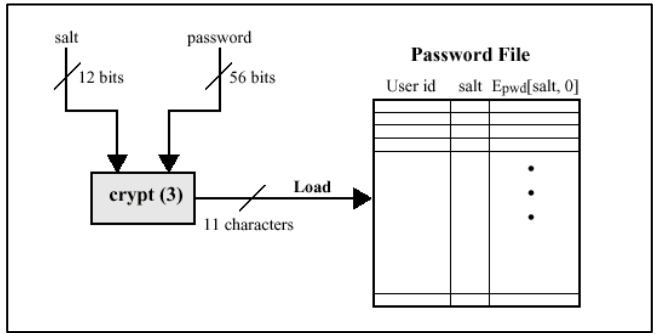
User ID and password:

- User authorized to gain access to the system
- Privileges accorded to the user
- Discretionary access control

Password Protection

- Unix system (user ID, cipher text password, plain text salt)
  - password 8 printable characters - 56-bit value (7-bit ASCII)
  - encryption routine (crypt(3)) based on DES
  - modified DES algorithm with 12-bit salt value (related to time of password assignment)
  - 25 encryptions with 64-bit block of zeros input
  - 64-bit - 11 character sequence

Loading A New Password

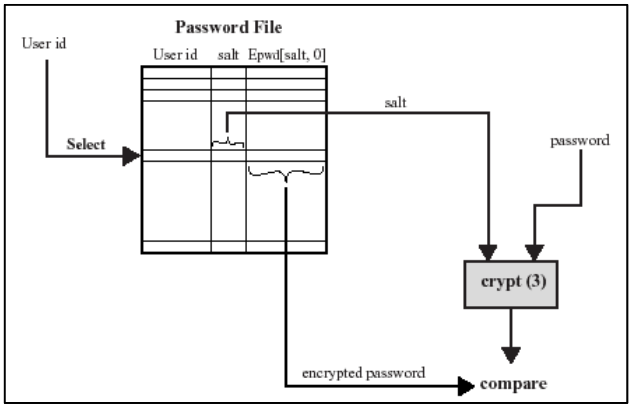


Password Protection

Purposes of salt:

- Prevents duplicate passwords from being visible
- Effectively increases password length without the user needing to remember additional 2 characters (possible passwords increased by 4096)
- Prevent use of hardware DES implementation for a brute-force guessing attack

Verifying A Password



Password Protection

Unix password scheme threats:

- Gain access through a guest account and run a password cracker
- Obtain a copy of the password file and run a password cracker

Goal: Run a password cracker

- Rely on people choosing easily guessable passwords!

Observed Password Lengths In a Purdue Study

Length	Number	Fraction of Total
1	55	.004
2	87	.006
3	212	.02
4	449	.03
5	1260	.09
6	3035	.22
7	2917	.21
8	5772	.42
Total	13787	1.0

Passwords Cracked From A Sample Set

easy pickin's →

Type of Password	Search Size	Number of Matches	Percentage of Passwords Matched	Cost/Benefit Ratio[a]
User/account name	130	368	2.7%	2.830
Character sequences	866	22	0.2%	0.025
Numbers	427	9	0.1%	0.021
Chinese	392	56	0.4%	0.143
Place names	628	82	0.6%	0.131
Common names	2239	548	4.0%	0.245
Female names	4280	161	1.2%	0.038
Male names	2866	140	1.0%	0.049
Uncommon names	4955	130	0.9%	0.026
Myths & legends	1246	66	0.5%	0.053
Shakespearean	473	11	0.1%	0.023
Sports terms	238	32	0.2%	0.134
Science fiction	691	59	0.4%	0.085
Movies and actors	99	12	0.1%	0.121
Cartoons	92	9	0.1%	0.098
Famous people	290	55	0.4%	0.190
Phrases and patterns	933	253	1.8%	0.271
Surnames	33	9	0.1%	0.273
Biology	58	1	0.0%	0.017
System dictionary	19683	1027	7.4%	0.052
Machine names	9018	132	1.0%	0.015
Mnemonics	14	2	0.0%	0.143
King James bible	7525	83	0.6%	0.011
Miscellaneous words	3212	54	0.4%	0.017
Yiddish words	56	0	0.0%	0.000
Asteroids	2407	19	0.1%	0.007
TOTAL	62727	3340	24.2%	0.053

Access Control

One Method: ***Deny access to password file***

- Systems susceptible to unanticipated break-ins
- An accident in protection may render the password file readable compromising all accounts
- Users have accounts in other protection domains using the same passwords

Access Control

- *Answer:*

  Force users to select passwords that are difficult to guess
- *Goal:*

  Eliminate guessable passwords while allowing the user to select a password that is memorable

Password Selection Strategies

(Basic Techniques)

- User education
  - Users may ignore the guidelines
- Computer-generated passwords
  - Poor acceptance by users
  - Difficult to remember passwords

Password Selection Strategies

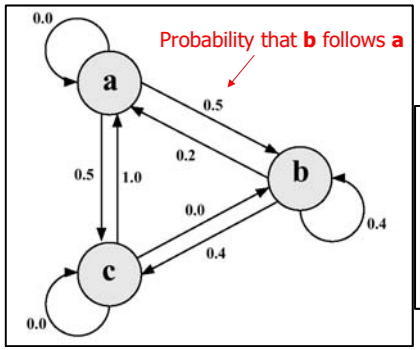
- **Reactive password checking**
  - System runs its own password cracker
  - Resource intensive
  - Existing passwords remain vulnerable until reactive checker finds them
- **Proactive password checking**
  - Password selection is guided by the system
  - Strike a balance between user accessibility and strength
  - May provide guidance to password crackers (what not to try)
  - Dictionary of bad passwords (space and time problem)

Proactive Password Checker

There are two techniques currently in use:

- Markov Model – search for guessable password
- Bloom Filter – search in password dictionary

Markov Model



M = {states, alphabet, prob, order}

M = {3, {a, b, c} , T, 1 } where

$$T = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.2 & 0.4 & 0.4 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$$

e.g., string probably from this language: abbcacaba

e.g., string probably not from this language: aaccbaaa

Markov Model

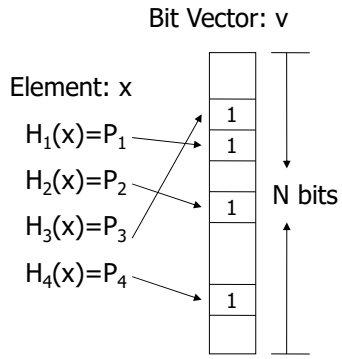
- "Is this a bad password?"…same as…
- "Was this password generated by this Markov model?"
- Passwords that are likely to be generated by the model are rejected
- Good results for a second-order model

Bloom Filter

- A probabilistic algorithm to quickly test membership in a large set using multiple hash functions into a single array of bits
- Developed in 1970 but not used for about 25 years
- Used to find words in a dictionary also used for web caching
- Small probability of false positives which can be reduced for different values of *k*, # hash funcs
- www.cs.wisc.edu/~cao/papers/summary-cache/node8.html – a good tutorial

Bloom Filter

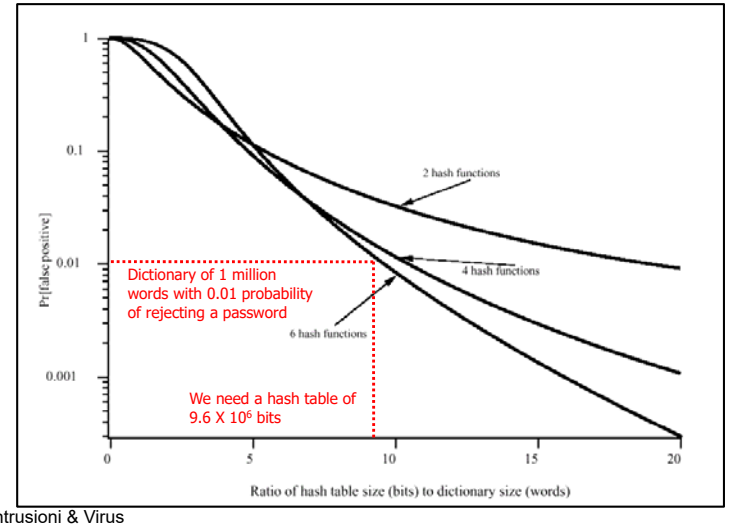
- A vector v of N bits
- k independent hash functions. Range 0 to N-1
- For each element x, compute hash functions $H_1(x)$, $H_2(x)$…$H_k(x)$
- Set corresponding bits to 1
- *Note:* A bit in the resulting vector may be set to 1 multiple times



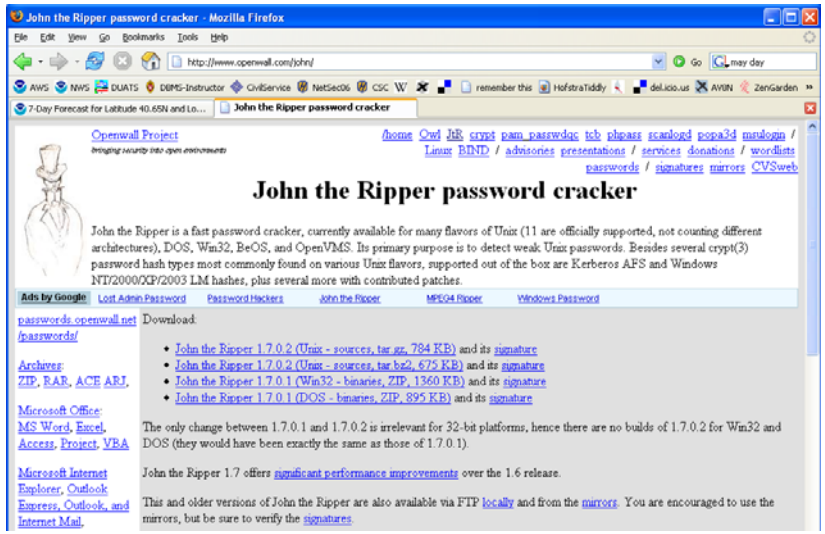
Bloom Filter

- To query for existence of an entry x, compute $H_1(x)$, $H_2(x)$…$H_k(x)$ and check if the bits at the corresponding locations are 1
- If not, x is definitely not a member
- Otherwise there may be a false positive (passwords not in the dictionary but that produce a match in the hash table). The probability of a false positive can be reduced by choosing k and N

Performance of Bloom Filter



Password Cracking



Password Cracking

Unix Password File (/etc/passwd):

```
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
nobody:x:60001:60001:Nobody:/:
eric:GmTFg0AavFA0U:1001:10:Eric Schwartz:/export/home/eric:/bin/ksh
temp:kRWegG5iTZP5o:1002:10:IP Administration:/export/home/ipadmin:/bin/ksh
jfr:kyzKROryhFDE2:506:506::/home/jfr:/bin/csh
```

Results of the password cracker:

```
$ john  passwd
Loaded 3 passwords with 3 different salts (Standard DES [24/32 4K])
temp             (temp)
jenny            (eric)
solaris1         (jfr)
```


Password Crackers

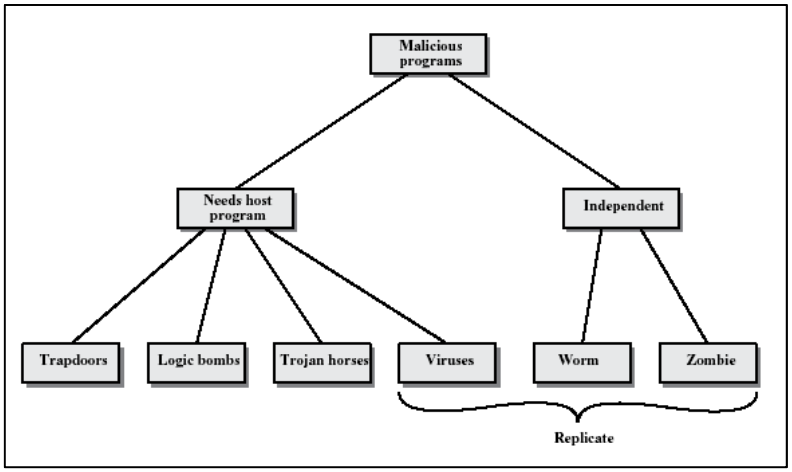
Tool	Capabilities	Website	Linux/ Unix	Win32	Cost
Crack 5	Unix password cracker	http://www.crypticide.org/users/alecm/	✓		Free
Description	*Crack is a password guessing program that is designed to quickly locate insecurities in Unix (or other) password files by scanning the contents of a password file, looking for users who have misguidedly chosen a weak login password.*				
IMP 2.0	Novell Netware password cracker	http://www.wastelands.gen.nz		✓	Free
Description	*Imp is a NetWare password cracking utility with a GUI (Win95/NT). It loads account information directly from NDS or Bindery files and allows the user to attempt to compromise the account passwords with various attack methods.*				
John the Ripper	Windows and Unix password cracker	http://www.openwall.com/john/	✓	✓	Free
Description	*John the Ripper is a fast password cracker, currently available for many flavors of Unix, DOS, Win32, and BeOS. Its primary purpose is to detect weak Unix passwords, but a number of other hash types are supported as well.*				
L0pht Crack	Windows password cracker	http://www.securityfocus.com/tools/1005		✓	$
Description	*A password cracking utility for Windows NT, 2000 and XP.*				
Nwpcrack	Novell Netware password cracker	http://ftp.cerias.purdue.edu/pub/tools/novell/		✓	Free
Description	*A password cracking utility for Novell Netware.*				

Virus and Related Threats

Malicious Programs

- Due categorie:
  - Quelli che necessitano di un host program – o frammenti di programmi - parasitic
  - Quelli indipendenti – self contained
- Some replicate – used as a differentiator

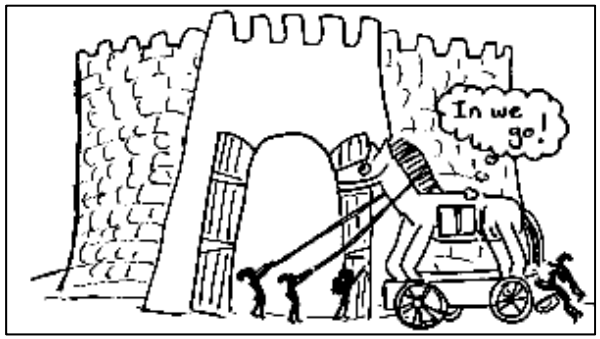
Taxonomy of Malicious Programs



Malicious Programs

- Logic Bombs: logic embedded in a program that checks for a set of conditions to arise and executes some function resulting in unauthorized actions
- Trapdoors: secret undocumented entry point into a program, used to grant access without normal methods of access authentication (*e.g.,War Games*)

Trojan Horse



Cavallo di Troia
Mimmo Paladino

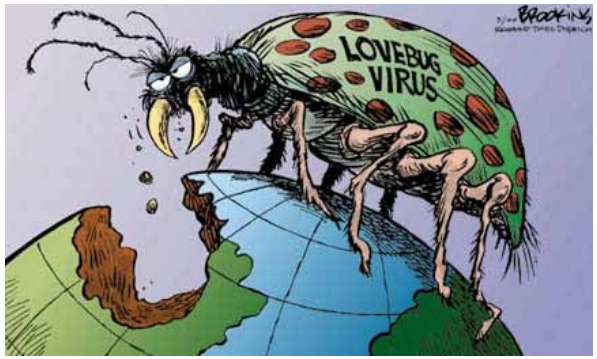
Malicious Programs

- Trojan Horse: secret undocumented routine embedded within a useful program, execution of the program results in execution of the routine
- Common motivation is data destruction

Malicious Programs

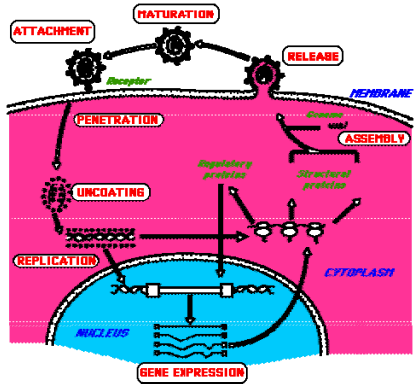
- Zombie: a program that secretly takes over an Internet attached computer and then uses it to launch an untraceable attack
- Very common in Distributed Denial-Of-Service attacks

Viruses



Viruses

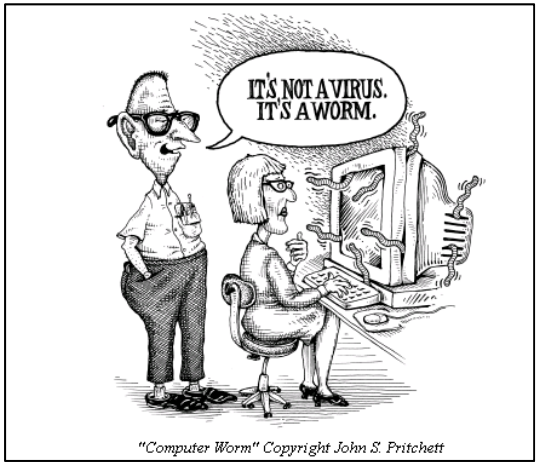
- A virus is a submicroscopic parasitic particle that infects cells in biological organisms.
- Viruses are non-living particles that can only replicate when an organism reproduces the viral RNA or DNA.
- Viruses are considered non-living by the majority of virologists



Viruses

- Viruses: code embedded within a program that causes a copy of itself to be inserted in other programs and performs some unwanted function
- *Infects* other programs
- *Code* is the *DNA* of the virus

Worms



"Computer Worm" Copyright John S. Pritchett

Worms

- Worms: program that can replicate itself and send copies to computers across the network and performs some unwanted function
- Uses *network connections* to spread from system to system

Bacteria

- Bacteria: *consume resources* by replicating themselves
- Do not explicitly damage any files
- *Sole purpose* is to *replicate* themselves
- Reproduce exponentially
- Eventually taking up all processors, memory or disk space

Nature of Viruses

Four stages of virus lifetime

- Dormant phase: virus idle
- Propagation phase: cloning of virus
- Triggering phase: virus activation
- Execution phase: unwanted function performed

Virus Structure

```
program V:=

{goto main:
1234567;          <-- special marker determines if infected

    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:   <-- transfer control to the original program

}
```


Avoiding Detection

- Infected version of program is longer than the corresponding uninfected one
- *Solution:* compress the executable file so infected and uninfected versions are identical in length

Avoiding Detection

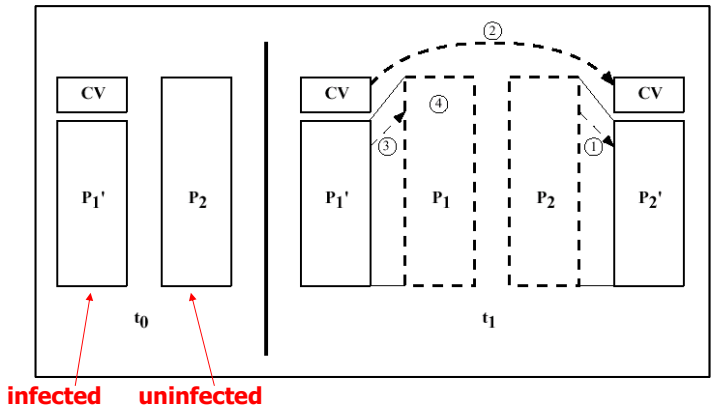
```
        program CV :=

{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;}
                }
```

Compression Program



Types of Viruses

- Parasitic Virus: attached to executables, replicates when program is executed
- Memory-resident virus: part of a resident system program, affects every program executed
- Boot sector virus: infects a master boot record and spreads when system is booted from infected disk

Types of Viruses

- Stealth virus: virus designed to hide itself from detection by antivirus software (compression, interception of I/O logic)
- Polymorphic virus: mutates with every infection making detection by "signature" impossible (mutation engine)
- Macro virus: infects Microsoft Word docs; 2/3's of all viruses

Macro Viruses

- 2/3s of all viruses
- Mainly Microsoft products – platform independent
- Affect documents not executables
- Easily spread by e-mail
- Autoexecuting macro is the culprit

Worms

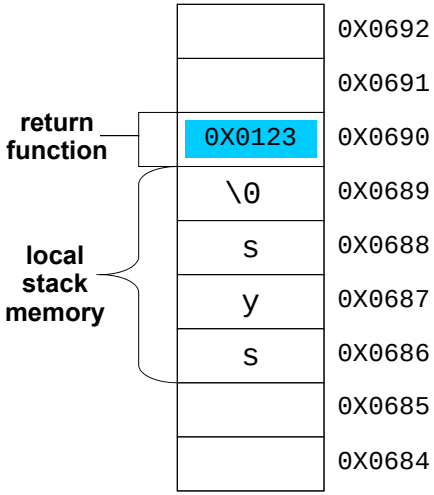
- Uses network connections to spread from system to system
- Similar to a virus – has same phases: dormant, propagation, trigger and execution
- Morris Worm – most famous
- Recent: OSX.Leap.A, Kama Sutra,Code Red

Buffer Overflow

- Program attempts to write more data into buffer than that buffer can hold…
- …Starts overwriting area of stack memory
- Can be used maliciously to cause a program to execute code of attackers choose
- Overwrites stack point

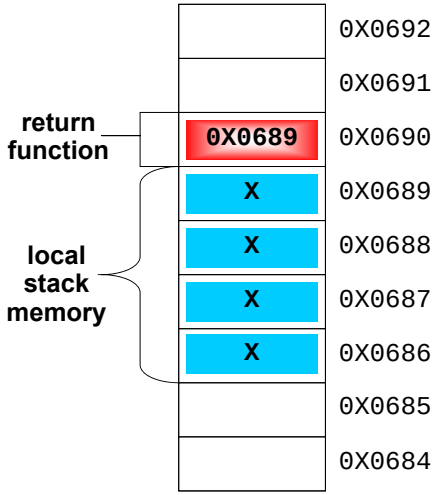
Mechanics of stack-based buffer overflow

- Stack is like a pile of plates
- When a function is called, the return address is pushed on the stack
- In a function, local variables are written on the stack
- Memory is written on stack
  - char username[4] reserved 4 bytes of space on stack



Mechanics of stack-based buffer overflow

- When function copies too much on the stack...
- ...the return pointer is overwritten
- Execution path of function changed when function ends
- Local stack memory has malicious code



Antivirus Approaches

- Detection – determine that it has occurred and locate the virus
- Identification – identify the specific virus
- Removal – remove all traces and restore the program to its original state

Generations of Antivirus Software

- First: simple scanners (record of program lengths)
- Second: heuristic scanners (integrity checking with checksums)
- Third: activity traps (memory resident, detect infected actions)
- Fourth: full-featured protection (suite of antivirus techniques, access control capability)

Advanced Techniques

- Generic Decryption
- Digital Immune System
- Behavior-Blocking Software

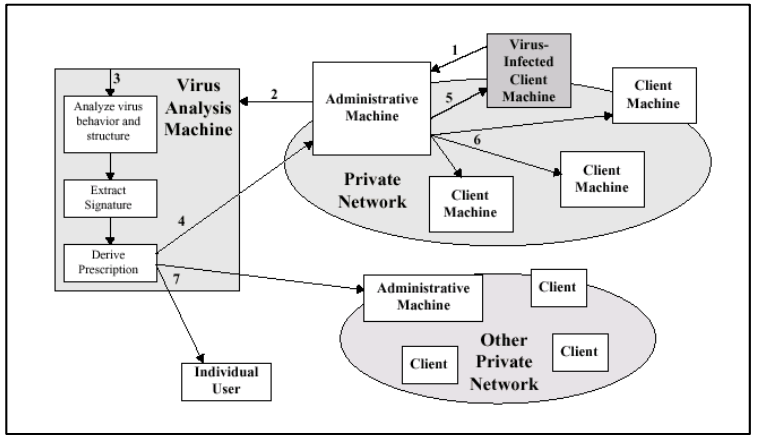
Generic Decryption

- Easily detects even most complex polymorphic virus
- No damage to the personal computer
- Contains following elements:
  - CPU emulator – software based virtual computer
  - Virus signature scanner – scans target code for known signatures
  - Emulation control module – control execution of target code

Digital Immune System

- Pioneered by IBM
- Response to rate of virus propagation
  - Integrated mail systems - Outlook
  - Mobile program systems – ActiveX, Java
- Expands the use of program emulation
- Depends on a central virus analysis machines

Digital Immune System



Behavior-Blocking Software

- Monitors program behavior in real-time for malicious actions – part of OS
- Look for well defined requests to the OS: modifications to files, disk formats, mods to scripts or macros, changes in config settings, open network connections, etc.
- IPS – Intrusion Prevention Systems

Malicious Code Protection
Types of Products

- Scanners - identify known malicious code - search for *signature strings*
- Integrity Checkers – determine if code has been altered or changed – *checksum* based
- Vulnerability Monitors - prevent modification or access to particularly sensitive parts of the system – user defined
- Behavior Blockers - list of rules that a legitimate program must follow – *sandbox* concept

Important URLs

- http://www.cert.org/
  Originally DARPA's computer emergency response team. An essential security site
- http://www.research.ibm.com/antivirus/
  IBM's site on virus information. Very good papers – a little outdated
- http://www.afsa.org/fsj/sept00/Denning.cfmHacktivism: An Emerging Threat to Diplomacy, another Denning term along with Information Warfare
- http://csrc.nist.gov/virus/Computer Security Resources Center – Virus information and *alerts*

Important URLs

- http://www.ciac.org/ciac/
  Computer Incident Advisory Capability -another bookmark-able site to visit regularly
- http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf
  Guideline on Network Security Testing – covers password cracking
- http://www.openwall.com/john/
  Very good password cracker, "John the Ripper"
- http://csrc.nist.gov/publications/nistpubs/800-36/NIST-SP800-36.pdf
  Guide to Selecting Information Security Products
- http://www.xensource.com/
  Xen Source - Hottest Area In Virtualization