

ECE458/ECE750T27: Computer Security

Programming Security

Dr. Kami Vaniea,
Electrical and Computer Engineering
kami.vaniea@uwaterloo.ca



UNIVERSITY OF
WATERLOO

FACULTY OF
ENGINEERING



First, the news...

- First 5 minutes we talk about something interesting and recent
- You will not be tested on the news part of lecture
- You may use news as an example on tests
- Why do this?
 1. Some students show up late for various good reasons
 2. Reward students who show up on time
 3. Important to see real world examples

INCOMPLETE MEDIATION

Incomplete Mediation

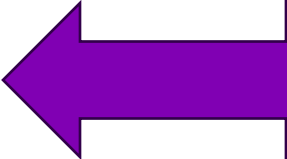
- Buffer overflows are an example of “incomplete mediation”
- Mediation – checking
- Incomplete mediation – failing to check the authorization and properties of a subject/object before using it

Client sends:

`https://exampleShop.com?price=4.99&user=4837&login=true`

Server:

```
function f(price, user, login)
    if (login == true)
        chargeUser(price, user)
```

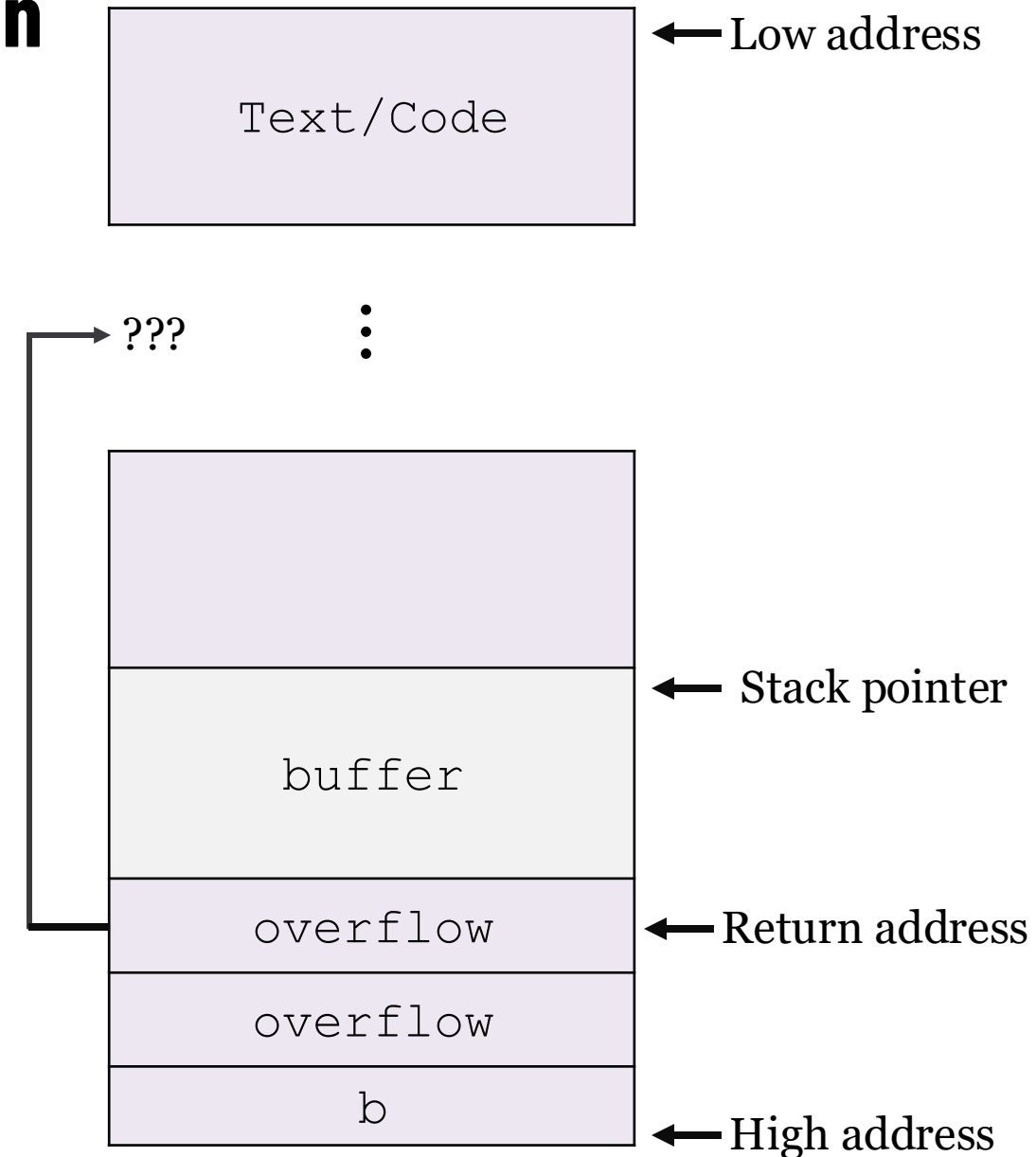


No server-side check if user is logged in. No server-side check if price is appropriate.

Stack smashing - incomplete mediation

- Below code never checks the length of the buffer
- Worse, gets doesn't even have a parameter to state what length of string is expected

```
1  #include <stdio.h>
2  void func(int a, int b) {
3      char buffer[10];
4      gets(buffer);
5      printf("%s", buffer);
6  }
7  void main() {
8      func(1, 4);
9  }
```

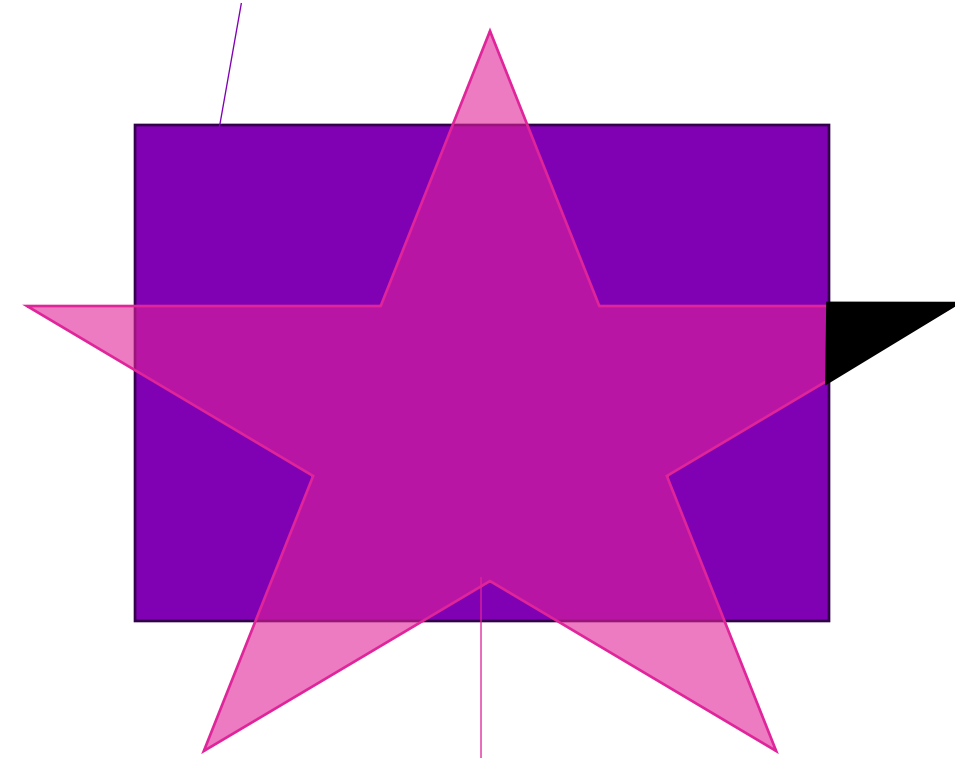


FUZZ TESTING

Testing

- Testing is how we figure out if the specification matches the implementation.
- White Box
 - Tester has access to the full specification and the code
 - Internal test engineer
- Black Box
 - Tester has access to a compiled binary or the physical device only.
 - Could be internal test engineer, could be attacker
- Grey Box
 - Partial information: documentation, algorithm used, knows which open source libraries are being used

Theoretically how the system works



Actually how the system works

Fuzz testing

Normal testing

- Take the specification and test that the program does what it is supposed to do
- Thoughtful and planned
- Skilled test engineer will include out-of-bounds inputs

Incomplete
Mediation

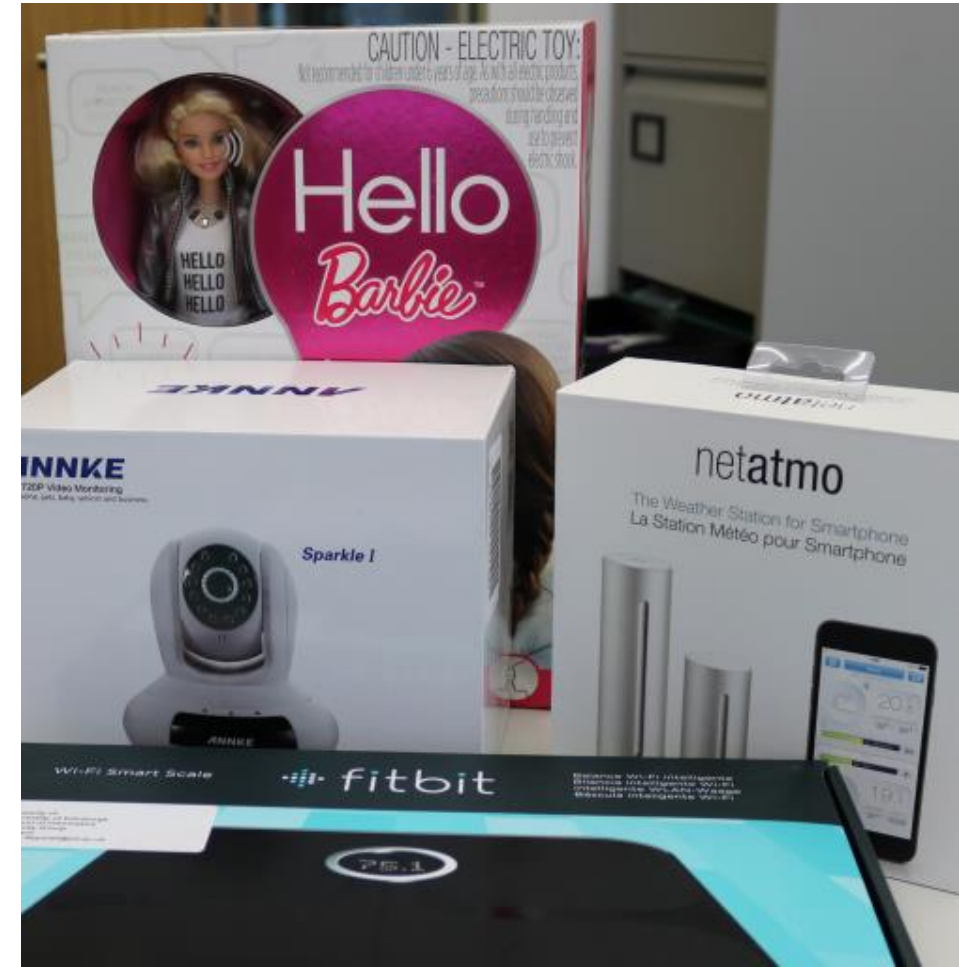


Fuzz testing

- Generate semi-valid random input
- Detailed knowledge of specification not needed
- Black box testing
- Goal is to find situations that crash the program or otherwise cause instability
- Ideally find input that crosses a trust boundary without being checked

Fuzzing: IoT devices

- IoT devices are black boxes
- Fuzz testing can be used to find potential failures
- Which then can be followed-up on manually
- Consumer evaluation groups like Consumer Reports (US) and Which? (UK) use Fuzz testing to evaluate products for “stability”



Volkswagen “defeat device”

- Car emissions are tested using pre-agreed tests
- Defeating these tests just requires detecting that one is happening
- Volkswagen did just that, the car looked for test conditions and then turned on the emission control system
- No test conditions, emission control system reduced to improve car functionality and performance

How Volkswagen’s ‘Defeat Devices’ Worked

By GUILBERT GATES, JACK EWING, KARL RUSSELL and DEREK WATKINS **UPDATED** March 16, 2017

Volkswagen admitted that 11 million of its vehicles were equipped with software that was used to cheat on emissions tests. This is how the technology works and what it now means for vehicle owners. [RELATED ARTICLE](#)

How Did the System Work?

The software sensed when the car was being tested and then activated equipment that reduced emissions, United States officials said. But the software turned the equipment down during regular driving, increasing emissions far above legal limits, most likely to save fuel or to improve the car’s torque and acceleration.

The software was modified to adjust components such as catalytic converters or valves used to recycle some of the exhaust gasses. The components are meant to reduce emissions of nitrogen oxide, a pollutant that can cause [emphysema, bronchitis and other respiratory diseases](#).

Exhaust system of a Volkswagen Golf

Volkswagen has used two basic types of technology to reduce emissions of nitrogen oxides from diesel engines, by either trapping the pollutants or treating them with urea. The first type is shown here.

fuel to allow the trap to work. The car’s **computer** could save fuel by allowing more pollutants to pass through the exhaust system. Saving fuel is one potential reason that Volkswagen’s software could have been altered to make cars pollute more, according to researchers at the International Council on Clean Transportation.

OpenSSL Fuzz Testing -> Heartbleed

- Open SSL was being evaluated by a couple different researchers
- Two groups (Google, Codenomicon) used Fuzz Testing and found what is now known as Heartbleed
- Heartbleed: a vulnerability where a client sends a short string AND the length to the server, and the server returns the string



MALWARE

Malware

- Virus - relies on someone or something else to propagate
- Worm – self propagating
- Trojan horse – appears to be one type of software but has unexpected (bad) functionality
- Trapdoor or Backdoor – allows unauthorized access to a system
- Rabbit – malicious program that exhausts system resources
- Spyware – steals information, often monitor keystrokes

- Virus
- Worm
- Trojan horse
- Trapdoor or backdoor
- Rabbit
- Spyware

10 Steps To Cyber Security



Defining and communicating your Board's Information Risk Management Regime is central to your organisation's overall cyber security strategy. CESG recommend you review this regime - together with the nine associated security areas described below - in order to protect your business against the majority of cyber threats.



SOME FAMOUS MALWARE AND LESSONS LEARNED

Brain Virus (1986)

- Mostly just annoying, non-harmful
- Replaced floppy drive boot sector with itself and installed copies of itself in other places
- Screened disk access to avoid detection
- Each time the disk was read it would check if it was still in the boot sector and if not reinstall itself
- Authors claimed it only infected copyright infringers
- Notable for being one of the first viruses

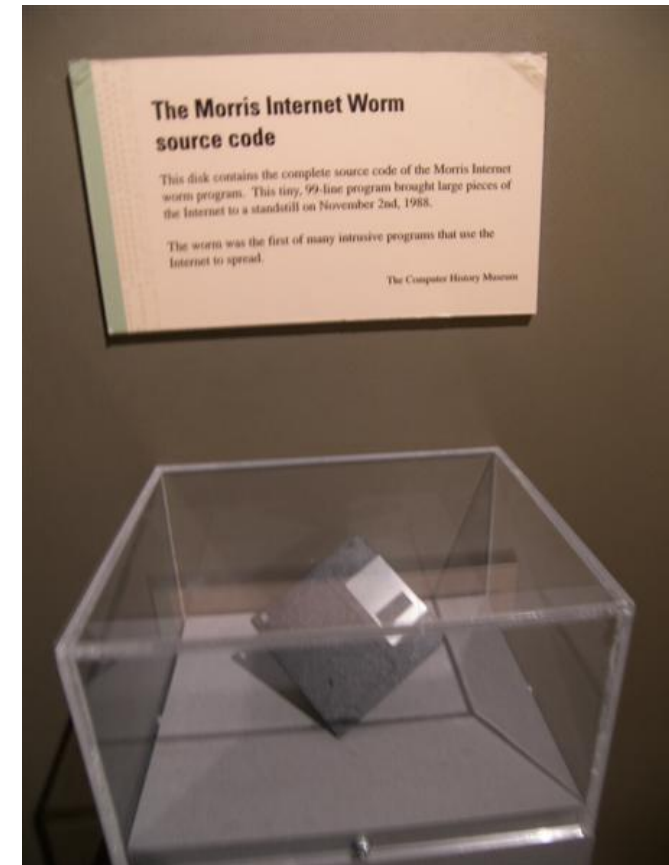
Boot record

Welcome to the Dungeon (c) 1986 Amjads
(pvt) Ltd VIRUS_SHOE RECORD V9.0
Dedicated to the dynamic memories of millions
of viruses who are no longer with us today -
Thanks GOODNESS!!! BEWARE OF THE
er..VIRUS : this program is catching program
follows after these\$#@%\$@!!

Displacement	Hex codes	ASCII value
0000(0000)	FA E9 4A 01 34 12 00 07 14 00 01 00 00 00 20	-0J04↑●Π0
0016(0010)	20 20 20 20 20 20 57 65 6C 63 6F 6D 65 20 74 6F	Welcome to
0032(0020)	20 74 68 65 20 44 75 6E 67 65 6F 6E 20 20 20 20	the Dungeon
0048(0030)	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
0064(0040)	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
0080(0050)	20 28 63 29 20 31 39 38 36 20 42 61 73 69 74 20	(c) 1986 Basit
0096(0060)	26 20 41 6D 6A 61 64 20 28 70 76 74 29 20 4C 74	& Amjad (put) Lt
0112(0070)	64 2E 20 20 20 20 20 20 20 20 20 20 20 20 20	d.
0128(0080)	20 42 52 41 49 4E 20 43 4F 4D 50 55 54 45 52 20	BRAIN COMPUTER
0144(0090)	53 45 52 56 49 43 45 53 2E 2E 37 33 30 20 4E 49	SERVICES.. 730 NI
0160(00A8)	5A 41 4D 20 42 4C 4F 43 4B 20 41 4C 4C 41 4D 41	ZAM BLOCK ALLAMA
0176(00B0)	20 49 51 42 41 4C 20 54 4F 57 4E 20 20 20 20 20	.IQBAL TOWN
0192(00C0)	20 20 20 20 20 20 20 20 20 20 20 4C 41 48 4F 52	LAHDR
0208(00D0)	45 2D 50 41 4B 49 53 54 41 4E 2E 2E 50 48 4F 4E	E-PAKISTAN..PHJN
0224(00E0)	45 20 3A 34 33 30 37 39 31 2C 34 34 33 32 34 3B	E :430791,443248
0240(00F0)	2C 32 38 30 35 33 30 2E 20 20 20 20 20 20 20	,280530.

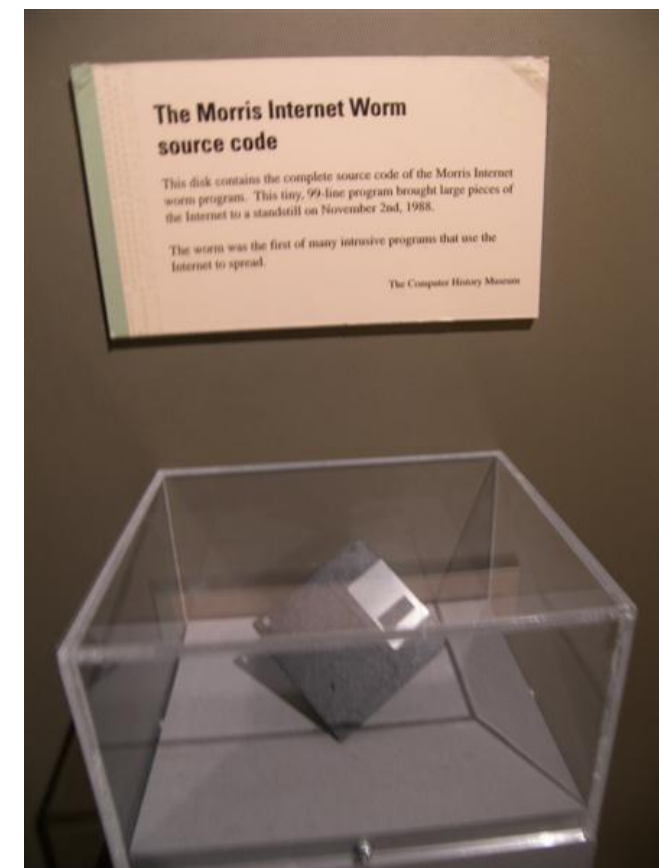
Morris Worm (1988)

- Written as a curiosity but had a flaw (according to Morris)
- Designed to:
 1. Determine where it could spread
 2. Spread infection wherever possible
 3. Remain undiscovered
- Exploited several vulnerabilities
 - Weak passwords – password guessing
 - Hole in the debug mode of Unix sendmail (patch available)
 - Buffer overflow hole in “fingerd”
- Sent 99 lines of C-code to victim which then downloaded the rest



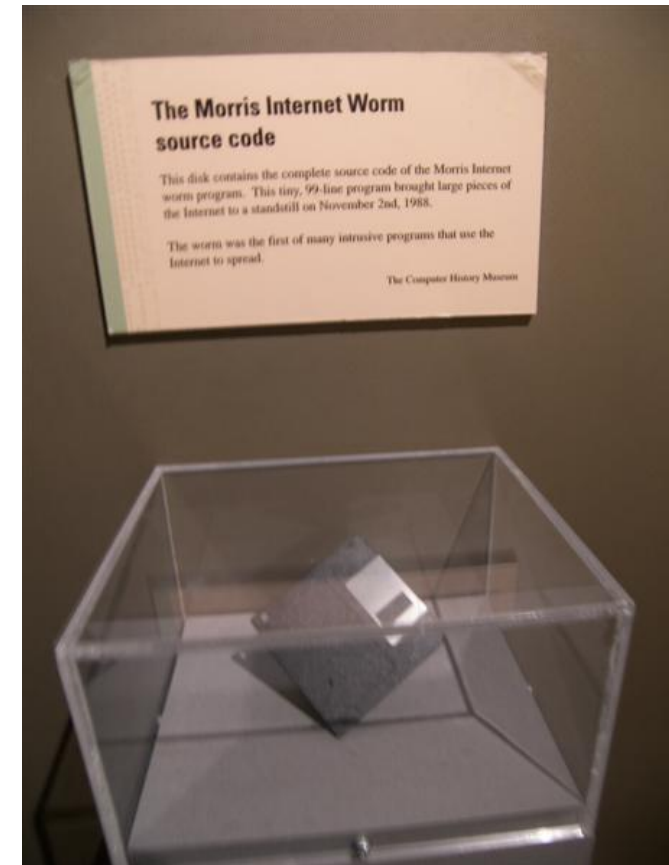
Morris Worm - also unintentionally a rabbit

- Worked hard to not be discovered
 - If transmission interrupted – self deleted
 - Code encrypted when downloaded (rare)
 - Downloaded code deleted after compilation
 - Periodically changed process name and PID
- Was supposed to periodically check if a system was infected before trying to infect it
 - But... that check was not always done
 - So re-infection of infected systems resulted in memory exhaustion (rabbit)



Morris Worm - important because

- Internet was supposed to survive nuclear attack. But a grad student took it down with some C code
- More damage caused by people panicking and unplugging their systems
- Lead to the creation of the Computer Emergency Response Team (CERT) at Carnegie Mellon University
 - CERTs now exist world-wide
- Sadly the Morris worm didn't result in a re-design of the internet protocols with more security in mind....



Code Red (2001) - Worm

- Infected more than 300,000 computers in about 14 hours
- Exploited a buffer overflow in Microsoft IIS server software
- The monitored traffic on port 80 looking for more possible victims
- Notable for how fast it spread across the internet
- Infection did different things on different days of the month
 - 1-19 – spread infection
 - 20-27 – attempt DDoS attack on www.whitehouse.gov

Code red – case study in update issues

- Microsoft quickly released a patch to fix Code Red
- But the patch had an error, the second patch fixed the first, but had a smaller error
- Third patch finally fixed the error
- Most systems installed first and second patch but not the third

SQL Slammer Worm (2003)

- Infected 75,000 systems in 10 minutes
 - At peak infections doubled every 8.5 seconds
- Fit into one 376-byte UDP packet
- Each infected computer randomly generated IP addresses and tried to infect them
- Firewalls let the single UDP packet through
- Firewalls at the time were often setup to let random packets through and then monitor the connection
- No one thought working code could fit into 376 bytes

Stuxnet (2005)

- Advanced information warfare
- Deliberately designed to disrupt Iranian nuclear fuel processing
- It infected nuclear refinement center and caused an expensive part (centrifuge) to wear out quickly
- Used 4 unpatched Windows vulnerabilities
- Compromised an “air gapped” system by infecting USB drives that were moved across the air gap



NotPetya (2017)

- A small Ukraine company that makes tax software got infected
- The attacker then modified their tax software code and pushed an automatic update
- That update was installed world-wide
- Used EternalBlue (developed by USA) to spread as a worm
- Pretended to be ransomware but was likely political

<https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>

ANDY GREENBERG

EXCERPT

SECURITY AUG 22, 2018 5:00 AM

The Untold Story of NotPetya, the Most Devastating Cyberattack in History

Crippled ports. Paralyzed corporations. Frozen government agencies. How a single piece of code crashed the world.

IT WAS A perfect sunny summer afternoon in Copenhagen when the world's largest shipping conglomerate began to lose its mind.

The headquarters of A.P. Møller-Maersk sits beside the breezy, cobblestoned esplanade of Copenhagen's harbor. A ship's mast carrying the Danish flag is planted by the building's northeastern corner, and six stories of blue-tinted windows look out over the water, facing a dock where the Danish royal family parks its yacht. In the building's basement, employees can browse a corporate gift shop, stocked with Maersk-branded bags and ties, and even a rare Lego model of the company's gargantuan Triple-E container ship, a vessel roughly as large as the Empire State Building laid on its side, capable of carrying another Empire State Building-sized load of cargo stacked on top of it.

That gift shop also houses a technology help center, a single desk manned by IT troubleshooters next to the shop's cashier. And on the afternoon of June 27, 2017, confused Maersk staffers began to gather at that help desk in twos and threes, almost all of them carrying laptops. On the machines' screens were messages in red and black lettering. Some read "repairing file system on C:" with a stark warning not to turn off the computer. Others, more surreally, read "oops, your important files are encrypted!" and demanded payment of \$200 worth of

Trojan Example

- Trojans look like normal software but are not
- Imagine downloading “freeMusic.mp3”
- Then double clicking the icon
- Expected behavior: iTunes opens and plays music
- Actual behavior: iTunes opens and plays weird laughter then pop-up appears



freeMusic.mp3

Yep, this is an application

What is your iTunes playing now?

OK

Trojan Example - what happened?

- Things like icons and file extensions are mostly for people, access control is not linked to them
- This file has the mp3 icon set
- It is an executable
 - File extensions are not always shown, or a vulnerability can cause an override of default file extension actions



freeMusic.mp3

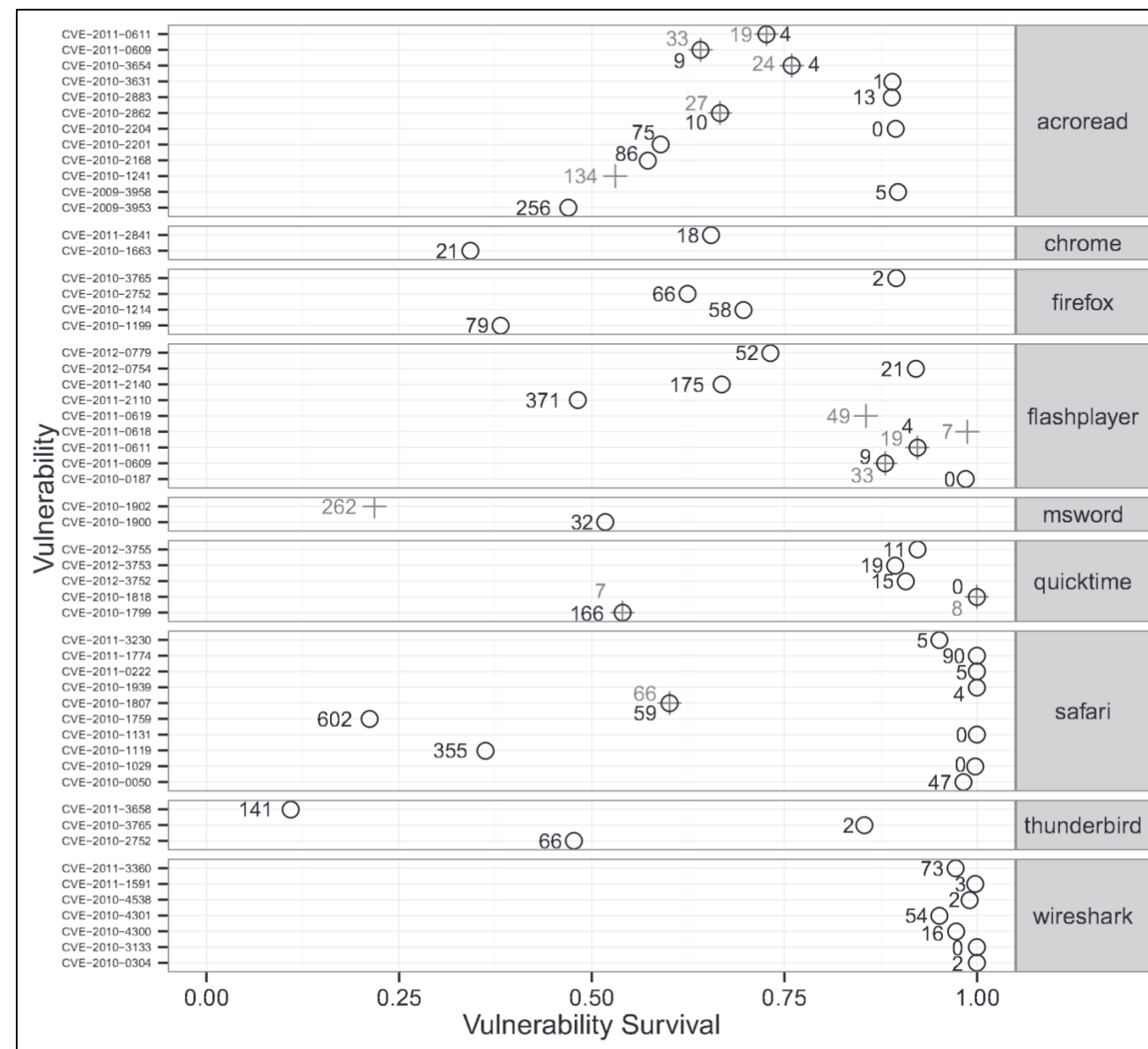
Yep, this is an application

What is your iTunes playing now?

OK

There are many copies of software on a computer

- Webkit – for example – used to be the main engine for Safari AND iTunes
- Users might update Safari because bad things come from the internet
- But they might not update iTunes because its “just a music program”
- Trojan can then use Webkit vulnerability via iTunes



Think-pair-share

- Patching is widely accepted to be a best practice
- Bad software often targets unpatched systems
- What makes patching risky?
- How might those risks be mitigated? (Not just testing.)

Trojans common in phishing



PROTECTED VIEW

Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

Enable Editing



- Complex programs like Microsoft Word use things like macros to create trojan type behavior
- To protect you, programs auto turn off uncommon features, like macros, to limit chance of damage from attachments
- Editing also turned off
- Kami's opinion: blame-the-user security rather than actually fixing the software. Just viewing a Word document should really never infect a computer!

Trojans in peer to peer (P2P) file sharing (2024)

- Bandwidth costs money
 - Even “unlimited” plans typically have secret limitations
 - Companies like to get rid of their heavy users
- Korean telecom infecting heavy users with malware to slow their usage of file sharing

 SIGN IN / UP

The  Register®

SECURITY 7 

Korean telco allegedly infected its P2P users with malware

KT may have had an entire team dedicated to infecting its own customers

 [Laura Dobberstein](#)

Thu 27 Jun 2024 // 01:46 UTC

A South Korean media outlet has alleged that local telco KT deliberately infected some customers with malware due to their excessive use of peer-to-peer (P2P) downloading tools.

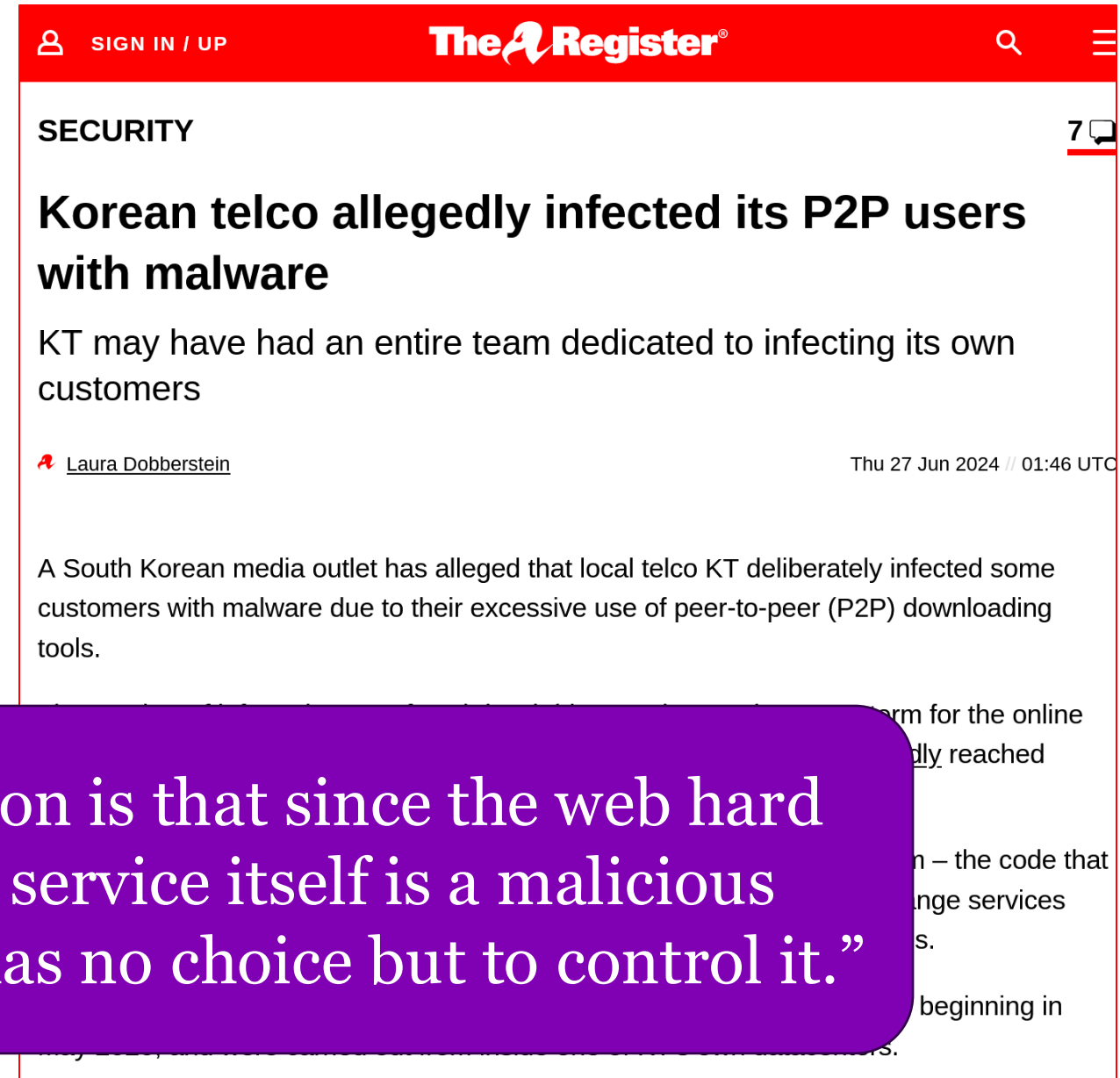
The number of infected users of “web hard drives” – the South Korean term for the online storage services that allow uploading and sharing of content – has reportedly reached 600,000.

Malware designed to hide files was allegedly inserted into the Grid Program – the code that allows KT users to exchange data in a peer-to-peer method. The file exchange services subsequently stopped working, leading users to complain on bulletin boards.

The throttling shenanigans were reportedly ongoing for nearly five months, beginning in May 2020, and were carried out from inside one of KT's own datacenters.

Trojans in peer to peer (P2P) file sharing (2024)

- Bandwidth costs money
 - Even “unlimited” plans typically have secret limitations
 - Companies like to get rid of their heavy users
- Korean telecom infecting heavy users with malware to slow their usage of file sharing



The screenshot shows a news article from The Register. The header is red with 'The Register' logo and a search icon. Below the header, the article is categorized under 'SECURITY'. The title is 'Korean telco allegedly infected its P2P users with malware'. The sub-headline reads 'KT may have had an entire team dedicated to infecting its own customers'. The author is 'Laura Dobberstein' and the date is 'Thu 27 Jun 2024 // 01:46 UTC'. The main text states: 'A South Korean media outlet has alleged that local telco KT deliberately infected some customers with malware due to their excessive use of peer-to-peer (P2P) downloading tools.'

“KT's position is that since the web hard drive P2P service itself is a malicious program, it has no choice but to control it.”

Disney employee installs “fun AI” loses Disney

- Disney employee on his *home* computer downloaded a free AI tool
- It didn't work, so he deleted it
- It was really a trojan, and it happily broke into his OnePassword account
- OnePassword included some of his Disney employee passwords
- Attackers then took more than a TB from internal Disney Slack and published it online

Hackers stole this engineer's 1Password database. Could it happen to you?

A software engineer for the Disney Company unwittingly downloaded a piece of malware that turned his life upside down. Was his password manager to blame?



Written by **Ed Bott**, Senior Contributing Editor

Feb. 27, 2025 at 2:00 a.m. PT



rob dobi/Getty Images

Here's the very definition of a nightmare scenario.

In February 2024, Matthew Van Andel downloaded a free AI tool on the computer in his home office. Five months later, the Southern California-based engineer learned that the app included an unwelcome extra

MALWARE DETECTION

Signature Detection

- Relies on finding patterns – older method and common for anti-virus programs
 - Known malware is analyzed to find commonalities
 - Commonalities are converted into signature patterns which the anti-virus software looks for
 - Files with the pattern are then further analyzed to limit false positives
- Example: w32/Beast virus always contains the code:
 - 83EB 0274 EB0E 740A 81EB 0301 000

Security intelligence

Microsoft Defender Antivirus uses security intelligence to detect threats. We try to automatically download the most recent intelligence to protect your device against the newest threats. You can also manually check for updates.

Security intelligence version: 1.413.558.0

Version created on: 6/27/2024 8:29 AM

Last update: 6/27/2024 6:48 PM

Check for updates

Change Detection

- Monitor files for unexpected changes
- Malware code must live somewhere, so monitor common places it might be
 - Heartland data breach, the majority of the malware was in unallocated memory
- Look for weird anomalies
 - Inode numbers (unique file ID) on critical system files should be sequential

proofpoint.

**Heartland**
PAYMENT SYSTEMS®

The Highest Standards | The Most Trusted Transactions

What's the worst thing that could happen in a data breach? If you said millions of dollars in losses, a business forced to go on hiatus, scores of compliance violations and tons of bad press, then you might have worked at [Heartland Payment Systems back in 2008](#). (Although the breaches took place over several months in 2008, the company did not go public with the findings until January 2009.)

The Fortune 1000 company, which specializes in payment, point-of-sale and payroll systems, suffered one of the worst data breaches in history. Here's a quick recap of the breach:

The Details

The company was first notified by Visa and MasterCard in October 2008 about suspicious transactions stemming from accounts Heartland processed. Suspecting a cyber attack, Heartland hired cybersecurity forensics experts to investigate the issue. It took more than two months to unravel the mystery.

Anomaly Detection

- Looking for inconsistencies in behavior
- Labor intensive – a technical professional needs to check up on identified issues
- Good way to find unknown malware and other issues
- Intrusion Detection Systems for networking
- File access patterns
- Process creation patterns

Anomaly Detection: Machine Learning

- Machine learning is a natural match for anomaly detection
- It is good at looking at “normal” traffic patterns and then flagging cases where traffic seems odd



Introducing VirusTotal Code Insight: Empowering threat analysis with generative AI

MONDAY, APRIL 24, 2023 | [BERNARDO.QUINTERO](#) | [LEAVE A COMMENT](#)

At the RSA Conference 2023 today, we are excited to unveil VirusTotal Code Insight, a cutting-edge feature that leverages artificial intelligence for code analysis. Powered by Google Cloud [Security AI Workbench](#), Code Insight produces natural language summaries of code snippets with ease. This functionality empowers security experts and analysts by providing them with deeper insights into the purpose and operation of analyzed code, significantly enhancing their capability to detect and mitigate potential threats.

For quite some time, artificial intelligence (AI) and machine learning (ML) have played a crucial role in anti-malware and cybersecurity, mainly focusing on classification tasks. However, recent advancements in large language models (LLMs) have expanded their capabilities to encompass text generation and summarization.

Impressively, when these models are trained on programming languages, they can adeptly transform code into natural language explanations. This innovation not only expedites malware analysis but also bolsters a variety of cybersecurity applications. Recognizing the immense potential of this cutting-edge technology, we have incorporated it into the VirusTotal platform, significantly enhancing its capabilities.

Target data breach (2013)

- Target had a point-of-sale compromise (lost credit card data) right before Christmas
- Third party HVAC vendor had network access, and they had bad security
- Target failed to respond to multiple automated warnings from the anti-intrusion software about malware install
- Also ignored warnings from network anti-intrusion systems as data was exfiltrated

Executive Summary

In November and December 2013, cyber thieves executed a successful cyber attack against Target, one of the largest retail companies in the United States. The attackers surreptitiously gained access to Target's computer network, stole the financial and personal information of as many as 110 million Target customers, and then removed this sensitive information from Target's network to a server in Eastern Europe.

This report presents an explanation of how the Target breach occurred, based on media reports and expert analyses that have been published since Target publicly acknowledged this breach on December 19, 2013. Although the complete story of how this breach took place may not be known until Target completes its forensic examination of the breach, facts already available in the public record provide a great deal of useful information about the attackers' methods and Target's defenses.

This report analyzes what has been reported to date about the Target data breach, using the "intrusion kill chain" framework, an analytical tool introduced by Lockheed Martin security researchers in 2011, and today widely used by information security professionals in both the public and the private sectors. This analysis suggests that Target missed a number of opportunities along the kill chain to stop the attackers and prevent the massive data breach. Key points at which Target apparently failed to detect and stop the attack include, but are not limited to, the following:

- Target gave network access to a third-party vendor, a small Pennsylvania HVAC company, which did not appear to follow broadly accepted information security practices. The vendor's weak security allowed the attackers to gain a foothold in Target's network.
- Target appears to have failed to respond to multiple automated warnings from the company's anti-intrusion software that the attackers were installing malware on Target's system.
- Attackers who infiltrated Target's network with a vendor credential appear to have successfully moved from less sensitive areas of Target's network to areas storing consumer data, suggesting that Target failed to properly isolate its most sensitive network assets.
- Target appears to have failed to respond to multiple warnings from the company's anti-intrusion software regarding the escape routes the attackers planned to use to exfiltrate data from Target's network.

No-fault

- Areas like aviation and nautical take a “no fault” view towards incident management
- The sectors consider it more important to learn from problems than to blame and punish

Boeing Plane Incidents Timeline: Full List of 9 Issues in 3 Months

Published Mar 25, 2024 at 10:54 AM EDT

Updated Mar 25, 2024 at 11:15 AM EDT

By **Thomas Kika**
Weekend Staff Writer

FOLLOW



The Boeing aviation company announced on Monday that current CEO Dave Calhoun intends to step down by the end of the year.

This move came after months of incidents involving the company's planes have created a bruising PR nightmare, causing widespread uncertainty among the general public about the quality of Boeing planes and air travel in general. Larry Kellner, chairman of the board for Boeing, will also not be seeking reelection and will be replaced by Steve Mollenkopf, the former CEO of Qualcomm and a Boeing board member since 2020, with the search ongoing for a replacement for Calhoun.

While the company has weathered incidents and accidents with its planes at various points in the past, its current troubles flared up after a January incident involving an Alaska Airlines flight that went viral online. [Every incident involving troubles with a Boeing craft since then](#) has received renewed and intense public scrutiny.

Newsweek reached out to Boeing via email on Monday morning for comment.

Here is a look at the incidents on Boeing planes since January.

Alaska Airlines, January 5

On this date, a Boeing 737 Max 9 craft operated by Alaska Airlines was forced to make an emergency landing shortly after departing Portland International Airport after a shoddily installed door plug flew off in midair. Images from inside the craft showing the sizable opening that had been left in the craft went viral online, causing widespread alarm, though no one on board had been injured.

Think-pair-share

- Why is no-fault not used in computer security?

Collaboration

- Defenders do need to collaborate
- No one likes viruses, malware, or phishing, so these are “safe” to share
- Organizations like VirusTotal and APWG collect together examples of such bad things and share those examples with everyone
- Company privacy costs money:
“By submitting data [you agree] to the sharing of your Sample submission with the security community.”



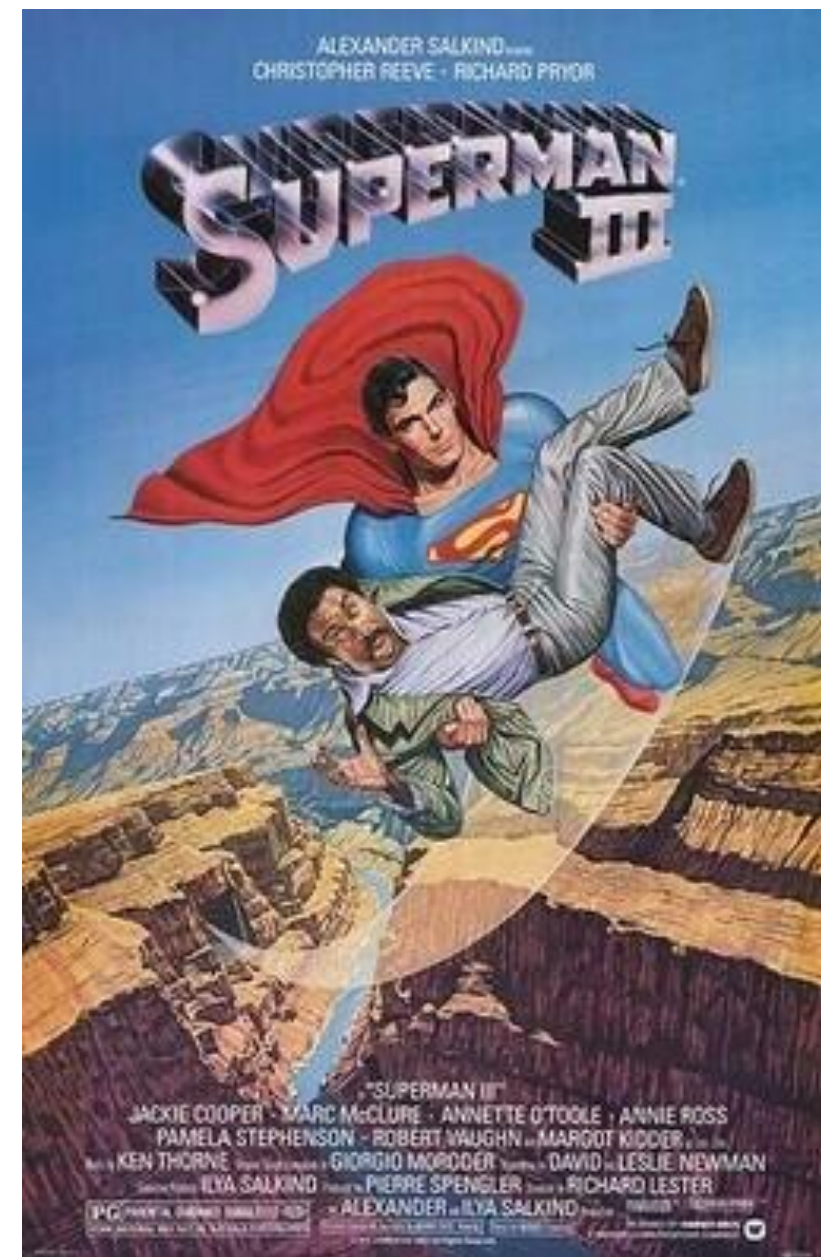
Attacking from the inside

- Not all attacks are from external, some are from employees
- Castle security thinking does not work well with insider attacks



Salami Attacks

- Attack that takes only a small amount of the resource each time to avoid being noticed
- Program bank software so that money that is rounded off is moved to a bank account



QUESTIONS