



Chapter 14

Denial of Service

- ❑ Denial of Service Threats
- ❑ System Implementation Vulnerabilities
- ❑ Countermeasures

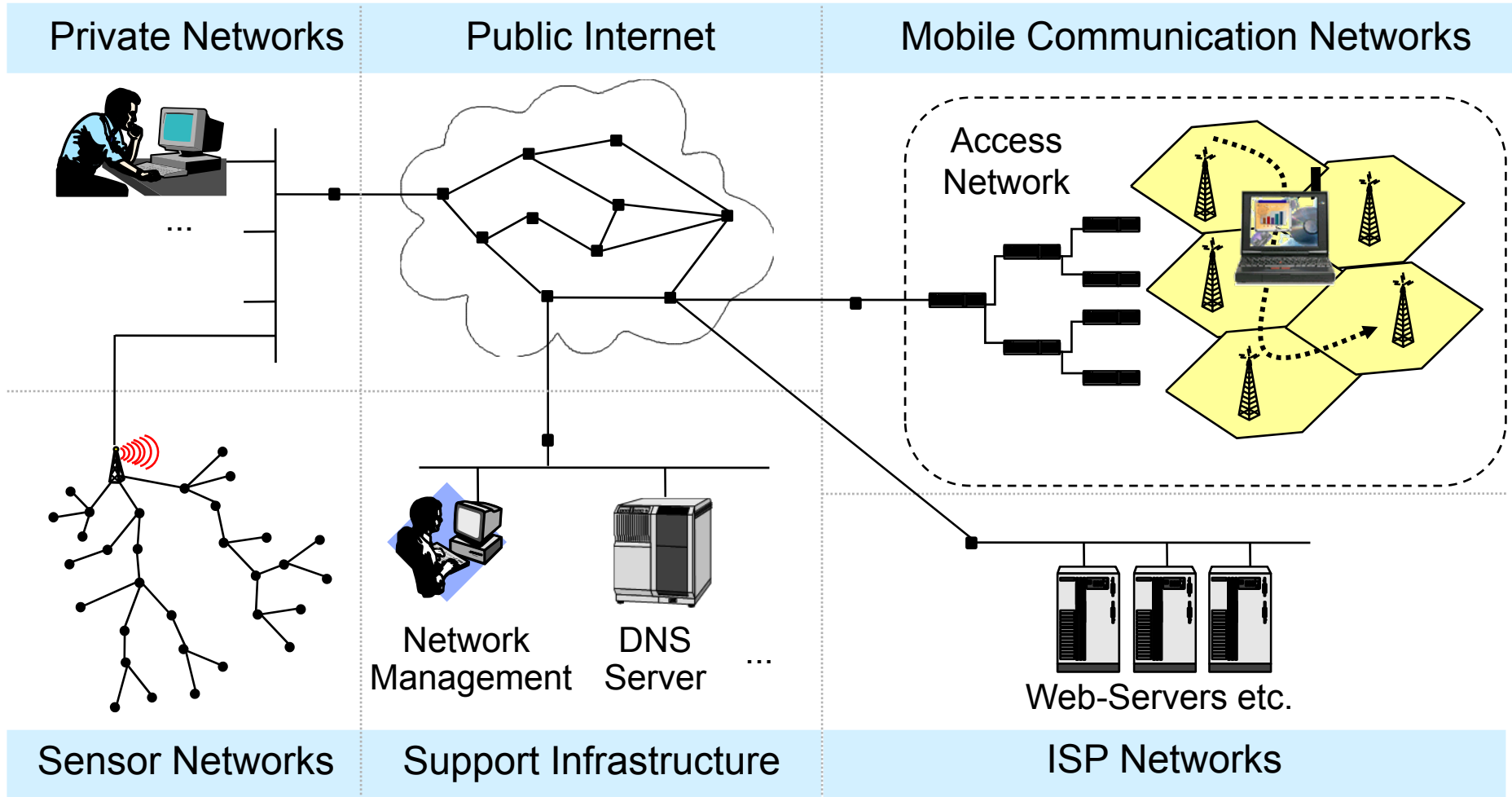
Motivation: A Changing World



- ❑ Mobile communication networks and ubiquitous availability of the global Internet have already changed dramatically the way we
 - ❑ communicate,
 - ❑ conduct business, and
 - ❑ organize our society
- ❑ With current research and developments in sensor networks and pervasive computing, we are even creating a new networked world
- ❑ However, the benefits associated with information and communication technology imply new vulnerabilities

→ Increasing dependence of modern information society on availability and secure operation of communication services

A High Level Model for Internet-Based IT-Infrastructure

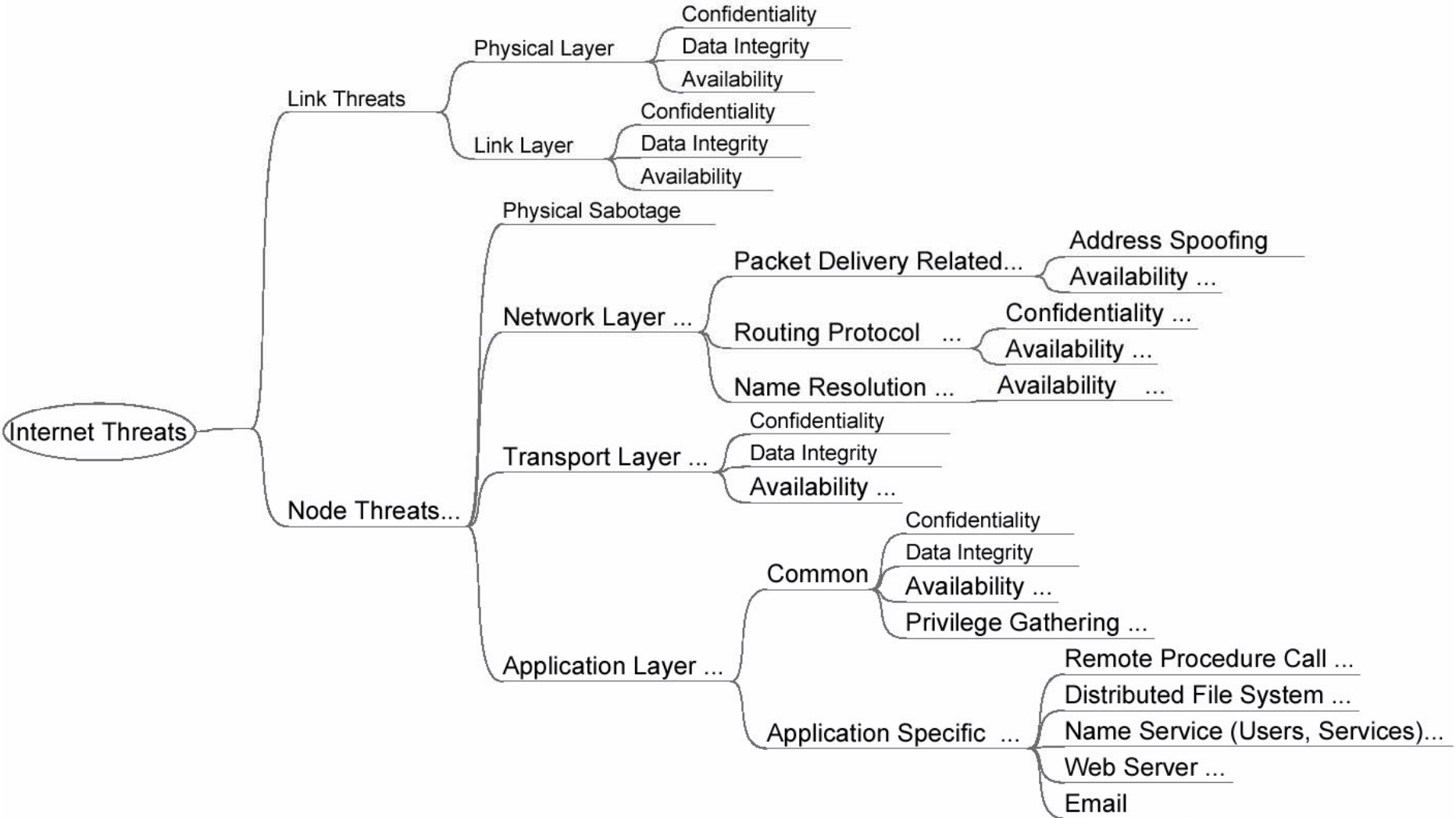


Categorizing Attacks



- ❑ Who / which device is attacking?
 - ❑ Normal user device located outside the infrastructure:
 - Examples: PC, PDA, mobile phone, ...
 - Commanded by a normal user not aware of what he is doing, or
 - Hacked and commanded by a malicious attacker
 - ❑ Device located inside the infrastructure:
 - Examples: router, management workstation, ...
 - Either deliberately placed by an attacker inside the infrastructure, or
 - Being part of the genuine infrastructure but hacked and commanded by a malicious attacker
- ❑ Which layer(s) is the attack aiming at?
 - ❑ Physical, MAC / Data Link, Network, Transport, Application
- ❑ Which kind of attack is performed?
 - ❑ Attacking user data PDUs: eavesdropping, replay, modification, ...
 - ❑ Resource depletion: TCP-SYN flood, SMURF attack, ...

A High Level Threat Tree for Internet-Based IT-Infrastructure



A Specific Example: Threats to Routing Protocols



- ❑ Threats to routing can be characterized according to:
 - ❑ *Threat source*:
 - Subverted link or subverted / rogue router
 - ❑ *Threat consequence* (generic):
 - Disclosure of (routing) information
 - Deception of other routers (e.g. with forged messages)
 - Disruption of normal (router) operation
 - Usurpation (= gaining control over a routers operation, e.g. by “stealing” traffic originally to be routed by that router)
 - ❑ *Threat consequence zone*:
 - Single node / part of a network / whole Internet
 - ❑ *Threat consequence period*:
 - Only during attack / for a certain period of time

(characterization mostly according to [Barbir2004])

Problems Beyond Simple Peer-to-Peer BGP Security



- ❑ Address space “ownership” verification:
 - ❑ Who has been assigned an IP address range and has thus the right to announce this range / delegate the announcement of this range?
- ❑ Autonomous System (AS) authentication:
 - ❑ To whom has a claimed AS-number actually been assigned?
- ❑ Router authentication and authorization (relative to an AS):
 - ❑ Are the entities pretending to belong to an autonomous system authentic?
- ❑ Route and address advertisement authorization:
 - ❑ Who is allowed to announce specific address ranges / routes
- ❑ Route withdrawal authorization:
 - ❑ Who is allowed to withdraw a route?

→ Need for further security measures, one approach for this is S-BGP
Basically, S-BGP introduces PKI-based attestations into routing [Kent2000]

Ensuring Availability: The Key Challenge for the Next Years



- ❑ Security of transmitted information in the sense of confidentiality, authenticity, etc. is well researched and many network security protocols have been developed & standardized during the past decade
 - ❑ Examples: PPP/PPTP, L2TP, IPSec, SSL/TLS, SSH, GSM/GPRS/UMTS security protocols,
- ❑ In “infrastructure networks” (like the Internet), routing threats can be effectively countered by deploying PKI-based approaches like S-BGP

→ However, **ensuring *availability*** of our IT- and communication infrastructure requires more than can be realized by standard network security protocols, and thus **turns out to be the major challenge** for the next years of security research!

- ❑ In the following, we will concentrate on two major issues:
 - ❑ Denial of Service (DoS)
 - ❑ System implementation vulnerabilities

Denial of Service



- ❑ What is Denial of Service?
 - ❑ *Denial of Service (DoS) attacks aim at denying or degrading legitimate users' access to a service or network resource, or at bringing down the servers offering such services*

- ❑ Motivations for launching DoS attacks:
 - ❑ Hacking (just for fun, by “script kiddies”, ...)
 - ❑ Gaining information leak (→ 1997 attack on bureau of labor statistics server; was possibly launched as unemployment information has implications to the stock market)
 - ❑ Discrediting an organization operating a system (i.e. web server)
 - ❑ Revenge (personal, against a company, ...)
 - ❑ Political reasons (“information warfare”)
 - ❑ ...

Denial of Service Attacking Techniques



- ❑ *Resource destruction* (disabling services):
 - ❑ Hacking into systems
 - ❑ Making use of implementation weaknesses as buffer overrun
 - ❑ Deviation from proper protocol execution
- ❑ *Resource depletion* by causing:
 - ❑ Storage of (useless) state information
 - ❑ High traffic load (requires high overall bandwidth from attacker)
 - ❑ Expensive computations (“expensive cryptography”!)
 - ❑ Resource reservations that are never used (e.g. bandwidth)
- ❑ Origin of malicious traffic:
 - ❑ Genuineness of source addresses: either genuine or forged
 - ❑ Number of sources:
 - single source, or
 - multiple sources (*Distributed DoS, DDoS*)

Examples: Resource Destruction

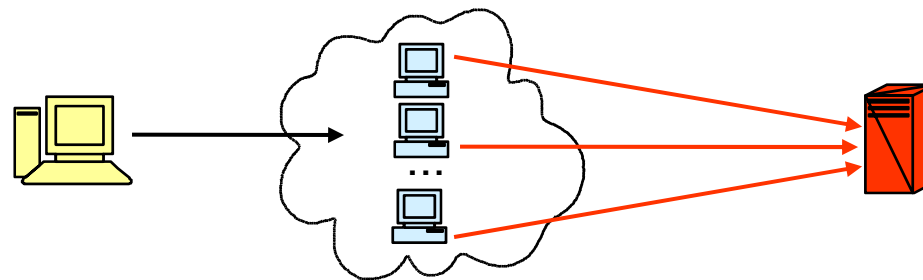


- ❑ *Hacking:*
 - ❑ Exploiting weaknesses that are caused by careless operation of a system
 - ❑ Examples: default accounts and passwords not disabled, badly chosen passwords, social engineering (incl. email worms), etc.
- ❑ *Making use of implementation weaknesses:*
 - ❑ See later slides on security aware system design & implementation
- ❑ *Deviation from proper protocol execution:*
 - ❑ Example: exploit IP's fragmentation & reassembly
 - Send IP fragments to broadcast address 192.168.133.0
 - Operating systems with origins in BSD often respond to this address as a broadcast address
 - In order to respond, the packets have to be reassembled first
 - If an attacker sends a lot of fragments without ever sending a first / last fragment, the buffer of the reassembling system gets overloaded
 - As some routers use BSD-based TCP/IP stacks, even the network infrastructure can be attacked this way!

Resource Depletion Example 1: Abusing ICMP



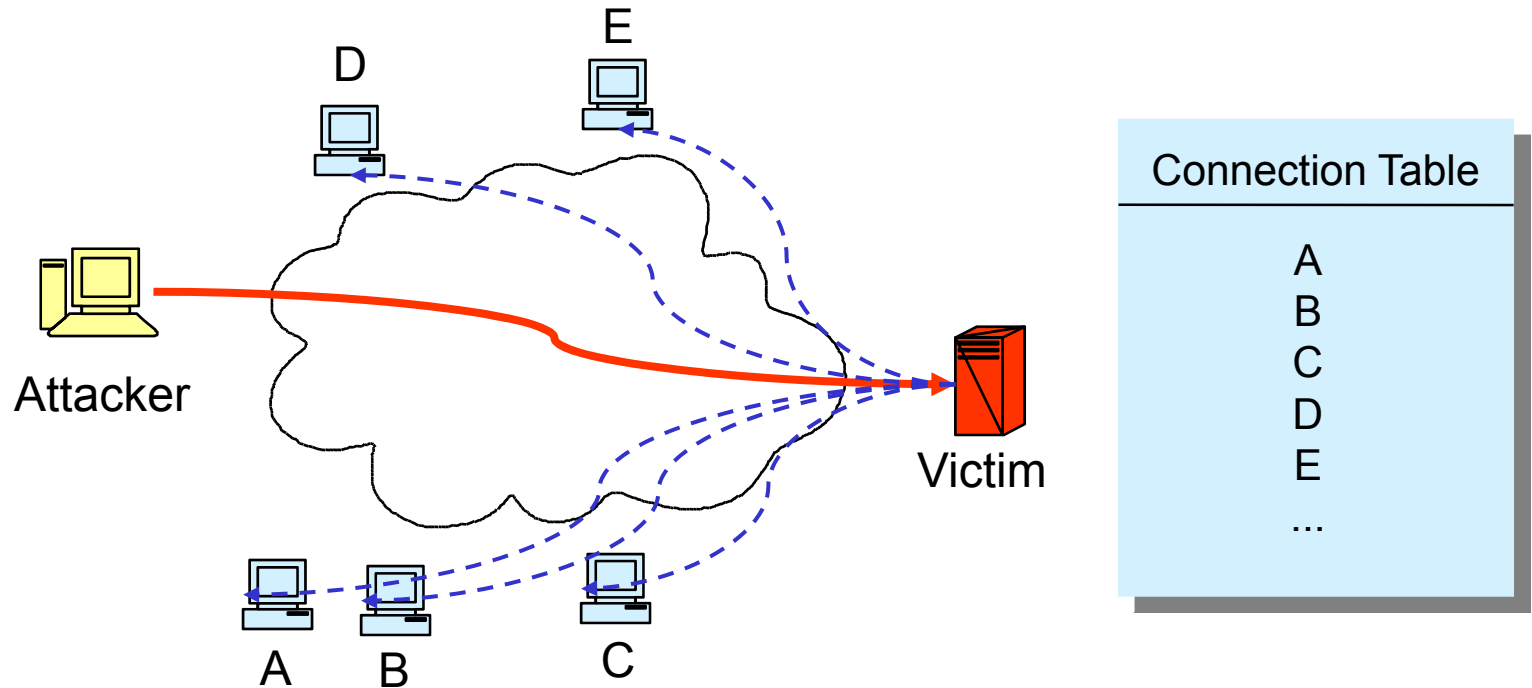
- ❑ Two main reasons make ICMP particularly interesting for attackers:
 - ❑ It may be addressed to broadcast addresses
 - ❑ Routers respond to it
- ❑ The *Smurf* attack - ICMP echo request to broadcast:
 - ❑ An attacker sends an ICMP echo request to a broadcast address with the source address forged to refer to the victim
 - ❑ Routers (often) allow ICMP echo requests to broadcast addresses
 - ❑ All devices in the addressed network respond to the packet
 - ❑ The victim is flooded with replies to the echo request
 - ❑ With this technique, the network being abused as an (unaware) attack amplifier is also called a *reflector network*:



Resource Depletion Example 2: TCP-SYN Flood



- ❑ Category *Storage of useless state information:*
 - ❑ Here: TCP-SYN flood attack

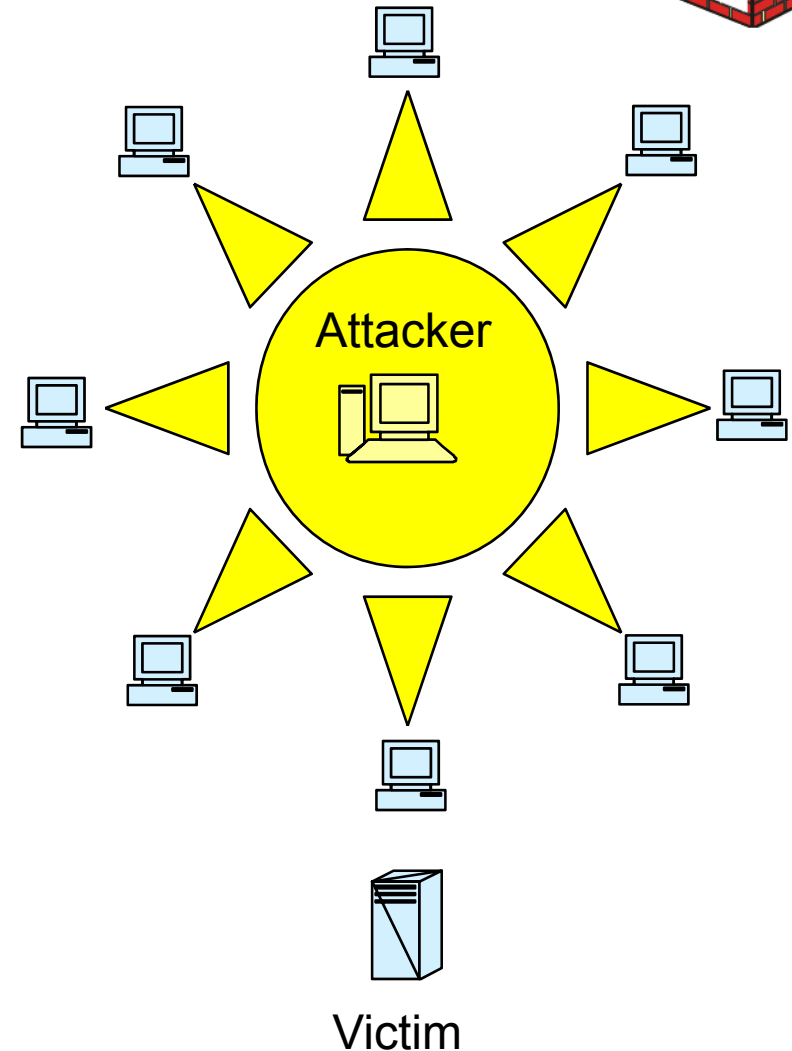


- TCP SYN packets with forged source addresses (“SYN Flood”)
- - - → TCP SYN ACK packet to assumed initiator (“Backscatter”)

Resource Depletion with Distributed DoS



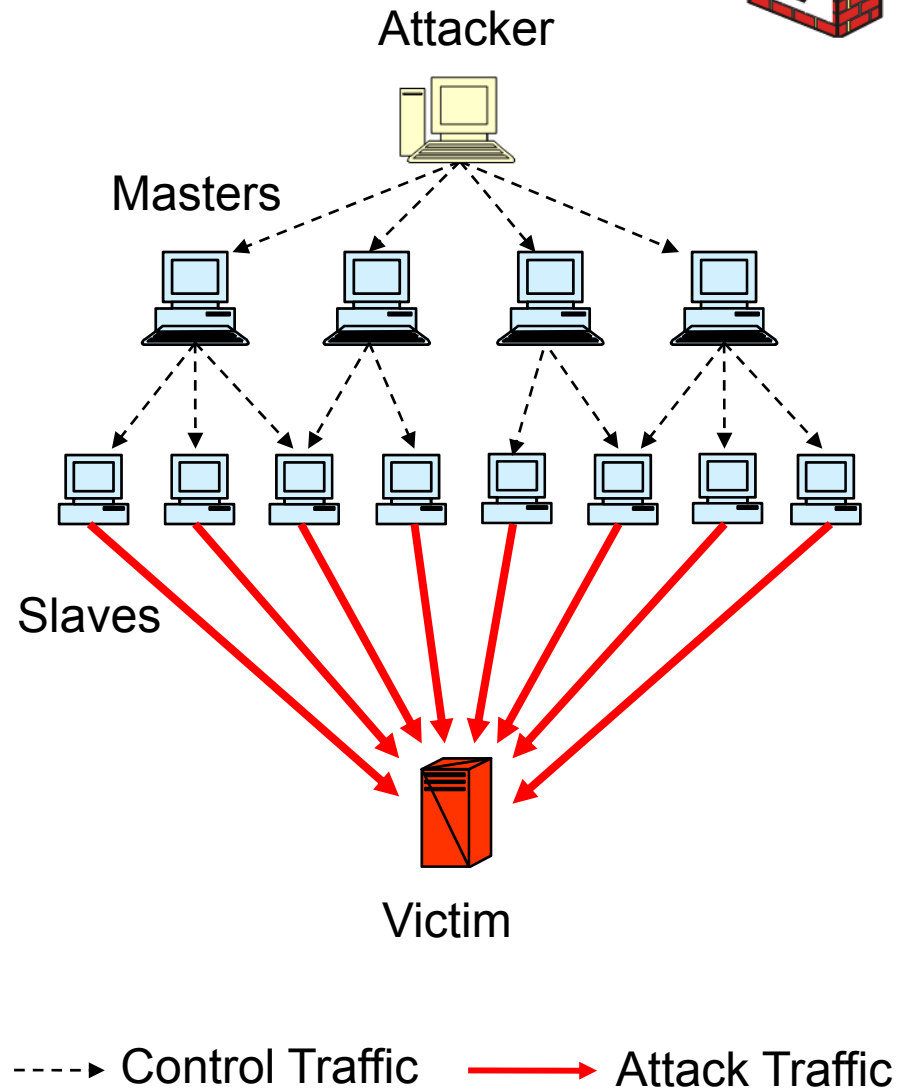
- ❑ Category *Overwhelming the victim with traffic*
- ❑ Attacker intrudes multiple systems by exploiting known flaws
- ❑ Attacker installs DoS-software:
 - ❑ „Root Kits“ are used to hide the existence of this software
- ❑ DoS-software is used for:
 - ❑ Exchange of control commands
 - ❑ Launching an attack
 - ❑ Coordinating the attack



Resource Depletion with Distributed DoS

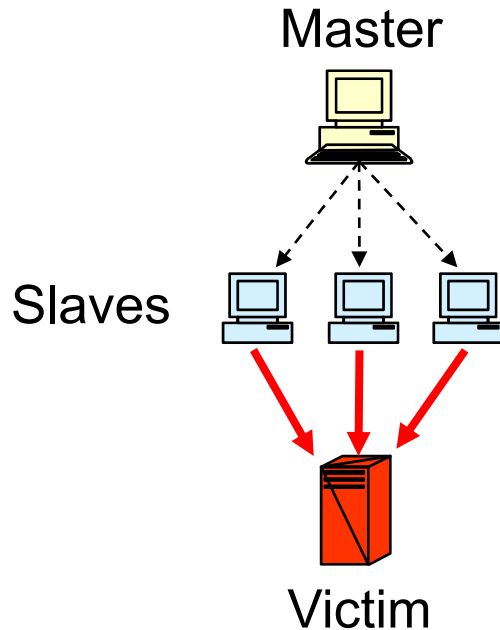


- ❑ The attacker classifies the compromised systems in:
 - ❑ Master systems
 - ❑ Slave systems
- ❑ Master systems:
 - ❑ Receive command data from attacker
 - ❑ Control the slaves
- ❑ Slave systems:
 - ❑ Launch the proper attack against the victim
- ❑ During the attack there is no traffic from the attacker

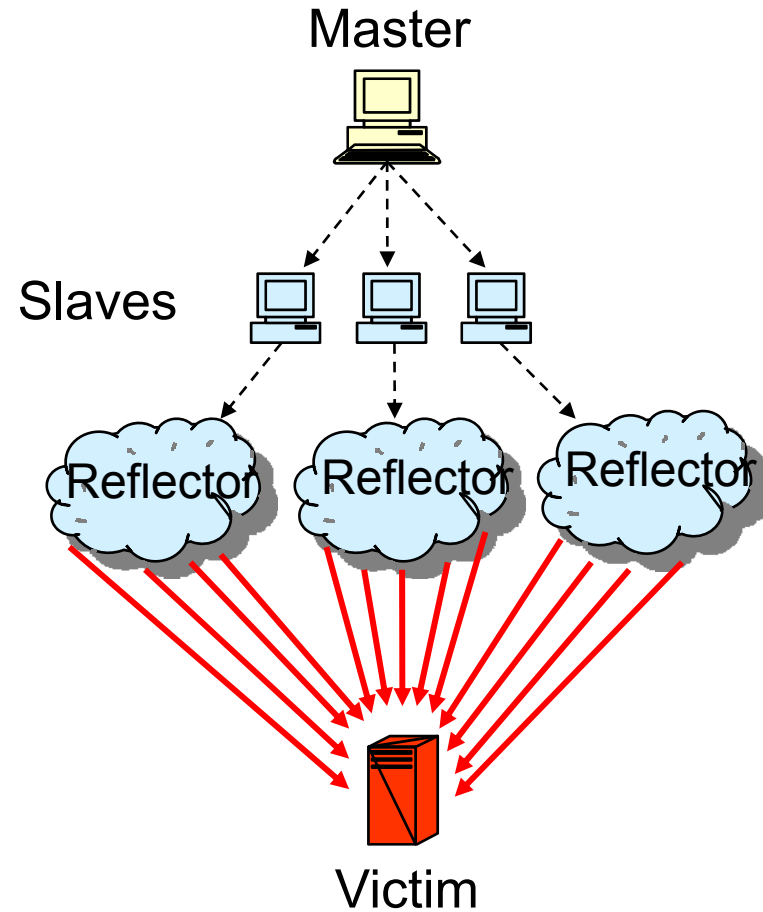




Different Attack Network Topologies



a.) Master-Slave-Victim

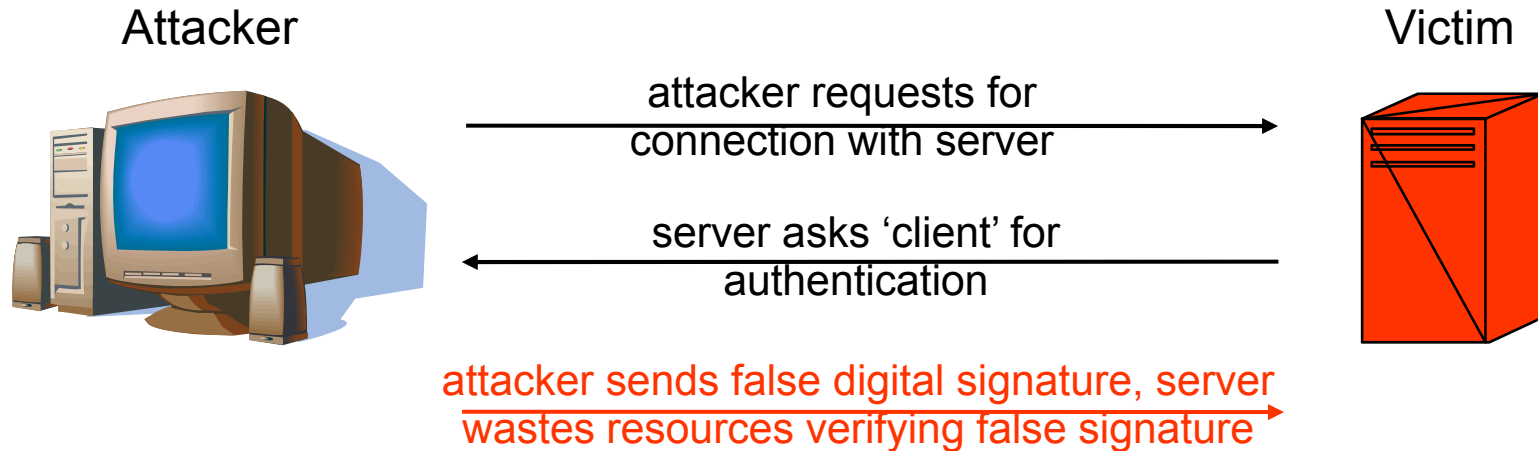


b.) Master-Slave-Reflector-Victim

Resource Depletion with CPU Exhaustion



- ❑ Category *CPU exhaustion by causing expensive computations*:
 - ❑ Here: attacking with bogus authentication attempts



- ❑ The attacker usually either needs to receive or guess some values of the second message, that have to be included in the third message for the attack to be successful
- ❑ Also, the attacker, must trick the victim *repeatedly* to perform the expensive computation in order to cause significant damage

→ Be aware of DoS-Risks when introducing security functions into protocols!!!

Problems of Practical System Security



- ❑ It is impossible to prove security of any moderately complex system
 - ❑ Complexity of a system makes it hard to understand, analyze and secure
- ❑ Software is at root of all common security problems:
 - ❑ Security holes and vulnerabilities are result of bad software design and implementation
 - ❑ There is too much information from too many sources for system administrators to keep up with patches for security vulnerabilities
 - ❑ System administrators are another class of “victims” of poorly-written software

Vulnerabilities Reported to the Computer Emergency Response Team (CERT)

Year	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007 Q1-3
Vulnerabilities	171	345	311	262	417	1090	2437	4129	3784	3780	5990	8064	5568

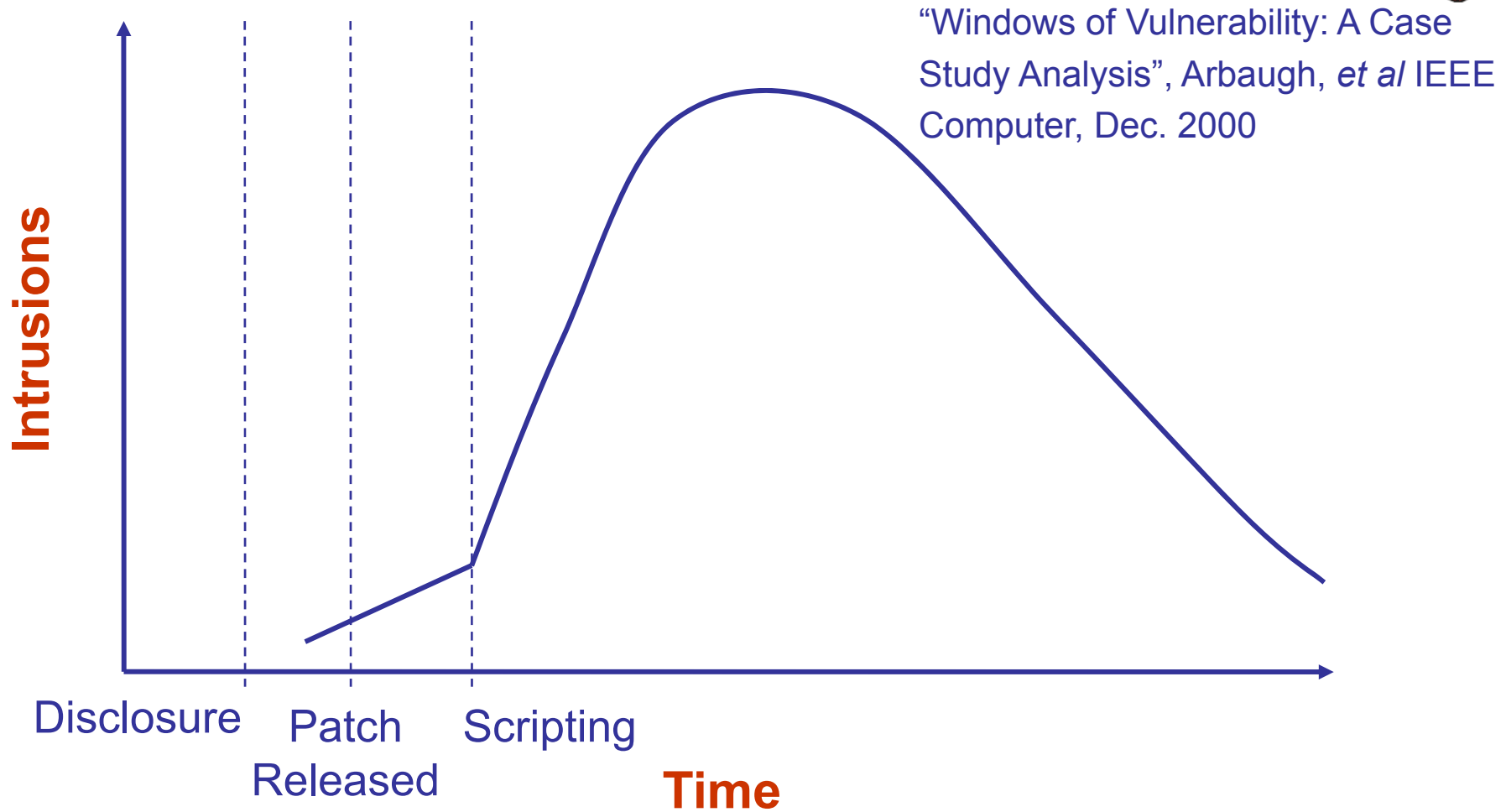
http://www.cert.org/stats/vulnerability_remediation.html

Technical Trends Leading to System Vulnerabilities



- ❑ Size and complexity of modern information systems and corresponding programs:
 - ❑ Widespread use of low-level programming languages like C, C++ that do not protect against simple types of attack
 - ❑ Improper configuration by retailers, administrators and users
- ❑ Degree to which systems have become extensible:
 - ❑ Extensible host accepts updates or extensions called “mobile code”, e.g., plug-ins
 - ❑ Extensibility makes it hard to prevent malicious code from slipping in
 - ❑ How to predict any possible extension to a product ...
- ❑ Other trends impacting security:
 - ❑ Lack of diversity in popular computing environments (i.e. pervasiveness of the Microsoft platform in end systems and the Unix OS in routers and servers)
 - ❑ “Internet time phenomenon”: short duration of development cycles – compressed development schedules & specs poorly written (if written)

A Typical “History” of a Vulnerability...



- ❑ Vulnerabilities are mostly exploited *after* a patch has been released!

System Vulnerabilities: Basic Attacking Styles



- ❑ Origin of attacks:
 - ❑ Remote attacks: attacker breaks into a machine connected to same network, usually through flaw in software
 - ❑ Local attacks: malicious user gains additional privileges on a machine (usually administrative)
- ❑ Main attacking techniques:
 - ❑ *Buffer overflow:*
 - Intentional manipulation of program state by causing an area of memory to be written beyond its allocated limits
 - ❑ *Race condition:*
 - Exploiting non-atomic execution of a series of commands by inserting actions that were “unforeseen” by the programmer
 - ❑ *Exploiting trust in program input / environment:*
 - It is often possible to maliciously craft input / environment variables to have deleterious side effects
 - Programmers are often unaware of this

Identifying Vulnerable Systems with Port Scans



❑ Background

- ❑ Identification of vulnerable systems / applications in order to identify systems to compromise
- ❑ Automated distribution of worms

❑ Scan types

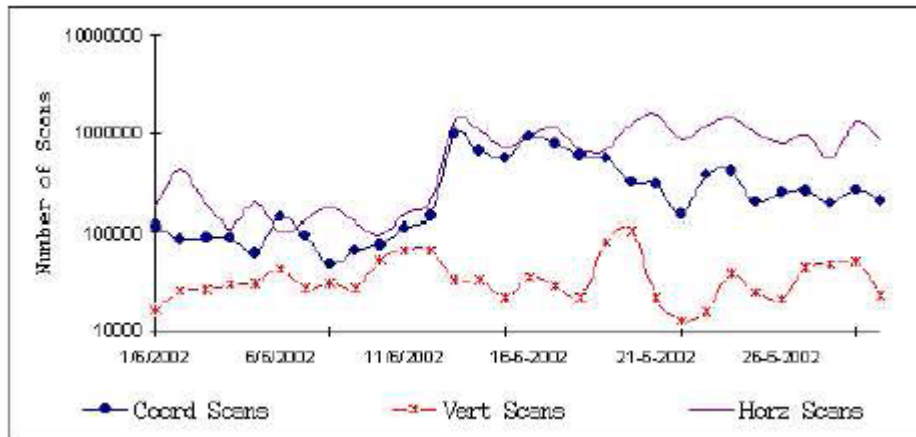
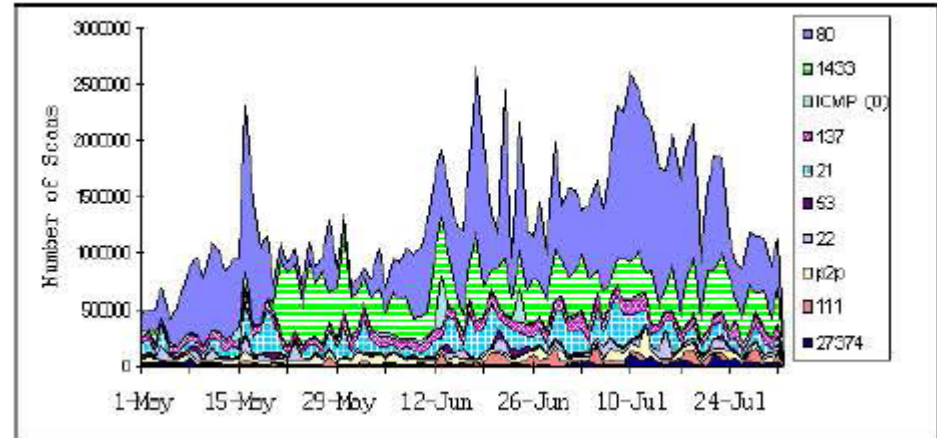
- ❑ Vertical scan: sequential or random scan of multiple (5 or more) ports of a single IP address from the same source during a one hour period
- ❑ Horizontal scan: scan of several machines (5 or more) in a subnet at the same target port from the same source during a one hour period
- ❑ Coordinated scan: scans from multiple sources (5 or more) aimed at a particular port of destinations in the same /24 subnet within a one hour window; also called distributed scan
- ❑ Stealth scan: horizontal or vertical scans initiated with a very low frequency to avoid detection

Identifying Vulnerable Systems with Port Scans



- Scan characteristics
 - Port distribution
 - Source distribution

Scan rates for top 10 destination port categories between May-July, 2002.



Distribution of coordinated, horizontal and vertical scans for the month of June, 2002

Source: [Yegneswaran2003]

Countering Attacks: Three Principle Classes of Action



❑ *Prevention:*

- ❑ All measures taken in order to avert that an attacker succeeds in realizing a threat
- ❑ Examples:
 - Cryptographic measures: encryption, computation of modification detection codes, running authentication protocols, etc.
 - Firewall techniques: packet filtering, service proxying, etc.
- ❑ Preventive measures are by definition taken *before an attack takes place*
- ➔ Attention: it is generally impossible to prevent every potential attack!

❑ *Detection:*

- ❑ All measures taken to recognize an attack *while or after it occurred*
- ❑ Examples:
 - Recording and analysis of audit trails
 - On-the-fly traffic monitoring and intrusion detection

❑ *Reaction:*

- ❑ All measures taken in order react to *ongoing or past attacks*

Prevention: Defense Techniques Against DoS Attacks



- ❑ Defenses against disabling services:
 - ❑ Hacking:
 - Good system administration
 - Firewalls, logging & intrusion detection systems
 - ❑ Implementation weakness:
 - Code reviews, stress testing, etc.
 - ❑ Protocol deviation:
 - Fault tolerant protocol design
 - Error logging & intrusion detection systems
 - “DoS-aware protocol design”:
 - Be aware of possible DoS attacks when reassembling packets
 - Do not perform expensive operations, reserve memory, etc., before authentication

Prevention: Defense Techniques Against DoS Attacks



- ❑ Defenses against resource depletion:
 - ❑ Generally:
 - Rate Control (ensures availability of other functions on same system)
 - Accounting & Billing (“if it is for free, why not use it excessively?”)
 - Identification and punishment of attackers
 - ❑ Authentication of clients plays an important role for the above measures
 - ❑ Expensive computations: careful protocol design, verifying the initiator’s “willingness” to spend resources himself (e.g. “client puzzles” [JuBr99])
 - ❑ Memory exhaustion: stateless protocol operation
- ❑ Concerning origin of malicious traffic:
 - ❑ Defenses against single source attacks:
 - Disabling of address ranges (helps if addresses are valid)
 - ❑ Defenses against forged source addresses:
 - Ingress Filtering at ISPs (if the world was an ideal one...)
 - “Verify” source of traffic (e.g. with exchange of “cookies” [TL00])
 - ❑ Widely distributed DoS: ???

Prevention: Countering CPU Exhaustion with Client Puzzles



- ❑ Basic idea:
 - ❑ Upon a new request, a server generates a new task (“client puzzle”) that the client has to solve before it will be served
 - ❑ Client puzzles can be easily generated and verified by a server, while clients must use a significant amount of computational resources in order to solve them
 - ❑ This may protect servers at the early stage of a normal authentication where the computations are the most CPU intensive
- ❑ Reusable client puzzles according to Aura et. al:
 - ❑ Server periodically broadcasts random number N_s and difficulty level k
 - ❑ Every client can then create a solution to a new instance of this puzzle by:
 - Generating a fresh random number N_c
 - Determining with brute force search (= trying all possible values) an X such that:

$$H(C, N_s, N_c, X) = \underbrace{00000000}_k Y$$

Prevention: Stateless Protocol Design



□ Basic idea:

- Avoid storing information at server, before DoS attack can be ruled out
- So, as long as no assurance regarding the client has been reached all state is “stored” in the network (transferred back and forth)
- Of course, data integrity of state has to be verified by server

Stateful Operation		Stateless Operation	
1. C → S: Msg_1		1. C → S: Msg_1	
2. S → C: Msg_2	S stores $State_{S1}$	2. S → C: $Msg_2, State_{S1}$	
3. C → S: Msg_3		3. C → S: $Msg_3, State_{S1}$	
4. S → C: Msg_4	S stores $State_{S2}$	4. S → C: $Msg_4, State_{S2}$	
...		...	

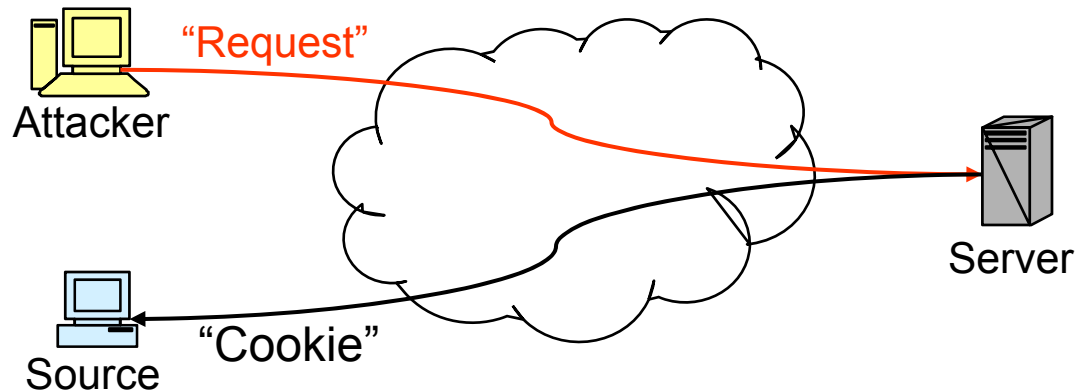
- Drawback: requires higher bandwidth and more message processing

Prevention: Verifying the Source of a Request



❑ Basic idea:

- ❑ Before working on a new request, verify if the “initiator” can receive messages send to the claimed source of the request



- ❑ Only a legitimate client or an attacker which can receive the “cookie”, can send the cookie back to the server
 - ❑ Of course, an attacker must not be able to guess the content of a cookie
- ## ❑ Discussion:
- ❑ Advantage: allows to counter simple spoofing attacks
 - ❑ Drawback: requires one additional message roundtrip

Reaction: Penetrate & Patch (the common naïve approach)



- ❑ Basic strategy:
 - ❑ Wait until software is compromised and then patch it to close the breach
 - ❑ “... desperately trying to come up with a fix to a problem that is being actively exploited by attackers.”
- ❑ Problems:
 - ❑ Developers can only fix problems they know about
 - ❑ Patches rushed out to fix problems can introduce new ones
 - ❑ Patches often only fix the symptoms
 - ❑ Patches are often unapplied
 - ❑ Patching communication protocol weaknesses is particularly difficult!
- ❑ Finding and fixing defects after release seems to be:
 - ❑ the most expensive approach
 - ❑ far too slow (cf. curve on vulnerability exploitation)!!!

→ Faster detection and reaction to ongoing attacks in networks is needed!!!

Intrusion Detection in Networks



- ❑ As we have seen, prevention is not sufficient in practice:
 - ❑ Because it is too expensive to prevent all potential attack techniques
 - ❑ Because legitimate users get annoyed by too many preventive measures and may even start to circumvent them (introducing new vulnerabilities)
 - ❑ Because preventive measures may fail:
 - Incomplete or erroneous specification / implementation / configuration
 - Inadequate deployment by users (just think of passwords...)

- ❑ What can be attained with intrusion detection?
 - ❑ Detection of attacks and attackers
 - ❑ Detection of system misuse (includes misuse by legitimate users)
 - ❑ Limitation of damage (if response mechanisms exist)
 - ❑ Gain of experience in order to improve preventive measures
 - ❑ Deterrence of potential attackers

Some Upcoming Challenges



- ❑ The introduction of Internet protocols in classical and mobile telecommunication networks also introduces the Internet's DoS vulnerabilities to these networks
- ❑ Programmable end-devices (PDAs, smart phones) may constitute a large base of possible slave nodes for DDoS attacks on mobile networks
- ❑ Software defined radio implementation may even allow new attacking techniques:
 - ❑ Hacked smart phones answer to arbitrary paging requests
 - ❑ Unfair / malicious MAC protocol behavior
 - ❑ ...
- ❑ The ongoing integration of communications and automation (→ sensor/actuator networks) may enable completely new DoS threats
 - Availability of communication infrastructures can not be ensured by preventive measures alone!

Summary (what do I need to know)



- ❑ Categories of denial of service attacks
 - ❑ Resource destruction
 - ❑ Resource depletion
 - ❑ Different origins of attacks
 - ❑ Some examples

- ❑ Candidate solutions for countermeasures

Additional References



- [Amo94] E. Amoroso. *Fundamentals of Computer Security Technology*. Prentice Hall. 1994.
- [BMY04] A. Barbir, S. Murphy, Y. Yang. *Generic Threats to Routing Protocols*. Internet Draft (work in progress), draft-ietf-rpsec-routing-threats-06, 2004.
- [Eck03] C. Eckert. *IT-Sicherheit: Konzepte, Verfahren, Protokolle*. zweite Auflage, Oldenbourg Verlag, 2003.
- [Gar96] Simson Garfinkel and Gene Spafford. *Practical Internet & Unix Security*. O'Reilly, 1996.
- [GW03] M.G. Graff, K.R. van Wyck. *Secure Coding*. O'Reilly, 2003
- [Kle04] Tobias Klein. *Buffer Overflows und Format-String-Schwachstellen*. dpunkt.verlag, 2004.
- [KLS00] S. Kent, C. Lynn, K. Seo. *Secure Border Gateway Protocol (S-BGP)*. IEEE Journal on Selected Areas in Communications. Vol. 18, No. 4, April 2000.
- [NN01] S. Northcutt, J. Novak. *Network Intrusion Detection - An Analyst's Handbook*. second edition, New Riders, 2001.
- [Sch03] G. Schäfer. *Netzicherheit - Algorithmische Grundlagen und Protokolle*. dpunkt.verlag, 435 Seiten Broschur, 44.- Euro, 2003.
- [VM02] J. Viega, G. McGraw. *Building Secure Software*. Addison-Wesley, 2003.