Security & Privacy

Stephen Farrell
stephen.farrell@cs.tcd.ie

Course materials:
https://down.dsg.cs.tcd.ie/cs7053/
https://github.com/sftcd/cs7053
Slideware + some papers
Introduction of some basic concepts and terminology in computer security
Next hour(s)...

- A bit more history...
- Application layer security
- Security evaluation
- Network security
- Identification & Authentication
Computer Security Goals

Traditional Security Goals: "CIA"

- Confidentiality
  - Keeping secrets
- Integrity
  - Preventing unauthorised modifications
  - Keeping data consistent
- Authentication/Assurance/Availability
  - The meaning of the “A” depends upon who you ask!
Services and mechanisms

• From the old “OSI” world, we've inherited two (sometimes confusing) concepts:
  – Security service: provides a security function to the system, based on the use of security mechanisms, e.g. confidentiality
  – Security mechanism is a technique, or protocol etc. which can be used to provide a service, e.g. encryption, access control
Services/Mechanisms

• Confidentiality
  • Access control
  • Cryptography
  • Secure communication channels

• Integrity
  • Checksums
  • Digital signatures
  • Secure communication channels

• Availability
  • Designing secure software and communication protocols
  • Load balancing
  • Anti-DDoS services
Structure of Military Security

- Information is classified according to national security
- The classification is clearly labeled on the binder
- All classified information is stored in a safe
- All users are “cleared” to see information up to a certain level
- Users have to prove their clearance to withdraw the binder from the safe
- Additional compartments enforce the need-to-know principle
Creating New Information

• New files are labelled with the classification of the most secret component
• Aggregation of unclassified information may generate a “top secret” file
• Sanitization downgrades the label of existing information
Old Problems Aggravated

• Aggregation
  – it is easier to aggregate information from a vast set of information

• Authentication
  – it is more difficult for a computer to identify a person (partly solved)

• Browsing
  – it is easier to read all files in a file system than all files in a safe

• Integrity
  – modification is harder to detect (mostly solved)

• Copying
  – digital copies are indistinguishable from the originals (really there are no originals at all!)

• Denial of Service
  – denial of service is a notorious characteristic of computer systems
Covert Channels

• Hidden means of communication, that allows information to be leaked to third parties

• Some types:
  – Timing Channels
    • observable differences in system utilization
  – Inference Channels
    • intersection of non-classified information
  – Fabrication Channels
    • aggregation of non-classified information
Security Perimeters

<table>
<thead>
<tr>
<th>System Boundary</th>
<th>subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Perimeter</td>
<td>user programs</td>
</tr>
<tr>
<td>Operating System</td>
<td>servers</td>
</tr>
<tr>
<td>Trusted Computing Base</td>
<td>services</td>
</tr>
<tr>
<td>Reference Monitor</td>
<td>objects</td>
</tr>
</tbody>
</table>

users, remote workstations, the Internet
Attack Surface

• Intuition: more possible ways in which attacks can be attempted => more likely an attack may succeed
  – Listener on a port, Callable API, Command line arguments, ...
• Think of the potential attack points as if they formed a geometric construct: the attack surface
  – Smaller attack surface == better! (probably)
  – Useful relative metric as you evolve a system
  – Argues (to an extent) against over-generic tools/re-use (e.g. CMS for web sites perhaps)
• Dec 2018 survey paper (paywall):
  – https://doi.org/10.1016/j.infsof.2018.07.008
• OWASP guidance:
  – https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet
• Michael Howard concept from 2003:
  – https://docs.microsoft.com/en-us/previous-versions/ms972812(v=msdn.10)
Sidebar: Operating Systems

• Won't cover OSes or virtualisation much as issues are very system dependent
  – UNIXes, incl. GNU/Linux, OpenWRT, Apple iOS/OS X, Android, flavours of busybox, Windows, Cisco IOS, JunOS,...

• Basic idea is to isolate processes (users are represented as processes) and privileges

• Cross-boundary vulns have most impact

• Local privilege escalation attacks: meh

• Remote code execution (RCE) or break out of VM/browser sandbox: BIG deal!

• Recent work on side-channels (spectre/meltdown etc.) tells us speed and security trade-offs have been mismanaged in many environments for many years
Commercial Security

• Historically the military paid more for security, were more accepting of the inconvenience (which almost all security mechanisms introduce), and had a chain-of-command upon which to fall back.

• Co-operating commercial enterprises have none of these.
The 1988 Internet worm

- Propagated (mainly) via SMTP
  - using the DEBUG build of sendmail (then the prevalent mail server)
  - Also exploited fingerd problems
- It killed the Internet for a whole day!
- Spafford: “The Internet worm program: An analysis” well worth a read

Typical Enterprise Security Model (circa 1995)

• System and network administrators setup and manage users and applications

• DCE exemplifies this approach
  – http://www.opengroup.org/dce/
    • Interesting that URL still works
  – Kerberos (RFC4120) – still persists as part of MSFT windows and a few other systems
Problems with the 1990’s enterprise security model

• Generally assumed a homogeneous network and set of applications
  – Which was false
• Assumption that all users and applications are centrally managed
• Deployment showed up performance and usability issues
Security APIs

• Periodically someone tries to develop an API to “hide” security
• Has worked fine for cryptographic primitives
  – MS-CAPI, PKCS#11, JCE
  – Javascript WebCrypto API
• Not so successful for higher layer functions
  – GSS-API, SPKM
Sidebar: Crypto makes interop hard

• It is **much** easier to get a system to work in “insecure” mode, compared to “secure” mode
  – And **much, much** easier if secure mode involves cryptography

• This is a general problem which leads many people to turn off security

• Answer: develop with the crypto turned on

• Bear it in mind as you design things
Along came the web

• The number of connected hosts rose exponentially for a while
• Highlights security issues with:
  – Proxies (various bad things can happen at a proxy)
  – Tunnelling (“protocol-X” over HTTP)
  – Having a ubiquitous tunnel end-point on many machines
  – Browser security models
Security Evaluation and Network security

Customers need some confidence that the system or network that they are about to purchase is “secure”.
Security Evaluation Criteria

[TCSEC] Department of Defence Trusted Computer Systems Evaluation Criteria (Orange Book)

[ITSEC] IT Security (UK, NL, FR, DE)

[Others] CTCPEC (Canada) JCSEC (Japan)[CC]

[CC] Common Criteria (ISO IS 15408)
Security Evaluation

• How do we compare the security of different computer systems?
  – Different authentication mechanisms
    • Kerberos, passwords, smart cards, ...
  – Different access control mechanisms
    • Access Control Lists, Role-Based Access Control, ...
  – Different cryptographic algorithms
    • DES, AES, RSA, ECC, D-H, Curve25519...
"Orange Book" Classification

<table>
<thead>
<tr>
<th>Class</th>
<th>Title</th>
<th>Key Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Verified Design</td>
<td>Formal top-level specification and verification, formal covert channel analysis, informal code correspondence demonstration</td>
</tr>
<tr>
<td>B3</td>
<td>Security Domains</td>
<td>Reference monitor (security kernel), “highly resistant to penetration”</td>
</tr>
<tr>
<td>B2</td>
<td>Structured Protection</td>
<td>Formal model, covert channels constrained, security oriented architecture, “relatively resistant to penetration”</td>
</tr>
<tr>
<td>B1</td>
<td>Labeled Security Protection</td>
<td>Mandatory access controls, security labeling, removal of security related flaws</td>
</tr>
<tr>
<td>C2</td>
<td>Controlled Access Protection</td>
<td>Individual accountability (authentication) and extensive auditing</td>
</tr>
<tr>
<td>C1</td>
<td>Discretionary Security Protection</td>
<td>Discretionary access controls, protection against accidents among cooperating users</td>
</tr>
<tr>
<td>D</td>
<td>Minimal Protection</td>
<td>Not Classified</td>
</tr>
</tbody>
</table>
ITSEC

• Separation of functionality and assurance
• Functionality-classes (F-C1 – F-B3) corresponds to the Orange Book
• Assurance-classes (E0 – E6)
  – E0: lowest, E6 highest
  – E4: commonest (maybe)
• Target of Evaluation
  – Set of evaluated components: security policy, security related functions, definition of the required security mechanisms for the target level of evaluation
Common Criteria

• Combines previous criteria
  – Separation of functionality and assurance
  – Target of Evaluation (TOE)
  – Security Target (ST) = desired level of evaluation

• Protection Profiles (PP)
  – Security Requirements
  – Security objectives
  – Independent of Implementation
Other aspects of assurance

• Installation and use must be considered
  – What are the defined security policies?
  – How is the software installed?
  – How are the machines administered?
• Development tools & environments
  – SNMP exploit due to buggy 3rd party ASN.1 handling library
Assurance != "Works-as-advertised"

• Assurance only shows that the product/system matches a specification
  – NOT that it does what the customer wants!
• The product owner pays the evaluator (lots!!)
  – so although the evaluator is Government licensed, they have reasons to be nice to the developer.
• Caveat emptor rules ok.
Network security

• Different from application layer security
  – Usually no real interest in APIs here
  – Normally trying to secure either a (sub)network, or network node from network based attacks
  – Frequently network nodes don't use standard operating systems
    • E.g. Cisco routers, 802.11 access points
Network Security

• First, read about networks, you need to have some level of understanding of IP (both), DNS, TCP, BGP, HTTP and how e.g. Javascript/PHP works on the web
  – Go find your own URLs:-) Or a book.

• Then, general network security:
  – https://www.sans.org/ training stuff etc.
  – https://isc.sans.edu/ “Internet Weather”
  – https://first.org/library/ more reading
  – https://www.owasp.org/ web security
Learn about your network(s)...

- Browser developer UI to see HTTP/JS
  - network/JS/console: “shift-ctrl-I” or similar
- Wireshark/tshark/tcpdump to see packets
  - Wireshark dissectors for many protocols, e.g. DNS, TLS
- Protocol capture (PCAP) files
  - Careful of privacy – PCAPs can be revealing and are hard to anonymise
  - Please tell me if you know of good up-to-date PCAP anonymisation tools!
- If this is new to you - **do play with these** – you’ll learn stuff!
Network security view

- Original Internet “architecture” assumed end-to-end connectivity
- Hence end-to-end security was the main consideration for those developing the Internet
  - But they were very slow developing IPsec (about 10 years!)
  - And meanwhile NAT and firewalls arrived
The End-to-End Argument


https://dl.acm.org/citation.cfm?id=357402

READ THAT!!!

And then RFCs 1958 & 3439
And RFC 7258 (why not:-)

- caution, author present
E2E Argument has current consequences...

HTTP/2 and QUIC are protocols that attempt to have e2e encryption as a mandatory mechanism (ish) not in order to achieve a confidentiality service but primarily to mitigate ossification.

Some, but not all, of those who know what’s going on are happy with that.

FWIW, (which is not much) I am happy with that.
Network Address Translation (NAT)

• NAT is today mainly used to hide local addresses from the Internet for (mostly) provisioning reasons
  – NAT means that the higher layer end-points “see” different addresses for one another
  – Breaks many end-to-end assumptions

• “Carrier Grade” NAT (CGN) is coming to an ISP near you soon (or has already)
CGN & Security

Many more endpoints behind each public IPv4 address; put a NAT box in the ISP network

NAT444

There are other, maybe better, IPv6 transition mechanisms
Some think this is an IPv6 avoidance mechanism

Good way to freak out “legal intercept” (LI) fans

LI+CGN: Good way to freak out ISPs who have to log

Good way to freak out anyone who wants to offer a public service

PITA for DNSSEC

Bad way to get to an IPv6 world?

But may be needed for some time
NAT problems with SIP

UserA      NAT                          NS          UserB
10.1.1.221  192.168.221.1  63.88.221.88  192168.1.10

|----F1 INVITE---->|--F2 INVITE->|--F5 INVITE->|
|<---F4 100--------|<-F3 100------|<---F6 180----|
|<---F8 180--------|<-F7 180------|<---F9 200 OK--|
|<---F11 200 OK----|<-F10 200 OK--|<---F14 ACK---|
|----F12 ACK-------|---F13 ACK----|--F14 ACK--->
|<---F15 2WAY RTP-->|----------F16 2WAY RTP------>
|----F17 BYE-------|--F18 BYE---->|<---F19 BYE---->
|<---F22 200 OK----|<-F21 200 OK--|<-F20 200 OK-|
Firewalls

• Until the mid-90's most sites didn't bother filtering traffic

• It became clear that exposing your internal network topology could cause problems
  – E.g. If I know that router brand “X” has an open default configuration and I can see that you've got one of those, then my attack point is clear
Filtering routers

• Initially people put in filtering rules in border routers
  – E.g. “no packets with a destination on my inner network are allowed out”

• But, IP spoofing attacks meant that this wasn't sufficient
  – So firewall products developed
  – Note: has been biggest security products market, I've not checked recently
Denial-of-Service (DoS)

• Consume resources to make a service unavailable
  – Has been known as a vulnerability for many years
  – Began to be exploited in last decade on major Internet sites
    • DDoS attacks against the DNS roots
• DDoS = Distributed Denial of Service
TCP SYN flooding attack

- SYN packets with spoofed IP addresses cause the server to maintain state
- Flooding the server that way is your basic DoS attack
  - CERT Advisory CA-96.21

Network access protocols

• Issue is how to decide when to allow a node to join/use the network
  – IEEE 802.1x networks (eduroam)
  – Corporate networks
  – Mobile operator networks
• RADIUS protocol (or Diameter)
  – RFC2865 (or RFC6733)
Search Engines

- ...to find vulnerable systems
- Shodan HQ https://www.shodan.io/
  - Est. 2013 - search for “camera”
- Censys https://censys.io/
  - Mainly IPv4 based surveying (zmap)
- Ichidan ichidanv34wrx7m7.onion
  - Didn’t work for me
- There are more I guess...
Tor

- Onion routing
  https://www.torproject.org/
- Tor Browser (TBB)
  https://www.torproject.org/projects/tor-browser.html.en
- Tor Hidden services: “<hash>.onion”
  DNS-like names
- Search engine: https://ahmia.fi/
- Be a bit careful though!
My browser setup

- TBB: If searching for something sensitive (e.g. medical info)
- Search default: DuckDuckGo (Where “!g” works too:-)
- Default browser: FF + NoScript/Ghostery & disallowing cookies, with some white-listed sites
- Some sites don’t work with the above; mostly: screw ‘em
- If-need-be: chromium/incognito with a chmod of $HOME/.config/chromium to make almost all read-only (but not sure of effectiveness)
- If-all-else-fails: Brave or Vivaldi
- On phone: sailfish browser with no JS/no cookies and 2ndary open-kimono browsers if-need-be (Webcat/Web pirate)
- Recommend you figure out some browser-hygiene you prefer and implement that – be willing to help others do the same!
Identification and Authentication

Establishing identity and verifying credentials
Identification

• Establish the identity of principals by means of:
  – something known (password, PIN)
  – something possessed (smart card, Java ring)
  – something personal (fingerprint, retina scan)
  – something to do (signature)

• Combinations of above (smart card + PIN)
Passwords

• Most systems rely exclusively on passwords
  login: \textit{username}
  password: ******

• Password scheme problems:
  – users choose “bad” passwords (\textit{password, secret})
  – compromised via sniffing

• Good passwords won't be in a dictionary

• There are better password protocols
  – but first we need some more crypto
<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
<th>Known</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logins</td>
<td>10</td>
<td>6</td>
<td>Laptops, host systems (incl. root accounts)</td>
</tr>
<tr>
<td>Devices</td>
<td>5</td>
<td>0</td>
<td>DSL router, home print/file servers, sensor nodes</td>
</tr>
<tr>
<td>Network access</td>
<td>4</td>
<td>0</td>
<td>Work n/w, WLAN, ISP, etc</td>
</tr>
<tr>
<td>Protocol</td>
<td>14</td>
<td>1</td>
<td>Outbound HTTP proxy, IMAP, Jabber, skype, etc</td>
</tr>
<tr>
<td>Service</td>
<td>21</td>
<td>4</td>
<td>Mainly web sites with password stored outside browser</td>
</tr>
</tbody>
</table>
Kerberos Authentication

• Project Athena at MIT (mid to late 1980s)
• Hundreds of diskless workstations
  – open terminal access, no physical security
  – insecure network
• Few servers (files, print, mail, …)
  – physically secure
Simple Authentication

• A password per service is infeasible
• New authentication service (AS) introduced
• Both users and services have passwords
• AS identifies user by password
• AS/TGS issues a “ticket” to the user
  – ticket includes identity encrypted with the service’s password
  – if the ticket decrypts properly, access to the service is granted
  – TGS = ticket granting service
Insecure Workstations

• What happens to tickets after a user has logged out?
  – an opponent could log on to the workstation and use the tickets
  – could be explicitly destroyed when user logs out
  – sniffer could be used to capture tickets, hacker may then login to the same workstation and use the tickets (replay session)
Biometrics

• Many people (apparently) think that replacing username & password with a biometric is a great idea.

• It might be:
  – If secure biometrics exist
  – If the re-use implications are acceptable
  – If the context supports the full life-cycle
Biometric methods

• Fingerprint
• Retina scan
• Face recognition
• Gait (walking)
• Toe-smell
Fingerprint

• Prof. Tsutomu Matsumota's well publicised 2002 attack against most common fingerprint recognition engines
  – https://cryptome.org/gummy.htm
  – Information here is directly from his paper: “Impact of Artificial "Gummy" Fingers on Fingerprint Systems”
Fingerprint scanner

Figure 2.1 Typical structure of a fingerprint system
Make a gummy finger

Figure 4.5 Photographs of the outside appearance of the mold and a gummy finger. The gummy finger was produced from a residual fingerprint on a glass plate, enhancing it with a cyanoacrylate adhesive.

Figure 4.6 The Fingerprint image of the gummy finger, which was displayed by the system with Device H (equipped with a capacitive sensor).

Figure 4.7 Average number of acceptance for each device, in terms of gummy fingers which were cloned from residual fingerprints. Here, the subject is one person.
Results

• 11 of 11 scanners tested were broken
  – The list was extended a bit more since
• That's enough of a result!
But there’s more! ...

CCC 2017: 55 minute video of breaking biometrics
https://www.youtube.com/watch?
annotation_id=annotation_2684251971&feature=iv&src_vid=pIY6k4gvQsY&v=VVxL9ymiyAU

When are biometrics ok?

• What applications?
  – What type(s) of biometric?
  – What benefits?
  – What costs?
    • Financial and other (e.g. Privacy)
Privacy

- Rad RFC6973 - In addition to “normal” security threats we need to care about
  - Correlation
  - Identification
  - Secondary use
  - Disclosure
  - Exclusion
  - Re-identification
- Actually, it's not entirely clear to me that the risk analysis methodology we follow for security works well for privacy
  - There's scope for research there
Privacy Puzzle

- Emails contain a Received header field which can contain the mail user agent IP address
  - What consequences?
  - Overall good or bad from a privacy perspective?
What is your #1 mitigation for all problems?
What is your #1 mitigation for all problems?

Yes, Backup.

Do that. Early and often.
Summary

• Services vs. Mechanisms, CIA
• ~30 year history
• Common Criteria
• N/W security a bit different from system security
• Firewalls etc.
• Passwords (yuk)
• Biometrics (also yuk)
• Privacy (yay!)
• Do backups