


COMP 4384 Software Security

Module 1: *History of Computer Security*

Ahmed Tamrawi

 [atamrawi](#)  [atamrawi.github.io](#)  ahmedtamrawi@gmail.com

Security is a journey, **not a destination!**



Introduction

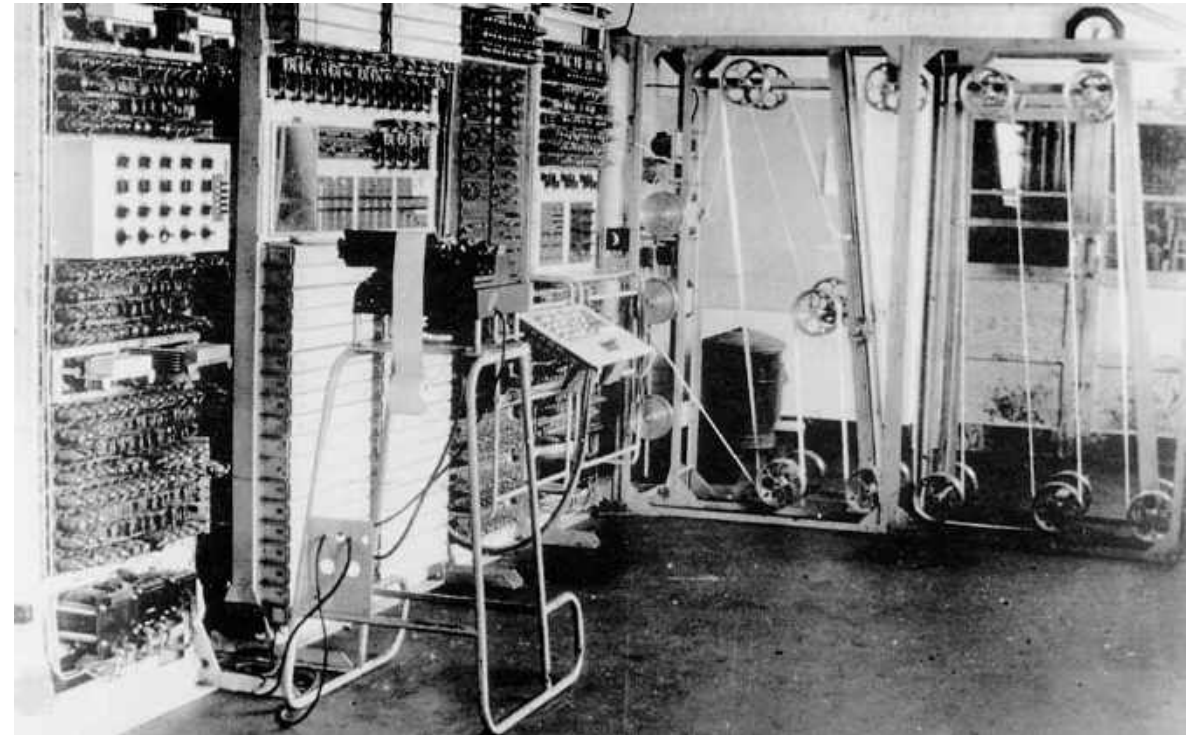
- Computer security has been travelling for forty years.
- New security challenges arise when new – or old – technologies are put to new use.
- The challenges faced have kept changing. So have the answers to familiar challenges.
- Security mechanisms must be seen in the context of the IT landscape they were developed for.

Epochs

- 1930s: people as “computers”
- 1940s: first electronic computers
- 1950s: start of an industry
- 1960s: software comes into its own
- 1970s: age of the mainframe
- 1980s: age of the PC
- 1990s: age of the Internet
- 2000s: age of the Web

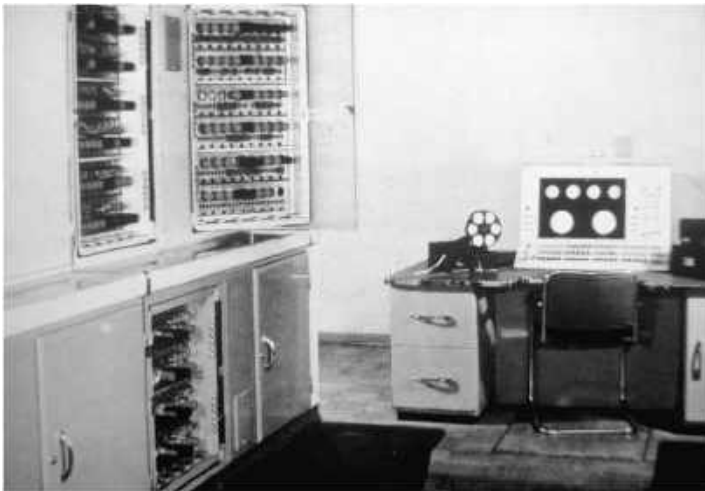
Colossus – Computer built to break Hitler's Codes

- The code breakers at Bletchley Park pioneered the use of electronic programmable computers during World War II.
- It was used by British codebreakers to help read encrypted German messages during World War II in 1944.



The Dawn of Computer Security

- The first electronic computers were built in the 1940s (Colossus, EDVAC, ENIAC) and found applications in academia (Ferranti Mark I, University of Manchester), commercial organizations (LEO, J. Lyons & Co.), and government agencies (Univac I, US Census Bureau) in the early 1950s.



Ferranti Mark I, University of Manchester



LEO – The World's First Business Computer



Univac I - US Census Bureau

The Dawn of Computer Security

- Computer security can trace its origins back to the 1960s when **multi-user systems emerged**, needing mechanisms for *protecting the system from its users*, and *the users from each other*.
- Two reports in the early 1970s signal the start of computer security as a field of research in its own right.
 - The RAND report by Willis Ware summarized the technical foundations computer security had acquired by the end of the 1960s.
 - The Ware's report was followed shortly by Anderson's report that laid out a research program for the design of secure computer systems, dominated by the requirement of protecting classified information.



Willis H. Ware. Security controls for computer systems. Technical Report R-609, The RAND Corporation, Santa Monica, CA, January 1970.

URL: <https://www.rand.org/pubs/reports/R609-1.html>

The RAND Report by Willis Ware, 1970

- The RAND report by Willis Ware summarized the technical foundations computer security had acquired by the end of the 1960s.
- The report also produced a detailed analysis of the **policy requirements of one particular application area, the protection of classified information in the US defense sector.**



Willis H. Ware. Security controls for computer systems. Technical Report R-609, The RAND Corporation, Santa Monica, CA, January 1970.
URL: <https://www.rand.org/pubs/reports/R609-1.html>

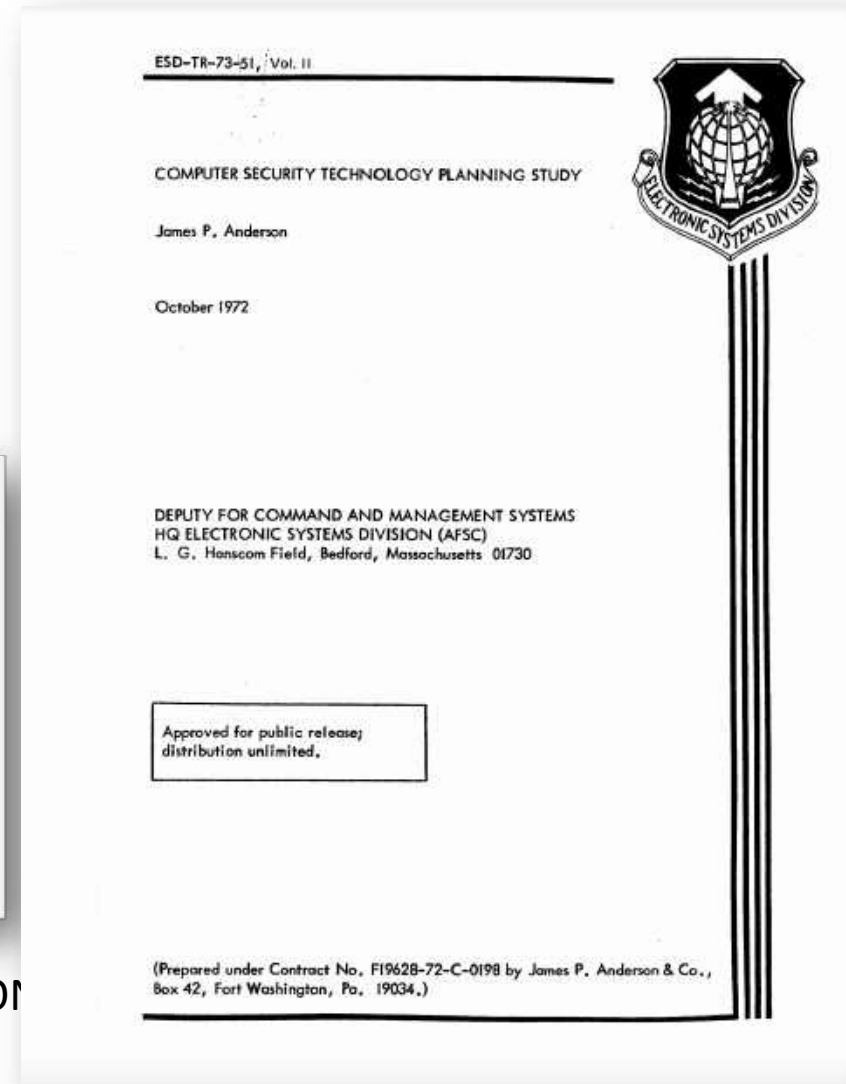


The Anderson Report, 1972

- Anderson's report laid out a **research program for the design of secure computer systems**, dominated by the **requirement of protecting classified information**.

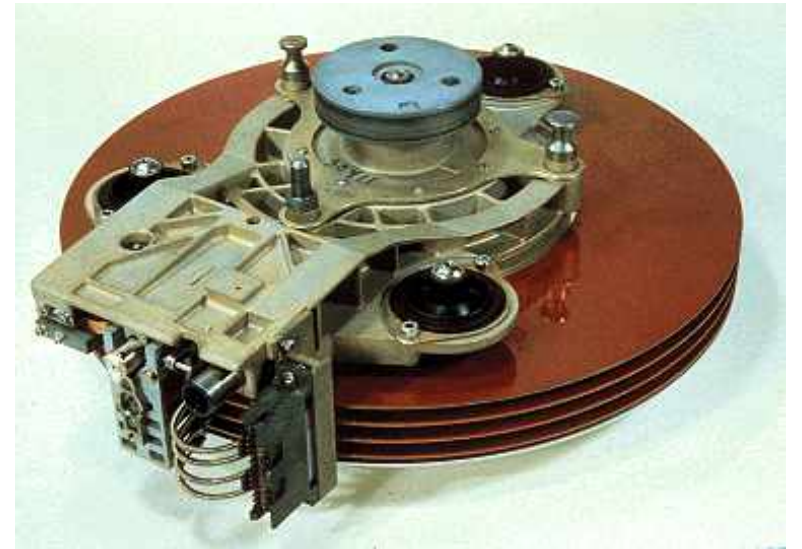
In recent years the Air Force has become increasingly aware of the problem of computer security. This problem has intruded on virtually any aspect of USAF operations and administration. The problem arises from a combination of factors that includes: greater reliance on the computer as a data processing and decision making tool in sensitive functional areas; the need to realize economies by consolidating ADP resources thereby integrating or co-locating previously separate data processing operations; the emergence of complex resource sharing computer systems providing users with capabilities for sharing data and processes with other users; the extension of resource sharing concepts to networks of computers; and the slowly growing recognition of security inadequacies of currently available computer systems.

Anderson, James P. Computer security technology planning study. ANDERSON (JAMES P) AND CO FORT WASHINGTON PA FORT WASHINGTON, 1972.
URL: <http://seclab.cs.ucdavis.edu/projects/history/papers/ande72.pdf>



1970s: Mainframes – Data Crunchers

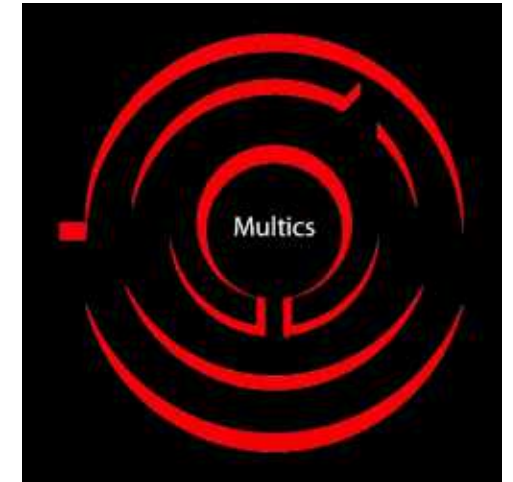
- **Technology:** Winchester disk (IBM) 35-70 megabytes memory.
- **Application:** data crunching in large organizations and government departments.
- Protection of classified data in the defense sector dominates security research and development.
- Social security applications and the like.
- **Security controls in the system core:** operating systems, database management systems
- Computers and computer security managed by professionals.



IBM's Winchester disk was a removable cartridge, but the heads and platters were built in a sealed unit and were not separable.

1970s: Security Issues

- Military applications:
 - Anderson report
 - Multi-level security (MLS)
 - Bell La-Padula model
- **Status today:** High assurance systems developed (e.g. Multics) but do not address today's issues.
- Non-classified but sensitive applications
 - DES, public research on cryptography
 - Privacy legislation
 - Statistical database security
- **Status today:** cryptography is a mature field, statistical database security reappearing in data mining.



1980s: PCs – Office Workers

- **Technology:** Personal Computer, GUI, mouse, ...
- **Application:** word processors, spreadsheets, i.e. office work.
- Liberation from control by the IT department.
- Single-user machines processing unclassified data: No need for multi-user security or for MLS.
- **Risk analysis:** no need for computer security.
- **Security evaluation:** Orange Book (TCSEC, 1983/85): Driven by the defence applications of the 1970s.

1980s: Security Issues

- Research on MLS systems still going strong; Orange Book, MLS for relational databases.
- **Clark-Wilson model**: first appearance of “commercial security” in mainstream security research.
- Worms and viruses: research proposals, before appearing in the wild.
 - Also the worm comes from Xerox park (1982) ...
- Intel 80386 drops support for segmentation.

1990s: Internet – Surfers Paradise?

- **Technology:** Internet, commercially used, Web 1.0.
- **Applications:** Web surfing, email, entertainment
- Single-user machine that had lost its defenses in the previous decade now exposed to “hostile” Internet.
- No control on who can send what to a machine on the Internet.
- **Buffer overrun attacks:**
 - Aleph One (1996): Smashing the Stack for Fun and Profit
- **“Add-on” security controls: firewalls, SSL, browser sandbox as reference monitor, etc.**

SMASHING THE STACK FOR FUN AND PROFIT

by Aleph One <aleph1@underground.org>

published in Phrack Volume 7, Issue 49

File 14 of 16

'smash the stack' [C programming] n. On many C implementations it is possible to corrupt the execution stack by writing past the end of an array declared auto in a routine. Code that does this is said to smash the stack, and can cause return from the routine to jump to a random address. This can produce some of the most insidious data-dependent bugs known to mankind. Variants include trash the stack, scribble the stack, mangle the stack; the term mung the stack is not used, as this is never done intentionally. See spam; see also alias bug, fandango on core, memory leak, precedence lossage, overrun screw.

INTRODUCTION

Over the last few months there has been a large increase of buffer overflow vulnerabilities being both discovered and exploited. Examples of these are syslog, splitvt, sendmail 8.7.5, Linux/FreeBSD mount, Xt library, at, etc. This paper attempts to explain what buffer overflows are, and how their exploits work.

Basic knowledge of assembly is required. An understanding of virtual memory concepts, and experience with gdb are very helpful but not necessary. We also assume we are working with an Intel x86 CPU, and that the operating system is Linux.

Some basic definitions before we begin: A buffer is simply a contiguous block of computer memory that holds multiple instances of the same data type. C programmers normally associate with the word buffer arrays. Most commonly, character arrays. Arrays, like all variables in C, can be declared either static or dynamic. Static variables are allocated at load time on the data segment. Dynamic variables are allocated at run time on the stack. To overflow is to flow, or fill over the top, brims, or bounds. We will concern ourselves only with the overflow of dynamic buffers, otherwise known as stack-

One, A. "Smashing the stack for fun and profit. Phrack Magazine, 7 (49), Nov. 8, 1996." DOI= <http://insecure.org/stf/smashstack.html> (1996).



1990s: Security Issues

- **Crypto wars:** is wide-spread use of strong cryptography a good idea?
 - Internet security treated as a communications security problem.
- **Buffer overrun attacks:**
 - Aleph One: Smashing the Stack for Fun and Profit
 - Internet security is mainly an end systems issue!
- **Java security model:** the sandbox.
- **Trusted Computing;** Digital Rights Management (DRM)
- **Status today:** mature security protocols (IPsec, SSL/TLS), better software security.

2000s: Web – e-Commerce

- **Technology:** Web services, WLAN, Public Key Infrastructure (PKI)??
- Web 2.0: dynamic content
- **B2C applications:** Amazon, eBay, airlines, on-line shops, Google, ...
- Criminal activity replaces “hackers”.
- Legislation to encourage use of electronic signatures.
- PKIs have not taken off; e-commerce has essentially evolved without them.

2000s: Security Issues

- SSL/TLS for secure sessions.
- **Software security**: the problems are shifting from the operating systems to the applications (SQL injection, cross-site scripting).
- Security controls moving to application layer: Web pages start to perform security checks.
- Access control for virtual organizations: e.g. federated identity management.
- Security of end systems managed by the user.

Disruptive Technologies

- Cheap and simple technologies that do not meet the requirements of sophisticated users but are adopted by a wider public.
- When the new technology acquires advanced features, it takes over the entire market.
- What happened to workstations?
- **Problem for security**: security features may not be required by the applications the new technology is initially used for; when it turns into a platform for sensitive applications, it becomes difficult to re-integrate security.

Summary

- Innovations (mouse, GUI, PC, WWW, worms, viruses) find their way out of research labs into the mass market.
- Innovations are not always used as expected: email on the Internet, SMS in GSM.
- Also the users are *inventive*!
- When new technology are used in innovative ways, old security paradigms may no longer apply and the well engineered 'old' security solutions become irrelevant.
- We can start all over again ...