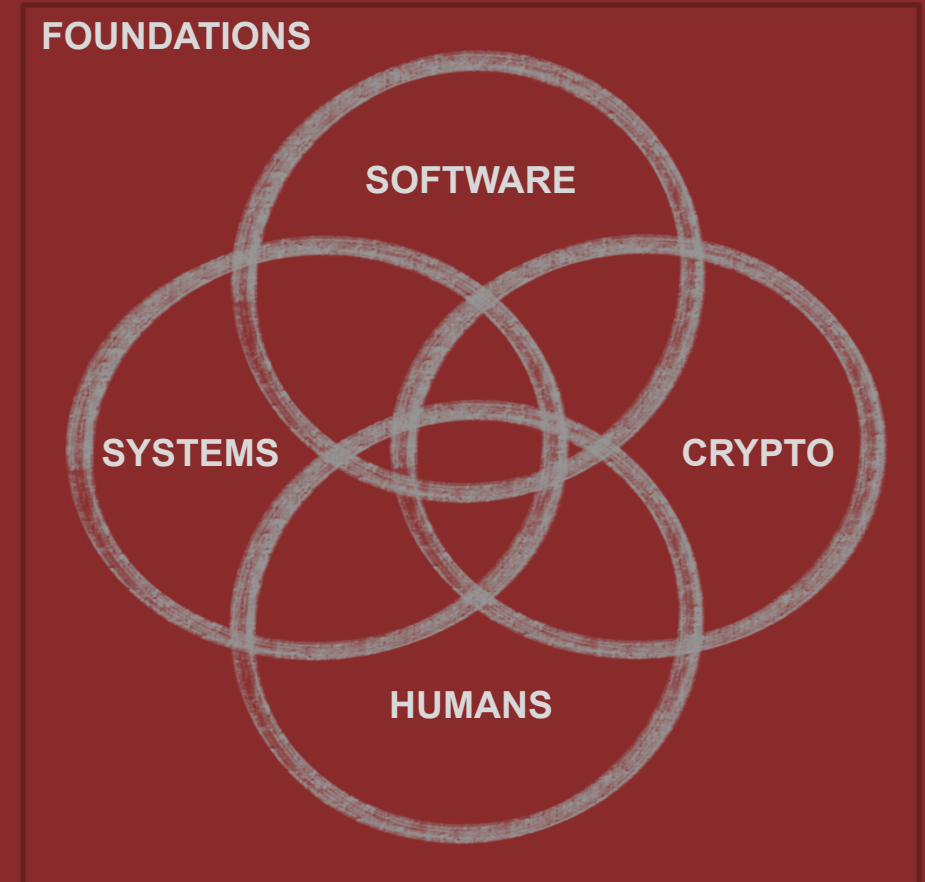
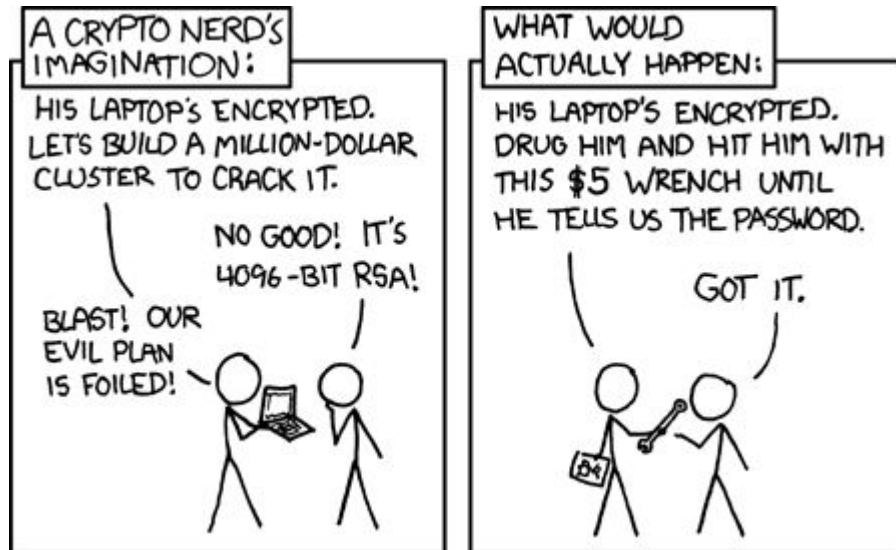


Ασφάλεια Υπολογιστικών Συστημάτων

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Θανάσης Αυγερινός



Huge thank you to [David Brumley](#) from Carnegie Mellon University for a lot of the content contained within these slides

Cloudflare Goes Down Again as a New Large-Scale Internet Outage Disrupts Major Online Services

Rosalia Mazza Friday, December 5th 2025 10:27 AM



React2Shell (CVE-2025-55182)

What?

A 10.0 critical severity vulnerability affecting server-side use of React.js, tracked as [CVE-2025-55182](#) in React.js and [CVE-2025-66478](#) specifically for the Next.js framework.

This vulnerability was responsibly disclosed by myself, [Lachlan Davidson](#) on 29 November 2025 PT to the Meta team. Initial disclosure and patch release was performed by React and Vercel on 3 December 2025 PT.

Update: Proof-of-Concepts

A real public PoC began circulating after around 30 hours from initial disclosure, I've now shared my [PoCs](#) several hours later. Full writeup in due course :)

Update: A note on scanners (4 December 20:55 UTC)

We've seen some great scanners from the likes of Assetnote, which are very effective at detecting unpatched Next.js instances that use Server Components.

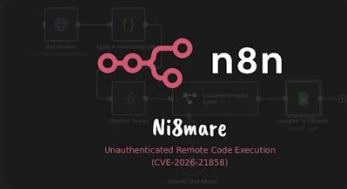
[Home](#) [For you](#) [Following](#) [News Showcase](#) [U.S.](#) [World](#) [Local](#) [Business](#) [Technology](#)

① How these stories are ranked

 The Hacker News

Critical n8n Vulnerability (CVSS 10.0) Allows Unauthenticated Attackers to Take Full Control

6 hours ago • By Ravi Lakshmanan



 Forbes

Google Chrome 143 Security Bypass — 3 Billion Users At Risk

8 hours ago • By Davey Winder



 CybersecurityNews

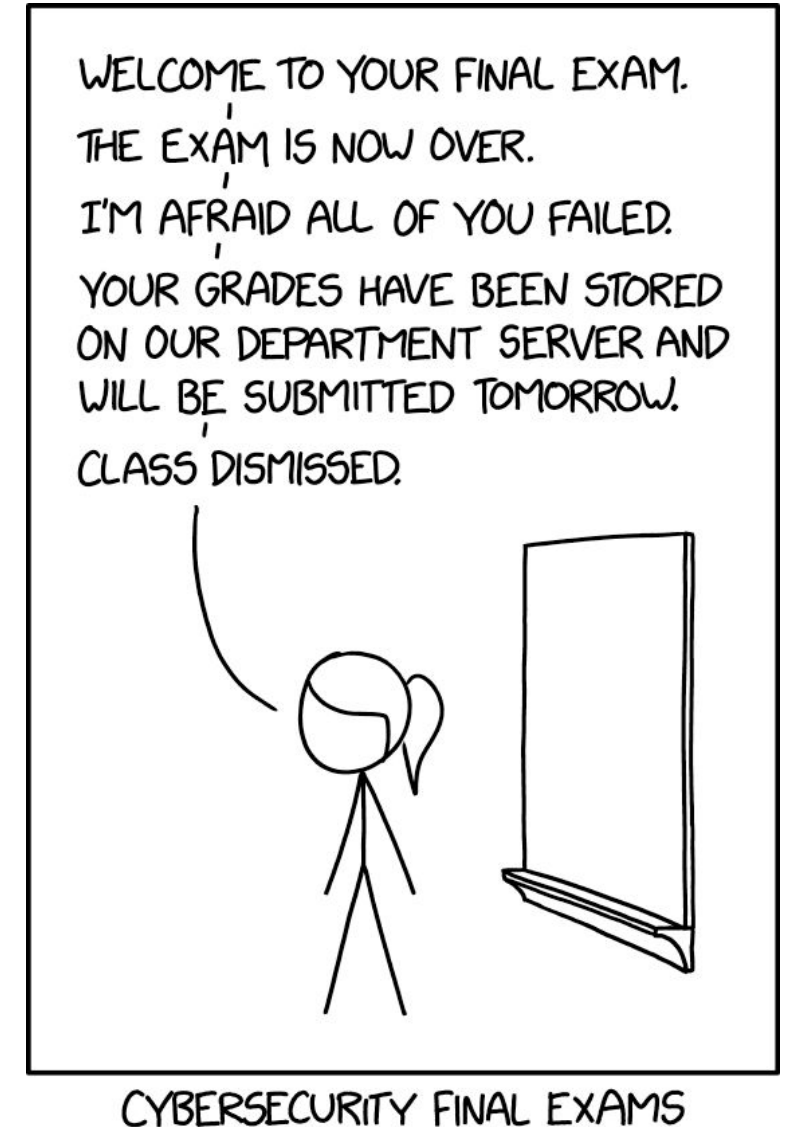
New macOS TCC Bypass Vulnerability Allow Attackers to Access Sensitive User Data

Yesterday



Στόχοι της Εισαγωγής στην Ασφάλεια (8ο εξάμηνο)

- Γενικό υπόβαθρο σε ασφάλεια (όροι, νοοτροπία)
- Βασικές τεχνικές άμυνας και επίθεσης σε υπολογιστικά συστήματα (λογισμικό, δίκτυα, κρυπτογραφία και forensics)
- Επαφή με state-of-the-art τεχνικές που έχουμε σήμερα



Σήμερα

- Τι είναι η "ασφάλεια";
- Βασική Ορολογία και Κατηγοριοποίηση
 - Adversary
 - Defense mechanisms
- Γράφοντας ασφαλή προγράμματα στην C
 - Buffer overflows



About Security

What is Computer Security?

Computer security, cybersecurity, digital security or information technology security (IT security) is the **protection of computer systems from attacks by malicious actors** that may result in unauthorized information disclosure, theft of, or damage to hardware, software, or data, as well as from the disruption or misdirection of the services they provide.

What do we mean by Hacker?

A hacker is a person skilled in information technology who achieves goals by non-standard means.

- Ethical ([white-hat](#)) hacking
- Unethical ([black-hat](#)) hacking



Stereotypically they have "[the knack](#)".



Security/Hacking Mindset

- Looking for weaknesses and strengths in any system
 - similar to puzzles
- Continuously asking "what can go wrong? why it works?"
- Proper engineering starts with security first



Security/Hacking Mindset

- Looking for weaknesses and strengths in any system
 - similar to puzzles
- Continuously asking "what can go wrong? why it works?"
- Proper engineering starts with security first

Πρωτοετής:

Καλησπέρα, χθες σας είπα πως το πρόγραμμα μου για την Άσκηση X έκανε compile στα Linux server αλλά το uoa bot έλεγε πως δεν έκανε.

Το πρόβλημα τελικά ήταν πως έκανα inline μια αναδρομική συνάρτηση το οποίο προκαλούσε link error χωρίς την παράμετρο -O3. Όπως το καταλαβαίνω χωρίς το -O3 ο linker δεν κάνει τόσα optimization και έτσι ψάχνει για τον ορίσμο όλων των συναρτήσεων (όπου το inline το αφαιρεί).

Security is ...

New and fast-moving. There's a reason there is no established textbook.

Critical and everywhere. Can you think of any applications?

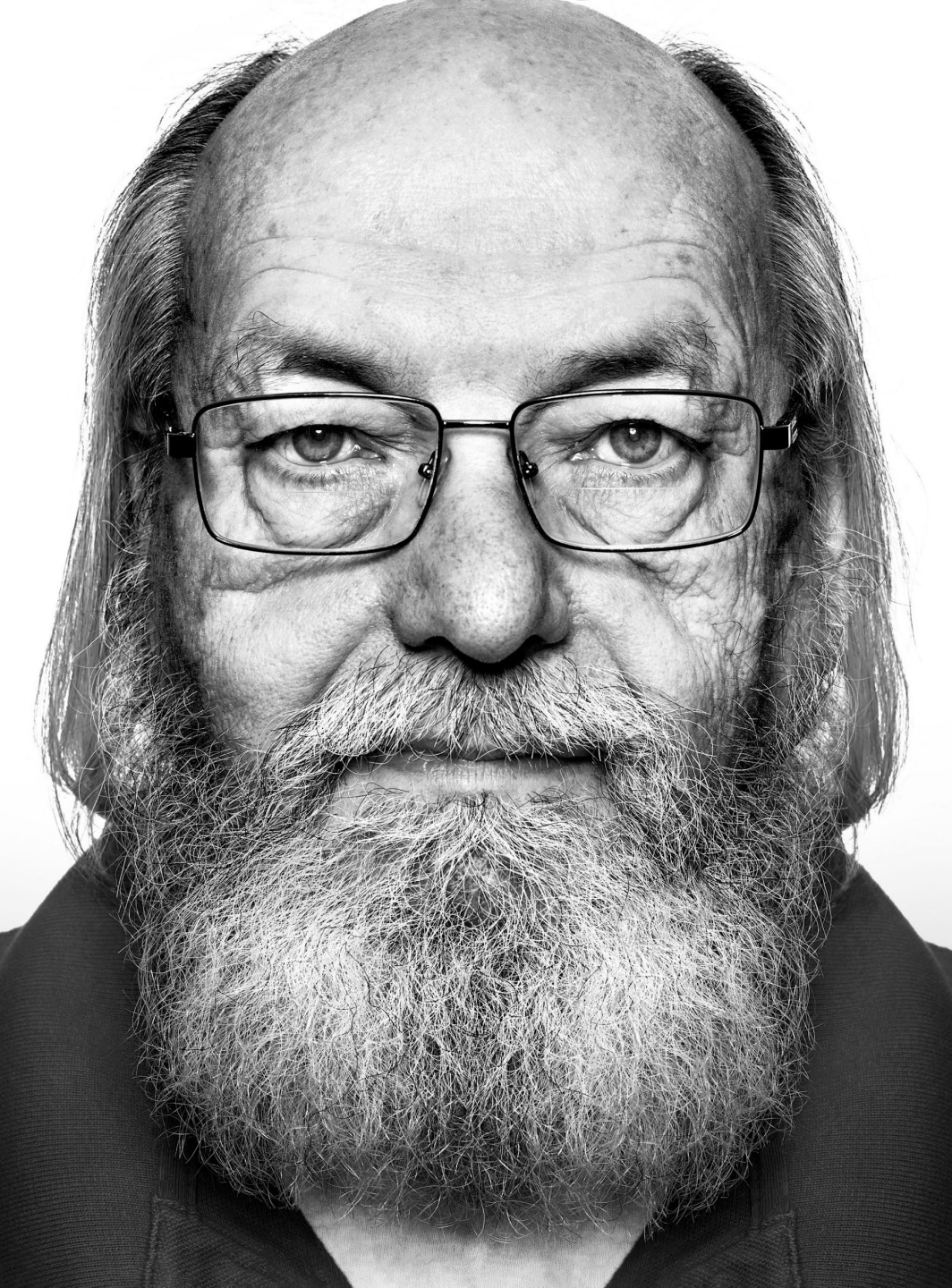
An arms race. Attack vs defense, red team vs blue team.

Hard. Proving something secure is not easy. Why?

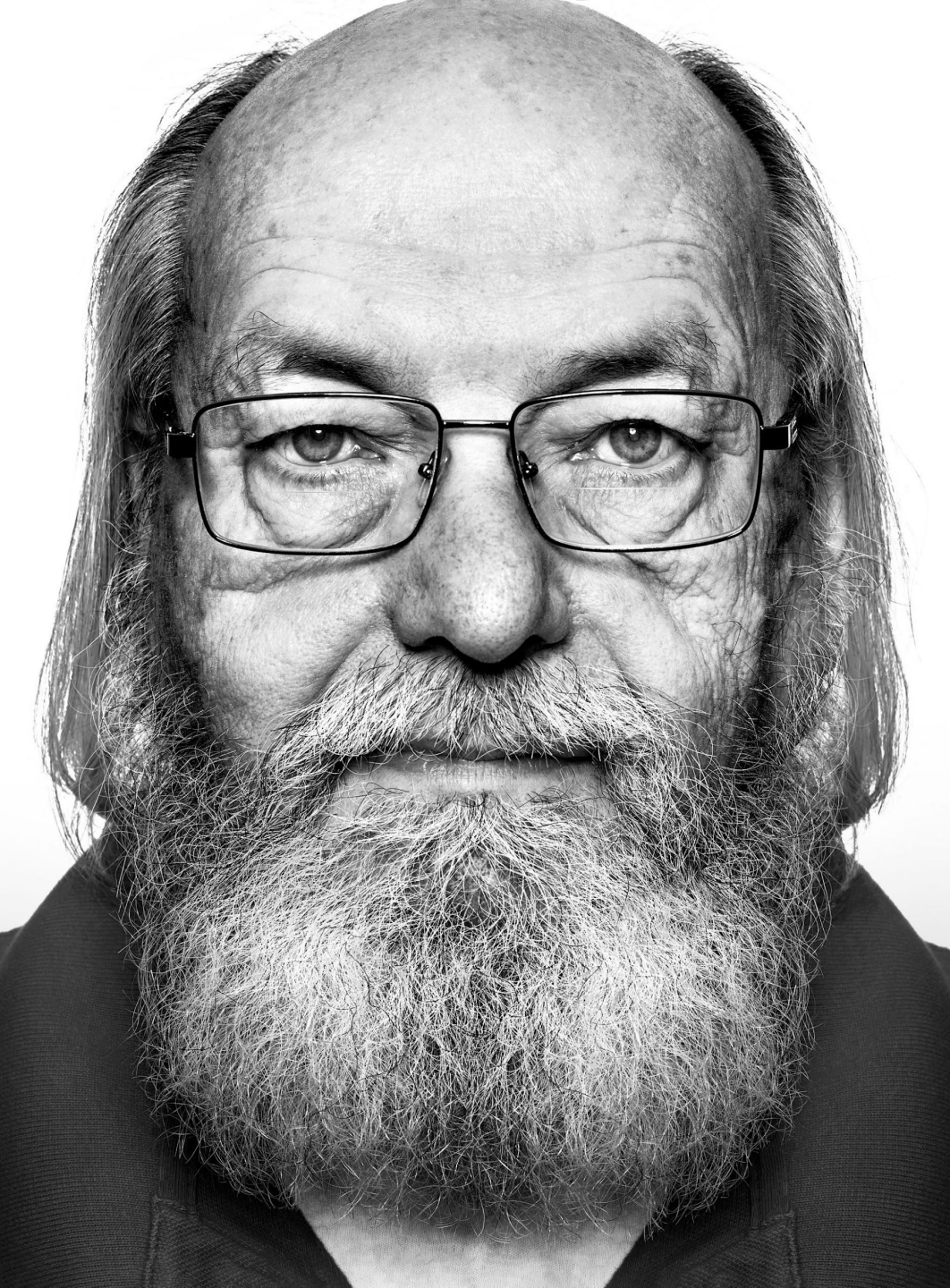
An active field of research. Looking for better solutions. ([AIxCC](#))



**Security is About
Trust**



DO YOU TRUST HIM?



KEN THOMPSON

Co-creator of UNIX and C
1983 Turing Award

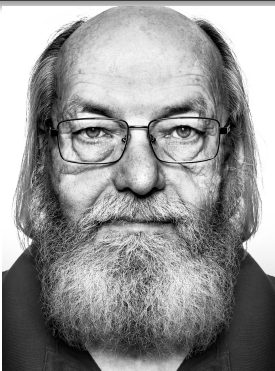
[Reflections on Trusting Trust](#)

login.c

```
static void load_credentials(struct login_context *cxt) {  
    char str[32] = { 0 };  
    char *env;  
    struct path_cxt *pc;  
  
    env = safe_getenv("CREDENTIALS_DIRECTORY");  
    if (!env)  
        return;  
  
    pc = ul_new_path("%s", env);  
    if (!pc) {  
        syslog(LOG_WARNING, _("failed to initialize path context"));  
        return;  
    }  
  
    if (ul_path_read_buffer(pc, str, sizeof(str), "login.noauth") > 0  
        && *str && strcmp(str, "yes") == 0)  
        cxt->noauth = 1;  
  
    ul_unref_path(pc);  
}
```

Compiler

\$./login



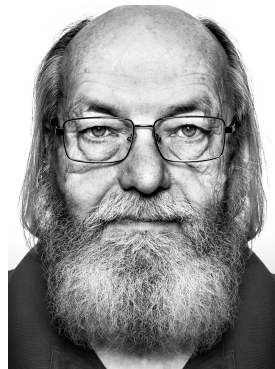
```
if(strcmp(pass, backdoor))  
    system('/bin/bash');
```

login.c

```
static void load_credentials(struct login_context *cxt) {  
    char str[32] = { 0 };  
    char *env;  
    struct path_cxt *pc;  
  
    env = safe_getenv("CREDENTIALS_DIRECTORY");  
    if (!env)  
        return;  
  
    pc = ul_new_path("%s", env);  
    if (!pc) {  
        syslog(LOG_WARNING, _("failed to initialize path context"));  
        return;  
    }  
  
    if (ul_path_read_buffer(pc, str, sizeof(str), "login.noauth") > 0  
        && *str && strcmp(str, "yes") == 0)  
        cxt->noauth = 1;  
  
    ul_unref_path(pc);  
}
```

Compiler

\$./login



```
if(program == "login")  
    add-login-backdoor();  
if(program == "compiler")  
    add-compiler-backdoor();
```



Guarantee that array and vector indices are within bounds

Mask signals handled by noninterruptible signal handlers

Sanitize the environment when invoking external programs

WOULD YOU TRUST MR ROGERS TO WRITE CODE?

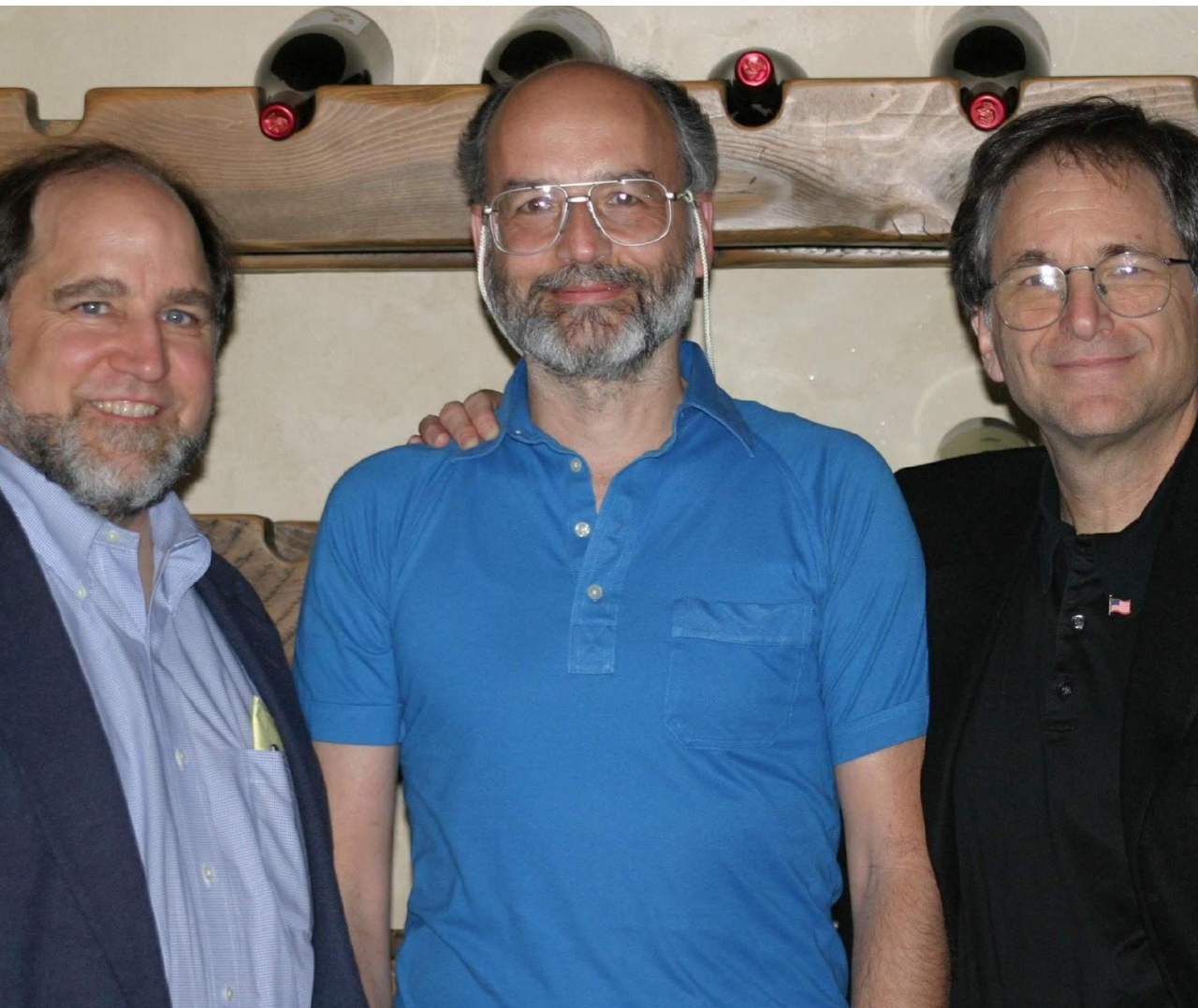
Exclude user input from format strings

Ensure that unsigned integer operations do not wrap

Do not call system()

Do not subtract or compare pointers that do not need to

Use the readlink() function properly



DO YOU TRUST SECURE CRYPTO ALGORITHMS?

$\forall m_0, m_1 \in M.$ where $|m_0| = |m_1|$

$\forall c \in C.$

$\Pr [E(k, m_0) = c] = \Pr [E(k, m_1) = c]$

Ron Rivest, Adi Shamir, Len Adleman

SERIOUSLY



**DO THESE PANTS
MAKE ME LOOK FAT?**

Implementations may still leak

```
message_t decrypt(ciphertext_t c, key_t k){  
  
    if(k == 1)  
        return decrypt(m) // Takes time 1  
    if(k == 2)  
        return decrypt(m) // Takes time 2  
    if(k == 3)  
        return decrypt(m) // Takes time 3  
  
}
```



NETWORKED SYSTEMS

- Software has same security concerns
- Protocols play a larger part
- New, interesting security properties desired

Sorry, but your password must contain an uppercase letter, a number, a hieroglyph, a feather from a hawk and the blood of a unicorn.

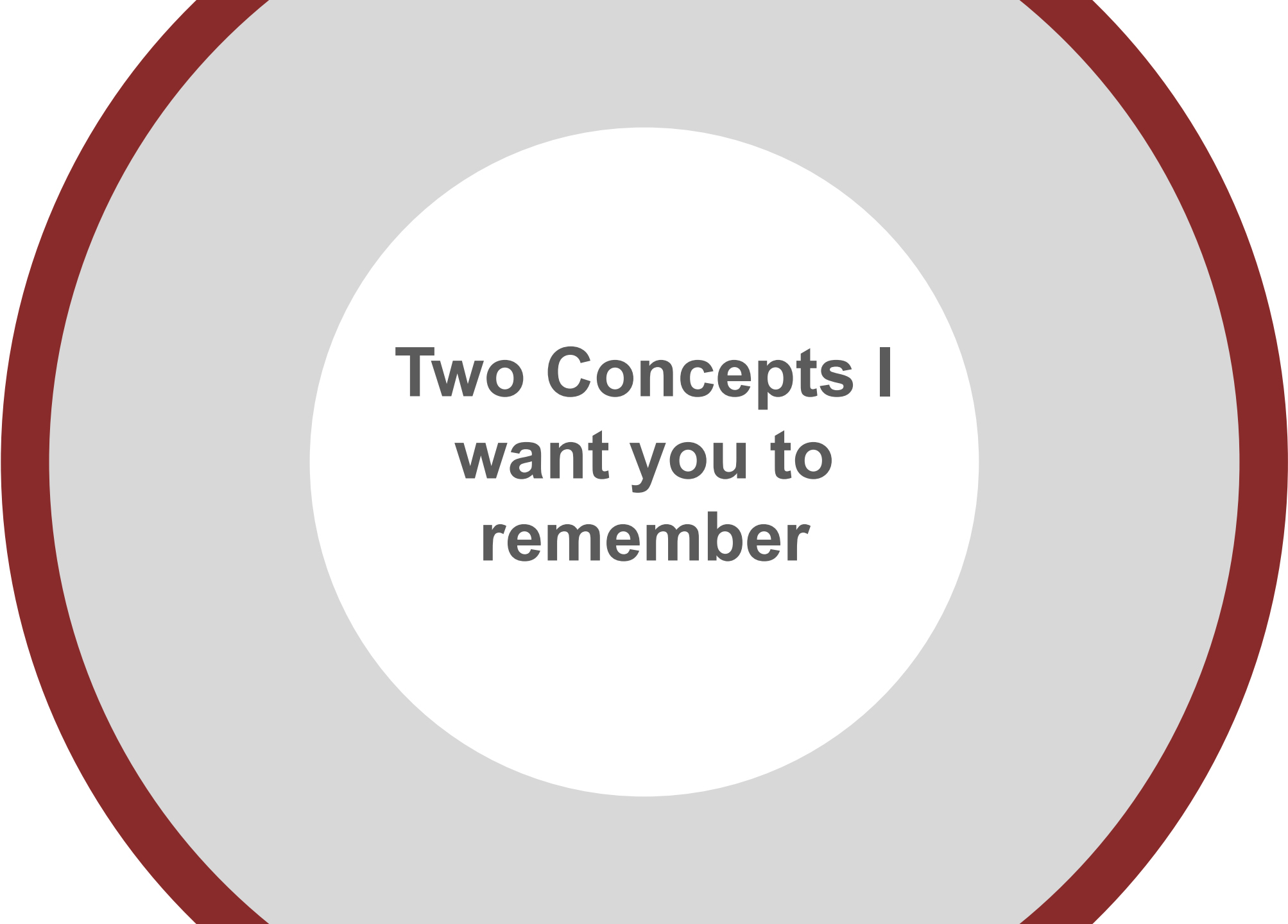
som^{ee}cards
user card



HUMAN INTERACTION

- Unusable systems become insecure
- Social engineering (prompt injections for humans!)

Username : admin
Password : admin



**Two Concepts I
want you to
remember**

Defining The Adversary (1/2)

- Adversary = < Goals, Capabilities >
- Goal: What constitutes success?
 - May involve subgoals
 - Example goal: Gain control of X's data
 - Sub-goal: Reconnaissance: search online for info about X
 - Sub-goal: Access: Guess X's ssh password on Linux lab
 - » Sub-goal: Lateral movement: Use ssh account to move to other services / linked accounts
- Capabilities: What resources can the adversary use?
 - 1 computer or millions?
 - Physical or remote access?
 - Access to source code?

Why don't we include
adversary's strategy?

Defining The Adversary (1/2)

- Adversary = < Goals, Capabilities >



Why don't we include
adversary's strategy?

Security Mechanism Classification for a property (2/2)

- 1. Prevention.** Prevent issues from happening. Any precautionary measures.
- 2. Detection.** Assuming an incident took place, detect them as early and as accurately as possible.
- 3. Resilience.** Assuming one or multiple incidents took place, ensure the overall system security degrades gracefully and does not collapse.
- 4. Deterrence.** Measures to ensure penalties for actors responsible for security incidents. Policy-based.



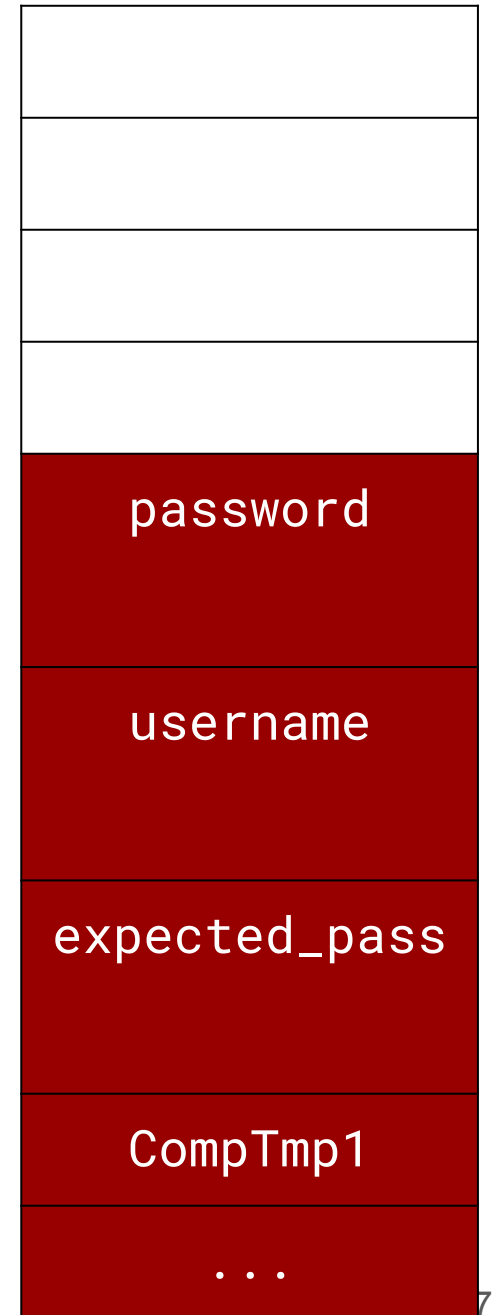
**Secure Coding is
Hard - One
Example: Buffer
Overflows**

```
int main() {  
    char expected_password[8] = "sesame";  
    char username[8];  
    char password[8];  
    printf("Enter username: ");  
    scanf("%s", username);  
    printf("Hello %s, Enter password: ", username);  
    scanf("%s", password);  
    if (strcmp(password, expected_password) == 0) {  
        printf("Success! You are now the admin.\n");  
    } else {  
        printf("Failed, enter the right password.\n");  
    }  
    return 0;  
}
```

Τι κάνει αυτό το πρόγραμμα;

```
int main() {  
    char expected_password[8] = "sesame";  
    char username[8];  
    char password[8];  
    printf("Enter username: ");  
    scanf("%s", username);  
    printf("Hello %s, Enter password: ", username);  
    scanf("%s", password);  
    if (strcmp(password, expected_password) == 0) {  
        printf("Success! You are now the admin.\n");  
    } else {  
        printf("Failed, enter the right password.\n");  
    }  
    return 0;  
}
```

main
stack
frame



```
$ ./login  
Enter username: ethan  
Hello ethan, Enter password: notsure  
Failed, enter the right password.
```

```

int main() {

    char expected_password[8] = "sesame";

    char username[8];

    char password[8];

    printf("Enter username: ");

    scanf("%s", username);

    printf("Hello %s, Enter password: ", username);

    scanf("%s", password);

    if (strcmp(password, expected_password) == 0) {

        printf("Success! You are now the admin.\n");

    } else {

        printf("Failed, enter the right password.\n");

    }

    return 0;
}

```

```

$ ./login
Enter username: ethan
Hello ethan, Enter password: notsure
Failed, enter the right password.

```

password

'n' 'o' 't' 's'
'u' 'r' 'e' '\0'
'e' 't' 'h' 'a'
'n' '\0'
's' 'e' 's' 'a'
'm' 'e' '\0'
CompTmp1
...

username

expected_pass

Επομένως, αν δεν έχουμε το `expected_password`
δεν μπορούμε να κάνουμε login, σωστά;

Demo

Συμπέρασμα: αποφεύγουμε την χρήση
`scanf("%s", ...)` όποτε μπορούμε.

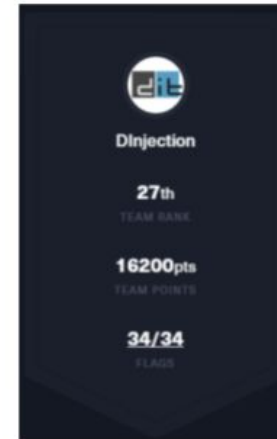
Πως αλλιώς; Πολλές δυνατότητες:

1. `getchar() + realloc`
2. `scanf("%ms", ...)` - non-standard :(
3. `getline`, κoκ
4. ...

Is that all?



UniCTF 2025 Results - Hack-The-Box (2025/12/19)



27	DInjection	GR	16200	34/34
28	Technical Cijg Napoca	RO	11250	33/34
29	Universidad del Gran Rosario	AR	11250	33/34
30	University of East London	GB	11250	33/34
31	Universitat Autònoma de Barcelona	ES	11250	33/34
32	Abu Dhabi University	AE	11250	33/34
33	GRU	EG	11250	33/34
34	UnEgint	GB	11250	33/34
35	NTUA	GR	11250	33/34

27η θέση στον κόσμο, 1η θέση στην Ελλάδα (μοναδική που έλυσε όλα τα προβλήματα), συμμετείχαν και πρωτοετείς!?

Lots of opportunities, just find a reason to get started!

Ευχαριστώ και καλή μέρα εύχομαι!

Let's start hacking!
(BUT **do not** hack professors :))