CNS Lecture 12

Network defenses
- firewalls
- intrusion detection systems (IDS)
- Forensics

Where to encrypt?

- **Link layer**
  - encrypting modem, NIC (wireless)
  - transparent, fast
  - suitable for private net
  - protects only one link (pt-to-pt)
  - info may be exposed in OS

- **Network/Transport layer**
  - swIPe, IPv6 (IPsec)
  - transparent
  - selectable (policy)
  - appl./host keying
  - works over public net
  - virtual private network (VPN)

- **System layer**: encrypting file systems (EFS/CFS)

- **Application layer**
  - end-to-end over public net
  - custom applications (PGP, ssh, ssl)
  - intrusive, but flexible
  - API for application development

Client layout with the BIO API (TCP wrapper)

```c
SSL_library_init();
SSL_load_error_strings();
ctx = SSL_CTX_new(SSLv23_method());
conn = BIO_new_connect("host:port");
BIO_do_connect(conn);
ssl = SSL_new(ctx);
SSL_set_bio(ssl, conn, conn);
SSL_connect(ssl);
```

SSL on the wire

<table>
<thead>
<tr>
<th>Content Type</th>
<th>SSL3_RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>change_cipher_spec</td>
<td>20, 0x14</td>
</tr>
<tr>
<td>alert</td>
<td>21, 15</td>
</tr>
<tr>
<td>handshake</td>
<td>22, 16</td>
</tr>
<tr>
<td>application_data</td>
<td>23, 17</td>
</tr>
</tbody>
</table>

Handshake:

- HELLO_REQUEST (0)
- CLIENT_HELLO (1)
- SERVER_HELLO (2)
- CERTIFICATE (11)
- SERVER_KEY_EXCHANGE (12)
- CERTIFICATE_REQUEST (13)
- CLIENT_HELLO DONE (14)
- CERTIFICATE_VERIFY (15)
- CLIENT_KEY_EXCHANGE (16)
- FINISHED (20)

You are here …

- Attacks & Defenses
  - Risk assessment
  - Viruses
  - Unix security
  - Authentication
  - Network security
  - Forensics

- Cryptography
  - Random numbers
  - Hash functions
  - Public key
  - Symmetric key

- Applied crypto
  - SSH
  - PGP
  - SSL
  - Kerberos
  - Crypto APIs
  - Secure Coding

- Secure Coding
IP vulnerabilities summary

- Denial of service
  - ICMP smurf, redirects, unreachable
  - SYN flooding
  - Frag, Teardrop
- Impersonation
  - Host rename (LAN)
  - DNS/ARP cache poisoning
  - Source routing
- Session capture
  - TCP seq number guessing
  - TCP hijacking
- Server attacks
  - Application flooding (FTP, Mail, Echo)
  - Buffer overflows
  - Software bugs

Net attacker MO

- Find active hosts (DNS, ICMP broadcasts)
- Scan ports (Nessus, nmap, idlescan, SATAN)
- Determine OS (rmap/qemu/selinux)
- OS's handle strange packets often in unique ways
- Try exploit (guest/stolen accounts/stack overflows)
- Exploit (root shell, shell service to inetd.conf, modify /etc/passwd)
- Install hacking tools (root kit)
- Clean up logs
- Install trojans/sniffer/bot
- Review sniffer logs, get accounts/passwords to other systems
- Market your botnet to the bad guys
- Tell the world

SANS top 10 ports

Network defenses

- Disable
- Configure properly
- Wrappers, topwrappers
  - Filters (allow, deny)
  - Audit, and alarm
- Filtering portmap
- Application filtering (securelib)
- Patches
- Scanners (Nessus, SATAN, ISS)
- Firewalls
- Intrusion detection & response
- Encryption, IPsec, virtual private networks (VPNs)

Defense in depth

- On a hill
- Most
- Outer wall
- Archer towers
- Inner wall

Assess your attack surface

Scanners
- ISS, Nessus, nmap -- probe and report network hosts and services
- Point, click, scan a net
- Port probes (nmap)
- OS type probes (nmap)
- Portmap probes
- X and NFS attempts
- Sendmail checks
- NIS probes

Host network services "wrappers"

- Host-based (wrappers, personal firewalls)
- Router-based (filters)
- Firewalls
- Intrusion Detection Systems (IDS/IPS)
- Authentication/encryption (IPsec/VPNs)
firewalls

- NO connection -- best 😊
- Toolkit, personal firewalls (Linux, PC)
- Filtering/screening routers
- dual-homed gateway ( bastion host )
- screened host gateway
- screened subnet ( NAT )
- commercial solutions ( enterprise firewalls )

Personal (host) firewalls

- Network access control lists
  - Which hosts/nets you permit/deny
  - Which services you permit/deny
- Make your host invisible to net (ping/port scans)
- PC/Windows — XP firewall (ICF), ZoneAlarm, NetIce
- Linux — iptables
- MAC — ipfw

Difficult to configure and EVERY host needs to do it.
If bad guy gets in to your host, he'll disable your host's firewall.

Windows XP firewall

- SP1, ICF ( properties of LAN connection )
- SP2, Security Center ( firewall, auto updates, virus)
  - Blocks outside requests
  - Alerts if program attempts to use Internet
  - Add exceptions ( program or port )
  - Keep a log

zonealarm

Linux firewalls

- ipfwadm begat ipchains begat iptables
- accept/reject rules ( tables ) + logging
- Redhat selects security ( high, medium, none )
- provides Network Address Translating ( NAT ), masquerading
- IP forwarding ( private net 10.0.0.0, 172.16.0.0, 192.168.0.0 )

Home protection

- Personal PC firewalls ( ZoneAlarm, iptables )
- DSL/Cable
  - Inexpensive router, NAT, firewall
  - Home network with perimeter protection
- Wireless
  - Enable 128-bit WEP key
  - Accept only designated ether addresses ( MAC filter )
  - Disable SSID broadcast
  - Use ssh or VPN
- Review logs

- Commercial solutions ( enterprise firewalls )
Screening routers

- router's job is to forward packets (fast)
- add filters (ACL's) for each interface
- can block IP address spoofing of internal addresses
- should permit out only legit. local addresses
- may deny/restrict specific services (ports)
- weaknesses
  - complicated filter expressions
  - may fail to the open mode
  - limited logging
  - no authentication
  - DNS spoofing

Fort deny list:
portmap, tftp, smmp, syslog, telnet
Restrict http to designated servers

Screening routers -- rules

| access list: 102 specifies what addresses are allowed out |
| access-list 102 deny   ip 128.219.250.0 0.0.1.255 0.0.0.0 255.255.255.255 |
| access-list 102 deny   udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 162 |
| access-list 102 permit ip 128.219.0.0 0.0.255.255 0.0.0.0 255.255.255.255 |
| access-list 102 permit ip 134.167.0.0 0.0.255.255 0.0.0.0 255.255.255.255 |
| access-list 102 permit ip 192.12.68.0 0.0.0.255 0.0.0.0 255.255.255.255 |

| access list: 112 denies local addresses from the outside |
| access-list 112 deny ip 128.219.0.0   0.0.255.255 0.0.0.0 255.255.255.255 |
| access-list 112 deny ip 134.167.0.0   0.0.255.255 0.0.0.0 255.255.255.255 |
| access-list 112 deny ip 192.12.68.0   0.0.0.255   0.0.0.0 255.255.255.255 |
| access-list 112 deny ip 130.225.220.16 0.0.0.0 0.0.0.0 255.255.255.255 |
| access-list 112 deny   udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 161 |
| access-list 112 deny   udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 69 |

| special internal hosts |
| access-list 112 deny ip 0.0.0.0 255.255.255.255 128.219.250.0 0.0.1.255 |

other rules for what routes are advertised

Bastion host

dual-homed gateway
- host with two network interfaces
- IP forwarding disabled
- reachable from either side, but packets do not flow from one side to the other
- user must login to bastion host, then to other side
- supplement with application gateway software (mail, ssh)
- strong authentication (SecureID), logging
- limited services (restricted shell), wrappers
- custom mail programs
- hides enterprise network (private IP addresses)

Screened host/net

Screened host
- common implementation
- traffic cut from Internet, allowed only to bastion host, though can let internal hosts access some Internet services (ssl, ftp, www)
- bastion host acts as an application gateway

Screened subnet (DMZ)
- one or more bastion hosts on subnet
- internal net can be private (invisible), network
- address translation (NAT)
- place some servers on DMZ (www, anon ftp)
- place intrusion detectors, traps on DMZ
- place external DNS on DMZ

Application gateways

- run minimal services (trusted OS?)
- no compilers, linkers
- use wrappers
- no local logins
- custom servers (minimal pkt forwards, logging, ACLs)
- connections from outside
  - strong authentication (deny, secureID)
  - encrypted (ssh, ssh2)
- user then connects to internal host and logs in again
- 2-part mail forwarder/forwarder (IPS)
  - Remove evil attachments
  - Block spam

Proxy servers

internal hosts accessing the outside ("relay")

- need socke/Find local applications
  #define connect Sconnect
- proxy on bastion host: (tn-gw, rlogin-gw, ftp-gw, x-gw, http-gw)
- servers are simple packet forwarders with ACLs, telnet and
  on (Telnet)
- some services support proxies (telnet, gopher)
- socke Library for building your own local proxy
Enterprise firewalls

- router with an attitude
- establish a perimeter
- single point of protection (rather than host by host)
  - Controls inbound and outbound network flows (by host/service)
- principle of layering, reference monitor
  - always invoked
  - tamper resistant
  - small and simple (understandable)
- establish a policy
  - What’s not denied is allowed
  - What’s not allowed is denied -- best
  - Complications arise
  - security vs convenience

Firewall limitations

- What they don’t do:
  - don’t do UDP very well
  - don’t prevent session hijacking
  - don’t provide privacy
  - don’t protect against viruses
  - don’t protect against insider (need internal firewalls/enclaves)
  - don’t prevent backdoors (modems, VPNs/tunnels)
  - don’t log/alarm like an IDS
  - don’t improve throughput!

Selecting an enterprise firewall

- commercial, consultant, kit
- filters for both in and out
- filter granularity (stateful, for support)
- IP fragmentation management
- filter language and user interface
- proxy applications, client, extensible
- authentication mechanisms
- network address translation (NAT) and VPN
- integration with intrusion detection (IDS)
- IPv6
- logging and audit tools
- ease of installation
- performance
- cost

Intrusions

- Prevention
  - host based -- wrappers, patches, strong authentication
  - net based -- filtering routers, firewalls
- an intrusion may still occur

- Detection & Response
  - detect quickly, limit damage
  - detection information can be used to strengthen prevention
  - based on assumption that behavior of intruder differs from norm
    - e.g., credit card mis-usage

Intrusion Detection

- any set of actions that attempt to compromise integrity, privacy, or availability of a resource
- objective: detect and report/alarm

- techniques
  - real-time/batch
  - statistical profiles (user, system, network) -- deviations from the norm
  - attack signatures -- known to be bad
- host-based, net-based
- components
  - sensors
  - knowledge base (rules, norms)
  - audit logs for establishing norms and detecting ab-norms, accurate timestamps
  - decision modules
  - reporting/action modules
  - adaptive -- update knowledge base

Statistical profiles

- detecting anomalous behavior
  - establish a norm
  - train the detector
  - establish alarm thresholds
  - pattern matching, AI, neural networks
  - system logs, special logging
  - continuous learning
**Statistical profiles**

**User profiles**
- login times, durations
- login location
- resource usage (files, CPU time)
- process usage (mail, compile)

**Host profile**
- load average/time-of-day
- process count
- logins/hour
- process usage (mail, compile), time-of-day
- process profiles (system calls, file accesses)

**IDS norms**

**Attack signatures**

**Batch and real-time**
- Virus scanners (actively remove malware attachments)
- login/su failures
- application logs
- file/file-mode changes (tripwire)
- sendmail, fingerd, tftp, snmp attacks
- host impersonation
- Rule-based intrusion events:
  - Copying system files
  - Reading from devices (rather than files)
  - Concurrent logins
  - Reading other people's directories

**IDS host tools**

- logs (lastlogin, messages, syslog, wrappers, NT event logs)
- process accounting
- maybe ACLs and logs
- log watchers (watcher, swatch)
- recent attacks (CERT)
- cpm, tripwire
- Anti-virus software, mail checker
- periodic checksum of home page
  
  need them on every host!

**Network IDS**

**LAN or perimeter detectors**
- Sniffers
- Passive or optional router control
- Login banners to avoid privacy violations

**Use in conjunction with host detectors**
- collect packet traffic summaries
- Forensics: reconstruct attack from logs
- construct norms
- detect "ab-norms"

**Internet**

**Statistical profiles**

**Network profile**
- Typical load (pkts/sec/time-of-day)
- Most active hosts
- Typical services (volume, time-of-day)
- Host interconnect patterns (who, when, how long)
- Host service/port profiles
- Flow statistics (packet size, rate, interarrival, burst, duration)
  - Not looking at contents of packet (because of volume or encryption)
  - Detect type of flow (interactive, web, email, streaming, chat...
Flow characterization
Firewalls allow only certain services to flow. Hackers often get into an allowed service, e.g., use port 25 to carry e-mail traffic.

Two flows from a compromised host
Can you characterize a flow (mail, telnet/ssh, www, chat) based on flow stats (interarrival rate, packet size, volume, duration)?

Attack signatures

- batch and real-time
- denial of service
- host impersonation
- port scans
- "known" malicious ports
- source routing
- "big" packets (buffer overflows)
- content scanning — keystrokes, viruses

Firewalls allow only certain services to flow. Hackers often will trojan an allowed service, e.g., use port 80 to carry ssh traffic.

There is a lot of "door-knob rattling"—hundreds of probes per day...

IDS research

- distributed IDS
  - standard report formats
  - sensors (host, net, autonomous agents)
  - central manager
  - adaptive aggregation
  - country aggregation
- higher speed nets too fast — host IDS become more important, parallel, FPGA
- replacing OSI
- host filtering specific packets
- providing attack signatures and out-of-band info
- adaptive detectors
- active response (IPS) — adding filters and coding, removing attachments, counter-attack?
- shutting down — shut port, short (block hole, honey pot, router & pass)
- content monitoring and privacy — encryption (key, mac)
- detecting
  - detecting covert channels
  - denial of detection
- OS/CPU diversity to reduce vulnerability, immunology models
Backtracking spoofed IP address flows

- Spoofed IP source addresses used by Denial of Service and session hijacking
- Perimeter routers SHOULD block spoofed addresses
  - Don't allow internal addresses as source address from external interfaces
  - Only allow packets with valid source addresses out
- For an active attack that is using spoofed IP source addresses
  - Manually check each router along the flow, backtracking
  - Automated program to access routers and backtrack flow and setting filter to block
  - Hard: crosses administrative domains
- Other approaches, marking packets, new ICMP... open research

Pursuit
Response -- following up an intrusion

- policy/procedures (check lists)
  - Identify
  - Isolate
  - Evaluate
  - Remediate
  - Monitor
- Incident response team
  - Technical staff
    - (network/computer)
  - Security staff
  - Legal, public relations
- Contact
  - Attacking host/ISP
  - Management
  - FBI/CERT/CIAC
- Collect evidence (forensics)

Forensics -- cyber CSI

- Preserving evidence of the attack
- Determining how, who, when, where, why
- Damage assessment
  - What's been taken, modified, added, deleted
- Forensic tools (for Windows/Mac/Linux)
  - Sleuthkit: www.sleuthkit.org
  - TCT (The Coroner's Toolkit), FTK
  - Encase
  - Password crackers
  - Net and keyboard sniffers
- Hard disk copies or remove disks and take to your forensic lab
- Forensics applicable to internal waste/fraud/abuse
  - Good job opportunity!

Forensics -- target site

- Preserving evidence
- Disconnect from net -- don't reboot? Open debate...
- Analyze/record system status/disk
- Review IDS/firewall logs -- attacking host "trail"
  - Observing chain of custody
    - Hard and log evidence (floppies, dongles, hard drives, etc)
  - Hash and sign digital evidence (use time-stamp notary service too)
- Image disk copies
  - Maybe keep media, can recover over-written bits
- Audit trails (accurate clocks? IDS), attacked machine, etc.
- Record of time spent (parts of cost of attack)
- Getting back online

Network IDS logs for forensics

- Record all packet headers (lots of disk space!)
  - Handy for daily monitoring, statistical profiles
  - Useful for reconstructing an attack
- Follow packet trail of attacking host
  - What protocols did he use
  - Which internal hosts were visited
- Packet trail of attacked host
  - What host did it visit before/during/after attack
  - 2nd order, what hosts did those hosts visit...
- Post attack monitoring
  - Remote host visiting attacked host (especially on a backdoor port)
  - Additional traffic from attacking host (or do you have firewall block?)
  - Traffic from attacked host (cleanup incomplete?)
Are system logs admissible evidence?

- Easy to forge computer logs
- Hacker may have tampered with logs
- Computer records are considered hearsay
- However, handwritten records are acceptable
  - No e-mail logs accepted if being collected as part of day-to-day operations
  - Must be able to attest to their authenticity (logged to secure machine, time-stamped/MD5)
- Logs started after the attack, probably not admissible, but you may get clues from these logs that lead you to admissible evidence

Forensic tools

- Hash verifier tool, rpm -Va
- Registry/log file analyzers
- Dissasemblers, uncompress, decode, decrypt (password crackers)
- Hash/sign images
- grep, rm, strings, ls, ld, file, find, ad, reticat, mfsau
- Hash/evid images
- Deseamlers, uncompress, decode, encrypt (password crackers)
- Jit/dubbo/log file analyzers
- Histogram of "good" version of executables, libraries, data (html)

Forensics – attacker's site

- court orders
  - Wind-up/self
- Keylogger capture (get passwords)
- configuration of agent
  - Log evidence collected, maintain chain of custody
- Look for post-defensive logs with passwords
- Preservation of evidence (see checklist or call the cops)
- disk analysis
  - Make ISO image copies of drives
  - Hash/sign and digital watermark log files and media images
- Use tools to look for keywords/evidence in files
- Hidden files (stegan?)
- Deleted files
- Compressed/encrypted files
- Encrypted files
- Investigative tools (deassemblers)
- Check time offsets of hacker's PC clock
- Establish time line of events, file modifications, system logs
- prosecution and trial

in situ analysis

- risk in getting on and analyzing the attacked or attacking engine
  - Plant the attacker
  - Attn: command (self-taught) could delete evidence (zero the disk) and chain machine
  - Filter the evidence (Hidering principle)
  - Can't trust the data (root kit)
  - Mount your CD with your tools and make it only thing in PATH
  - Disconnect from net
  - Can learn a lot
  - (State of registry, active network ports, active processes (astra/ltrace), open files, process memory, swap space, who's logged in, shell history files (disk encryption/rootkit)
  - Keep a record of what you do/find (script)
  - State of registry, active network ports, active processes (strace/ltrace), open files, process memory, swap space, who's logged in, shell history files (disk encryption/rootkit)
  - At least (safe), if you don't have IDS data, start external sniffers and monitor traffic from suspect engines
  - Though leaving suspect engines running increases risk of more damage
- Information volatility: cache, register, display frame buffer, swap space
  - passwd, shadow, .rhosts, credit card numbers

Forensic tools

The Coroner’s toolkit

- Encase (also see FTK)
- Popular commercial tool
- Disk imaging/diagnosis/restore
- Parallel search
- Remote diagnosis (several)
- Most file systems (Linux, windows, ...)
UNIX forensics

- **in situ**—mount your CD, make it only thing in PATH
- Use dd and netcat to copy disk images to trusted host
  - On trusted host: `dd bs=1024 if=/dev/ad0s1e of=disk1.img` or `mount -t ext2 -o ro,loop=/dev/loop0 disk1.img /mnt/badboy`
  - On suspect host: `dd bs=1024 < /dev/ad0s1e | netcat 192.168.0.4 10000 –w 3` also could `dd if=/dev/kmem of=disk2.img` or `dd if=/dev/mem of=disk2.img`
  - On trusted host: `md5sum disk1.img > disk1.md5`

Malware analysis

- mystery executable
- Use strings, nm, ldd to peek inside
- Maybe disassemble & reverse engineer
- Careful—only run it on disposable machine/OS (VMware), then restore
  - Run it with strace to see system calls
  - Use lsof to see what files/ports it has open
  - Debugger to single-step
  - Pcat to dump process memory or `/proc/mem`
  - `kill -s SIGKILL` to dump core

Assignment 10

Windows malware analysis

- Figure out what a bad.exe does?
  - What files/registry entries does it modify/stash?
  - Capture keystrokes?
  - Talk on the net?
- Be safe—VMware and/or private net with disposable CPUs
- Tools
  - Regmon, regshot
  - Process explorer, PEiD, PEview
  - Lxs
  - Filemon
  - Fiddler
  - Network sniffer
  - IDA pro
  - Ollydbg or IDA pro
  - Netcat tcpdump

What you really want to know is how they got in?
  - registry snapshots (XP restore points), logs, IDS logs

UNIX intrusion response

<table>
<thead>
<tr>
<th>Action</th>
<th>Expertise</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go back to work</td>
<td>Anyone who can install</td>
<td>4 hours</td>
</tr>
<tr>
<td>Minimal work</td>
<td>Junior sys admin</td>
<td>5 to 1 day</td>
</tr>
<tr>
<td>Minimum recommended</td>
<td>Junior sys admin</td>
<td>1 to 2 days</td>
</tr>
<tr>
<td>Serious effort</td>
<td>Sys admin</td>
<td>2 days to 2 weeks</td>
</tr>
<tr>
<td>Fanaticism</td>
<td>Forensic specialists</td>
<td>Week to months</td>
</tr>
</tbody>
</table>

Sophistication of attack:
account/password
buffer overflow for network daemon
root access
rootkits (killing threads)
backdoors/engines/self-replicating/self-installing/self-destruct
physical access (hardware mods, keyboard sniffing)

US security/privacy laws

- Computer fraud & abuse act (CFAA) — computer access
- Gramm-Leach-Bliley act (GLBA) — financial data
- Health information portability accountability act (HIPAA)
- Children’s online privacy protection act (COPPA)

Legal morass

- Federal laws
- Computer fraud and abuse act
- Identity theft
- Data breach
- Gramm-Leach-Bliley act (GLBA)
- HIPAA
- Children’s online privacy protection act (COPPA)
- Constitution
- Fourth Amendment
- Privacy
- First Amendment
- Jurisdiction
- Evidence
- Hearsay
- Testimony
- Jury
- Sentencing
- Adjudication
- Appeal

- Some state/local laws
- Criminal use of computing devices
- Computer fraud
- Data breaches
- Identity theft
- Theft
- Wiretapping
- Search & seizure
- Data breaches
- Identity theft
- Computer fraud
- Privacy
- Evidence
- Hearsay
- Testimony
- Jury
- Sentencing
- Adjudication
- Appeal
- Some state/local laws
- Criminal use of computing devices
- Computer fraud
- Data breaches
- Identity theft
- Theft
- Wiretapping
- Search & seizure
- Data breaches
- Identity theft
- Computer fraud
- Privacy
- Evidence
- Hearsay
- Testimony
- Jury
- Sentencing
- Adjudication
- Appeal

Sentencing guidelines

- Potential/actual loss
- Level of sophistication of attack
- For commercial or personal benefit
- Malicious intent
- Messin’ with national defense, national security, justice
- Messin’ with critical infrastructure
- Threat to people, public health

Detection & response is as important as prevention!

recall

Cost-benefit analysis for the attacker (Clark & Davis ’95)

\[ M_b + P_b > O_{cp} + O_{cm}P_aP_c \]

- \( M_b \): monetary benefit to attacker
- \( P_b \): psychological benefit to attacker
- \( O_{cp} \): cost of committing the crime
- \( O_{cm} \): cost of conviction to the attacker
- \( P_a \): probability of arrest
- \( P_c \): probability of conviction

Becoming a certified crypto geek

- SANS/GIAC
  - security administrator, management, operations, network engineer, sysadmin, legal, audits, security expert
- Certified Information Systems Security Professional (CISSP)
  - 250 question exam
  - security mgmt, architecture, access control, application development, operations security, physical security, networking, security engineering, risk assessment, security analysis, cryptography, network security
- CIW security analyst
- Cyber-investigator certification
  - EnCase, AIS, CEI, CFE, BSA, Cisco, RSA

These credentials may qualify you as an “expert” witness

Next time...

IPsec
Kerberos
Trusted systems and secure OS

Lectures:
1. Risk, viruses
2. UNIX vulnerabilities
3. Authentication & hashing
4. Random number classical crypto
5. Block ciphers DES, RC5
6. AES, stream ciphers RC4, LFSR
7. MIDDLETERM
8. Public key crypto RSA, D-H
9. ECC, PKC, signatures
10. PKC, SSL
11. Network vulnerabilities
12. Network defenses, IDS, firewalls
13. IPv6, VPN, Kerberos, secure OS
14. Secure coding, cryptos APIs
15. review