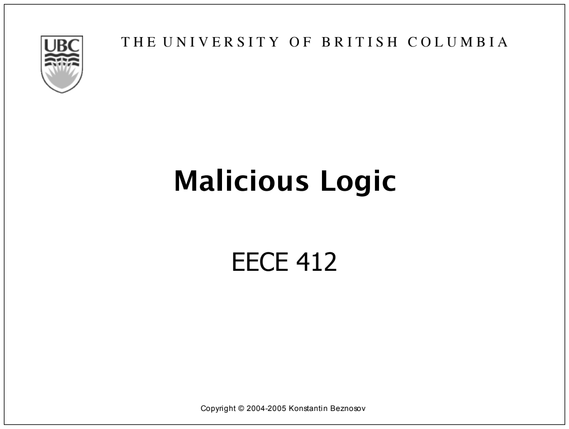
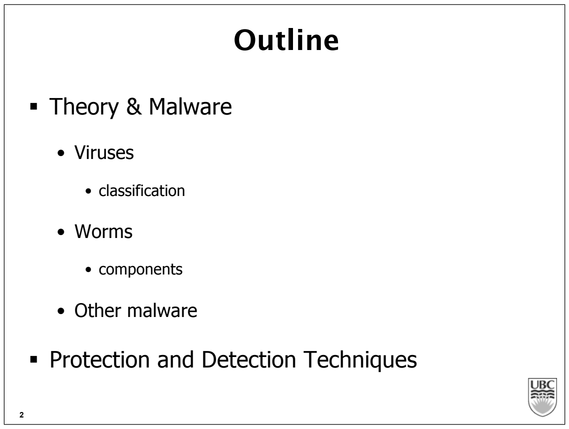


THE UNIVERSITY OF BRITISH COLUMBIA

Malicious Logic

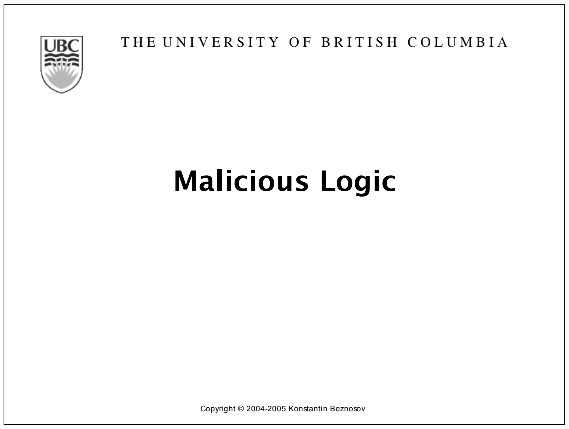
EECE 412



Outline

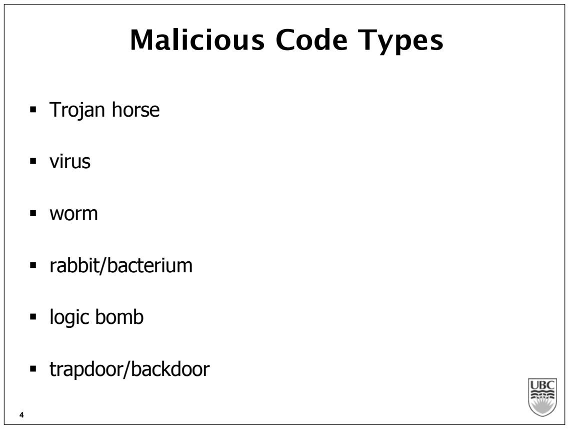
- Theory & Malware
  - Viruses
    - classification
  - Worms
    - components
  - Other malware
- Protection and Detection Techniques

2



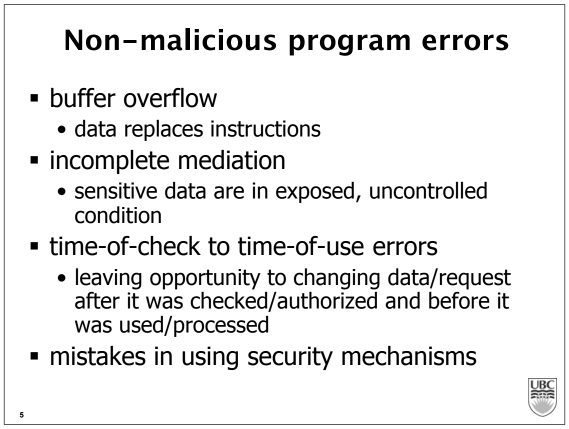
THE UNIVERSITY OF BRITISH COLUMBIA

Malicious Logic



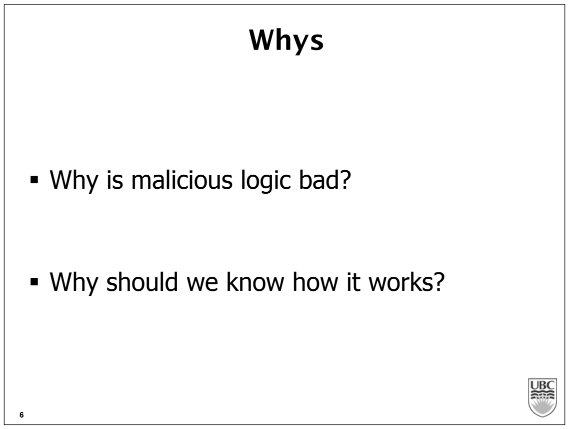
Malicious Code Types

- Trojan horse
- virus
- worm
- rabbit/bacterium
- logic bomb
- trapdoor/backdoor



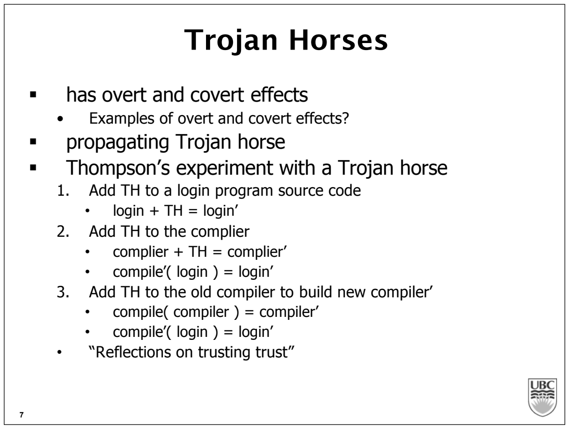
Non-malicious program errors

- buffer overflow
  - data replaces instructions
- incomplete mediation
  - sensitive data are in exposed, uncontrolled condition
- time-of-check to time-of-use errors
  - leaving opportunity to changing data/request after it was checked/authorized and before it was used/processed
- mistakes in using security mechanisms



Whys

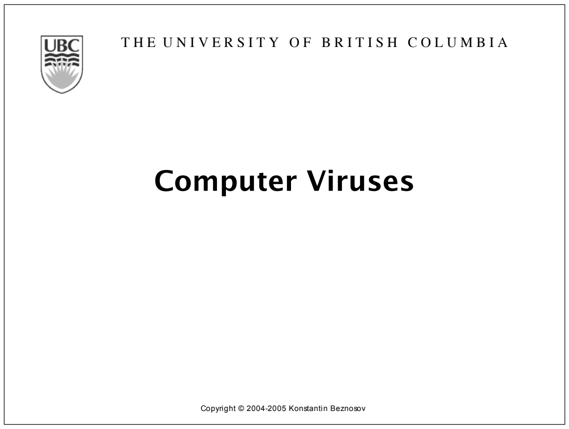
- Why is malicious logic bad?
- Why should we know how it works?



Trojan Horses

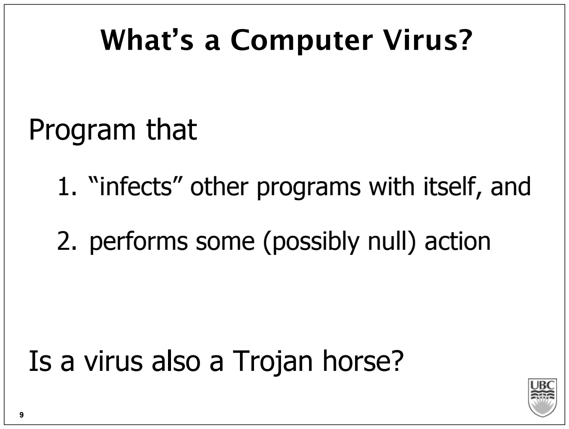
- has overt and covert effects
  - Examples of overt and covert effects?
- propagating Trojan horse
- Thompson’s experiment with a Trojan horse
  1. Add TH to a login program source code
     - login + TH = login’
  2. Add TH to the complier
     - complier + TH = complier’
     - compile’(login) = login’
  3. Add TH to the old compiler to build new compiler’
     - compile(compiler) = compiler’
     - compile’(login) = login’
  - “Reflections on trusting trust”

7



THE UNIVERSITY OF BRITISH COLUMBIA

Computer Viruses



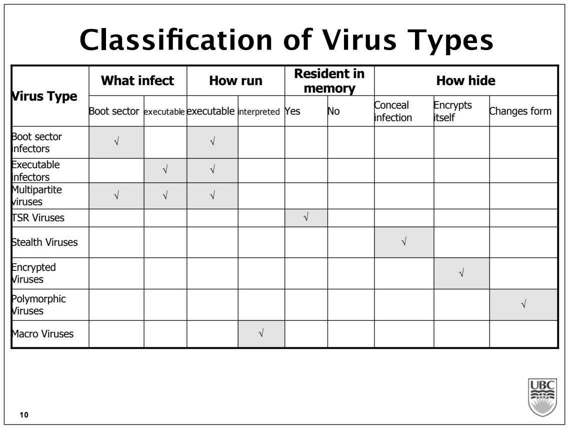
What’s a Computer Virus?

Program that

1. “infects” other programs with itself, and
2. performs some (possibly null) action

Is a virus also a Trojan horse?

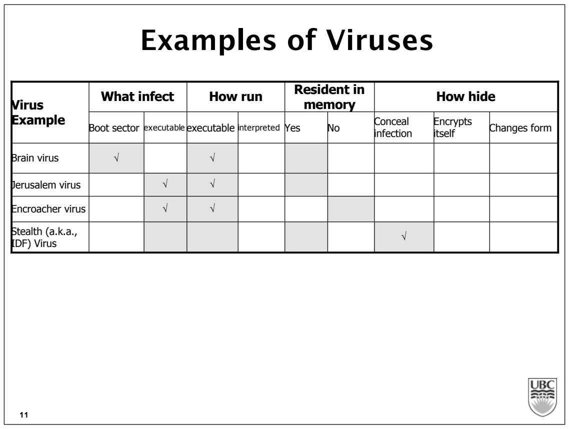
9



Classification of Virus Types

Virus Type	What infect		How run		Resident in memory		How hide		
	Boot sector	executable	executable	interpreted	Yes	No	Conceal infection	Encrypts itself	Changes form
Boot sector infectors	√		√						
Executable infectors		√	√						
Multipartite viruses	√	√	√						
TSR Viruses					√				
Stealth Viruses							√		
Encrypted Viruses								√	
Polymorphic Viruses									√
Macro Viruses				√					

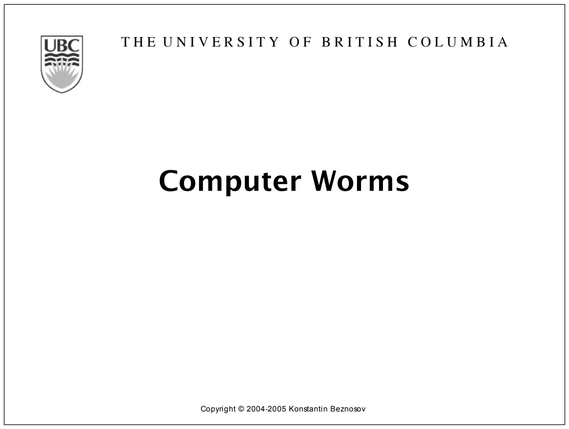
10



Examples of Viruses

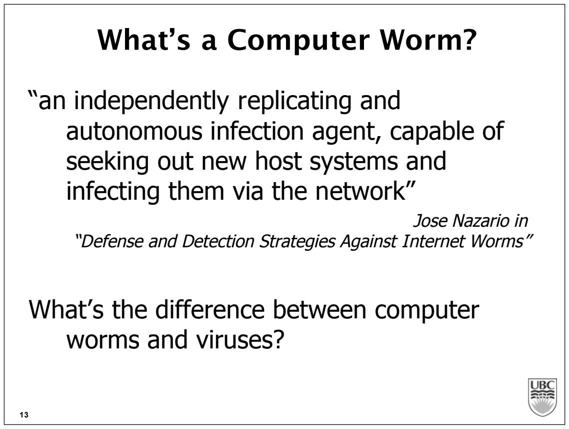
Virus Example	What infect		How run		Resident in memory		How hide		
	Boot sector	executable	executable	interpreted	Yes	No	Conceal infection	Encrypts itself	Changes form
Brain virus	√		√						
Jerusalem virus		√	√						
Encroacher virus		√	√						
Stealth (a.k.a., IDF) Virus							√		

11



THE UNIVERSITY OF BRITISH COLUMBIA

Computer Worms



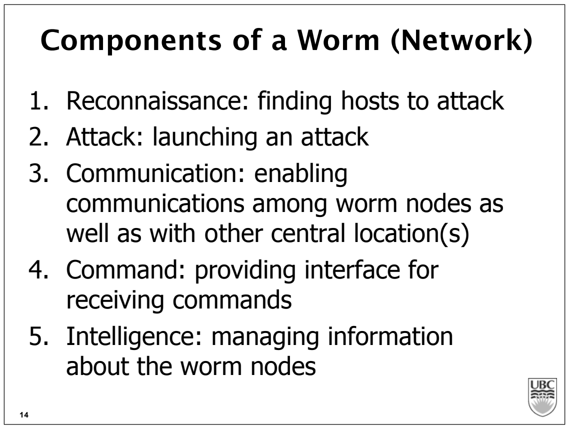
What's a Computer Worm?

"an independently replicating and autonomous infection agent, capable of seeking out new host systems and infecting them via the network"

Jose Nazario in
"Defense and Detection Strategies Against Internet Worms"

What's the difference between computer worms and viruses?

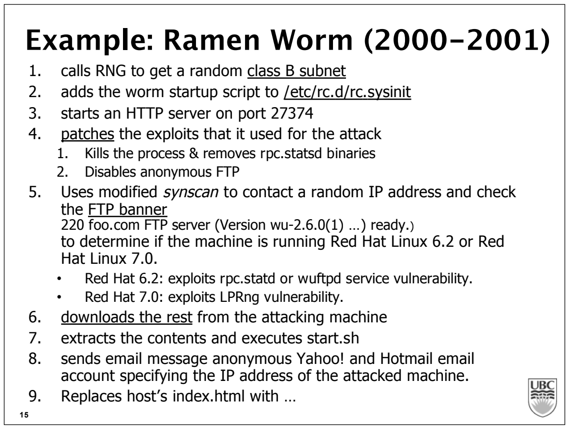
13



Components of a Worm (Network)

1. Reconnaissance: finding hosts to attack
2. Attack: launching an attack
3. Communication: enabling communications among worm nodes as well as with other central location(s)
4. Command: providing interface for receiving commands
5. Intelligence: managing information about the worm nodes

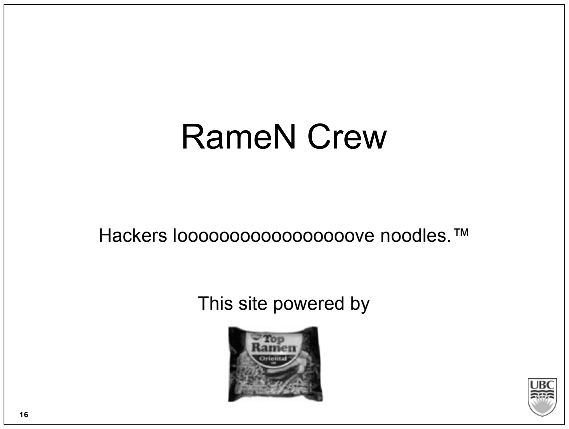
14



Example: Ramen Worm (2000–2001)

1. calls RNG to get a random class B subnet
2. adds the worm startup script to /etc/rc.d/rc.sysinit
3. starts an HTTP server on port 27374
4. patches the exploits that it used for the attack
   1. Kills the process & removes rpc.statsd binaries
   2. Disables anonymous FTP
5. Uses modified *synscan* to contact a random IP address and check the FTP banner
   220 foo.com FTP server (Version wu-2.6.0(1) …) ready.)
   to determine if the machine is running Red Hat Linux 6.2 or Red Hat Linux 7.0.
   - Red Hat 6.2: exploits rpc.statd or wuftpd service vulnerability.
   - Red Hat 7.0: exploits LPRng vulnerability.
6. downloads the rest from the attacking machine
7. extracts the contents and executes start.sh
8. sends email message anonymous Yahoo! and Hotmail email account specifying the IP address of the attacked machine.
9. Replaces host's index.html with …

15

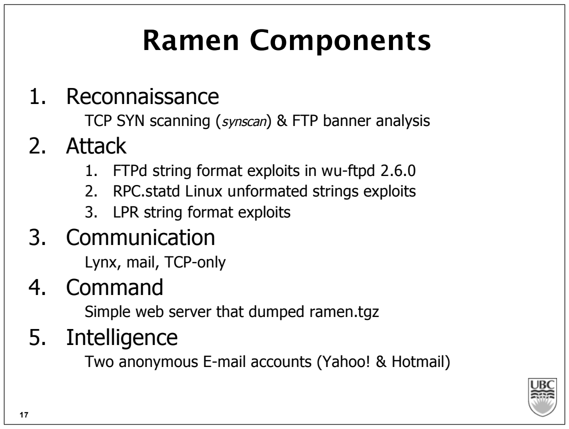


RameN Crew

Hackers looooooooooooooooove noodles.™

This site powered by

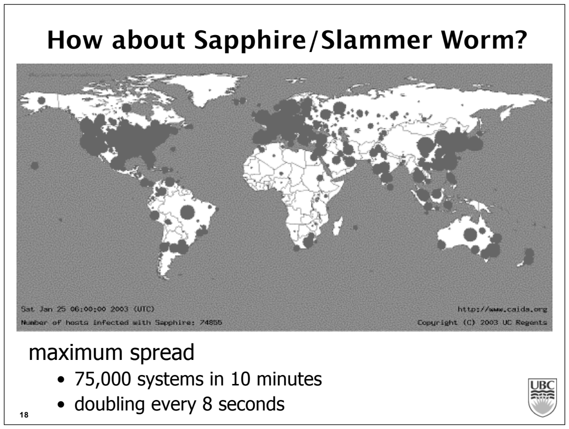
16



Ramen Components

1. Reconnaissance
   TCP SYN scanning (*synscan*) & FTP banner analysis
2. Attack
   1. FTPd string format exploits in wu-ftpd 2.6.0
   2. RPC.statd Linux unformated strings exploits
   3. LPR string format exploits
3. Communication
   Lynx, mail, TCP-only
4. Command
   Simple web server that dumped ramen.tgz
5. Intelligence
   Two anonymous E-mail accounts (Yahoo! & Hotmail)

17



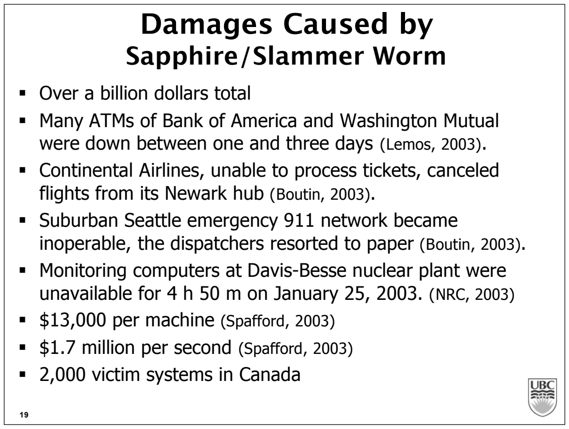
How about Sapphire/Slammer Worm?

Sat Jan 25 06:00:00 2003 (UTC)
Number of hosts infected with Sapphire: 74855
http://www.caida.org
Copyright (C) 2003 UC Regents

maximum spread

- 75,000 systems in 10 minutes
- doubling every 8 seconds

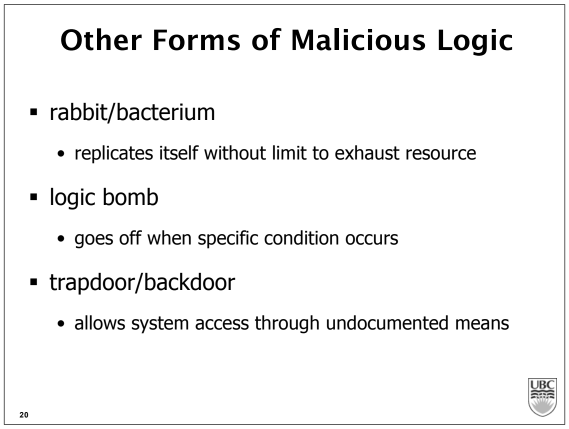
18



Damages Caused by Sapphire/Slammer Worm

- Over a billion dollars total
- Many ATMs of Bank of America and Washington Mutual were down between one and three days (Lemos, 2003).
- Continental Airlines, unable to process tickets, canceled flights from its Newark hub (Boutin, 2003).
- Suburban Seattle emergency 911 network became inoperable, the dispatchers resorted to paper (Boutin, 2003).
- Monitoring computers at Davis-Besse nuclear plant were unavailable for 4 h 50 m on January 25, 2003. (NRC, 2003)
- $13,000 per machine (Spafford, 2003)
- $1.7 million per second (Spafford, 2003)
- 2,000 victim systems in Canada

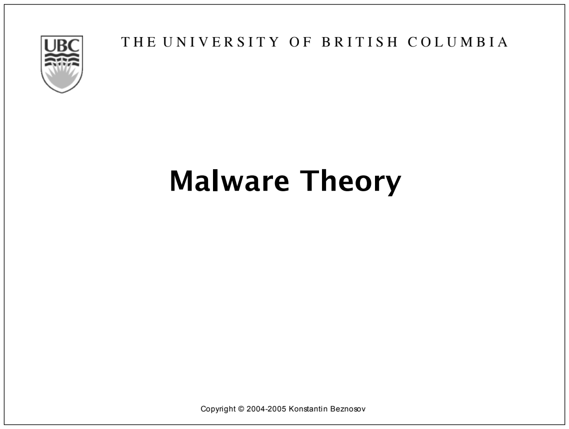
19



Other Forms of Malicious Logic

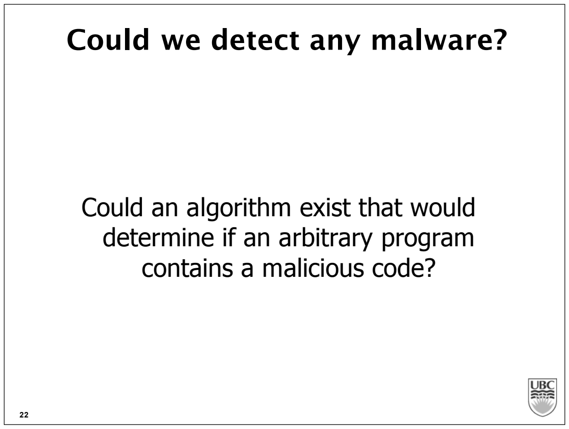
- rabbit/bacterium
  - replicates itself without limit to exhaust resource
- logic bomb
  - goes off when specific condition occurs
- trapdoor/backdoor
  - allows system access through undocumented means

20



THE UNIVERSITY OF BRITISH COLUMBIA

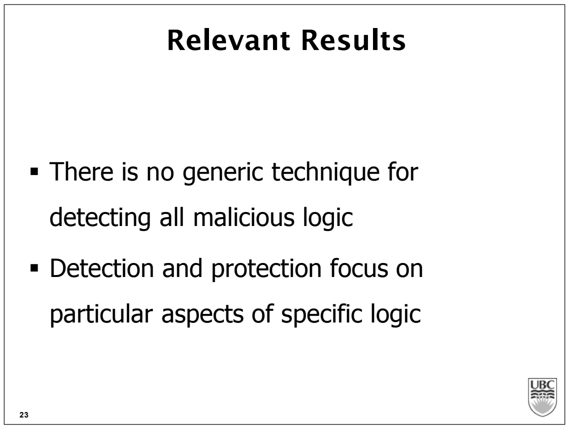
Malware Theory



Could we detect any malware?

Could an algorithm exist that would determine if an arbitrary program contains a malicious code?

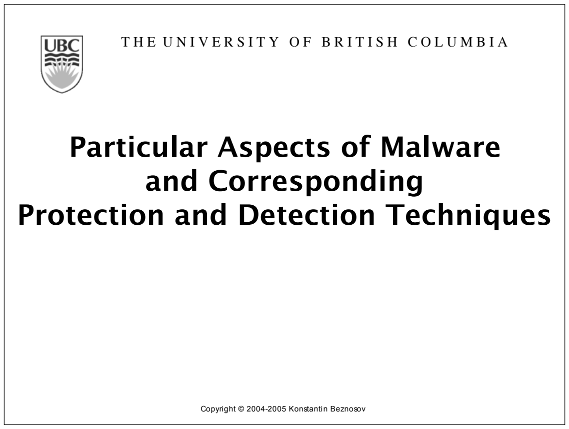
22



Relevant Results

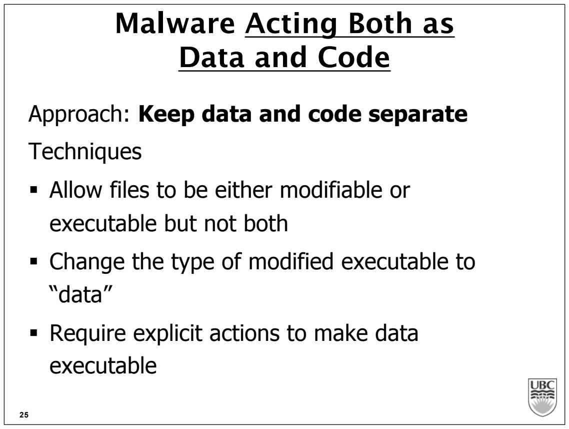
- There is no generic technique for detecting all malicious logic
- Detection and protection focus on particular aspects of specific logic

23



THE UNIVERSITY OF BRITISH COLUMBIA

Particular Aspects of Malware and Corresponding Protection and Detection Techniques



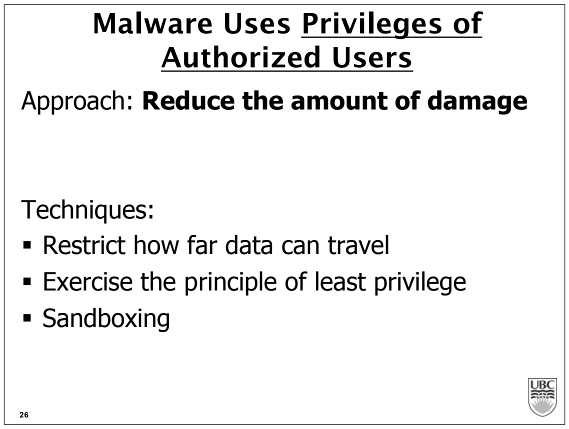
Malware Acting Both as Data and Code

Approach: **Keep data and code separate**

Techniques

- Allow files to be either modifiable or executable but not both
- Change the type of modified executable to “data”
- Require explicit actions to make data executable

25



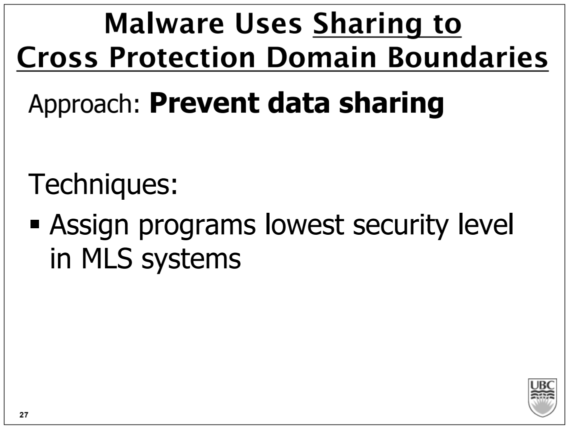
Malware Uses Privileges of Authorized Users

Approach: **Reduce the amount of damage**

Techniques:

- Restrict how far data can travel
- Exercise the principle of least privilege
- Sandboxing

26



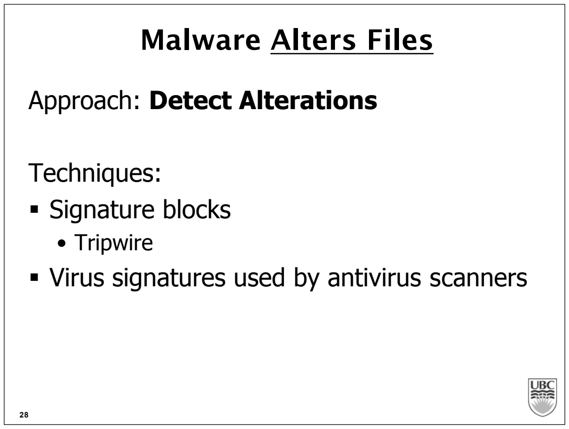
Malware Uses Sharing to Cross Protection Domain Boundaries

Approach: **Prevent data sharing**

Techniques:

- Assign programs lowest security level in MLS systems

27

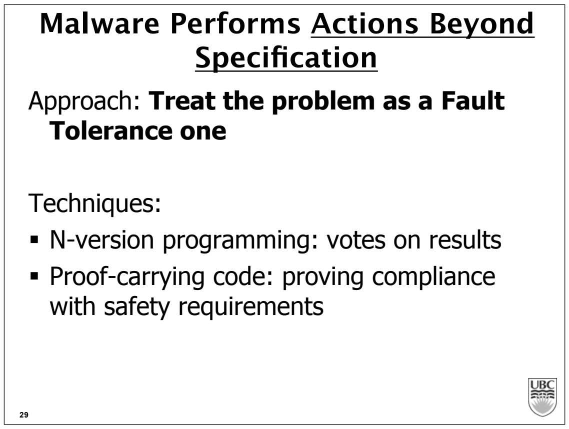


Malware Alters Files

Approach: **Detect Alterations**

Techniques:

- Signature blocks
  - Tripwire
- Virus signatures used by antivirus scanners

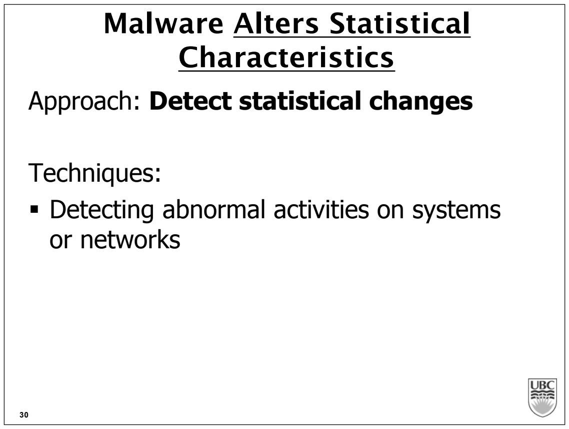


Malware Performs Actions Beyond Specification

Approach: **Treat the problem as a Fault Tolerance one**

Techniques:

- N-version programming: votes on results
- Proof-carrying code: proving compliance with safety requirements

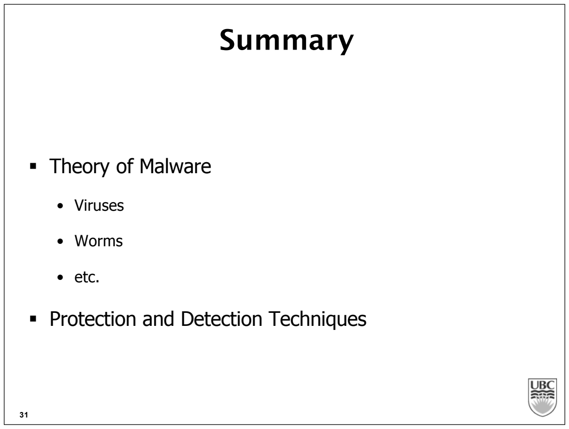


Malware Alters Statistical Characteristics

Approach: **Detect statistical changes**

Techniques:

- Detecting abnormal activities on systems or networks



Summary

- Theory of Malware
  - Viruses
  - Worms
  - etc.
- Protection and Detection Techniques

31