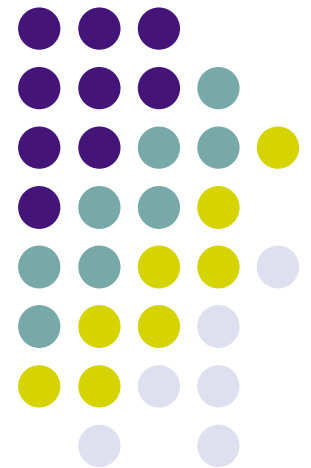


Network Intrusion Detection Systems

Beyond packet filtering

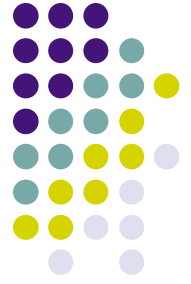




Goal of NIDS

- Detect attacks as they happen:
 - Real-time monitoring of networks
- Provide information about attacks that have succeeded:
 - Forensic analysis
- Passive systems: monitoring and reporting
- Active systems: corrective measures adopted
- Good place to establish a NIDS: The perimeter network, or DMZ.

Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*



Strategies

- Often NIDS are described as being composed of several parts
 - Event generator boxes
 - Analysis boxes
 - Storage boxes
 - Counter-measure boxes
- Analysis is the most complex element, and can use protocol analysis as well as anomaly detection, graph analysis, etc.



Elements of a NIDS

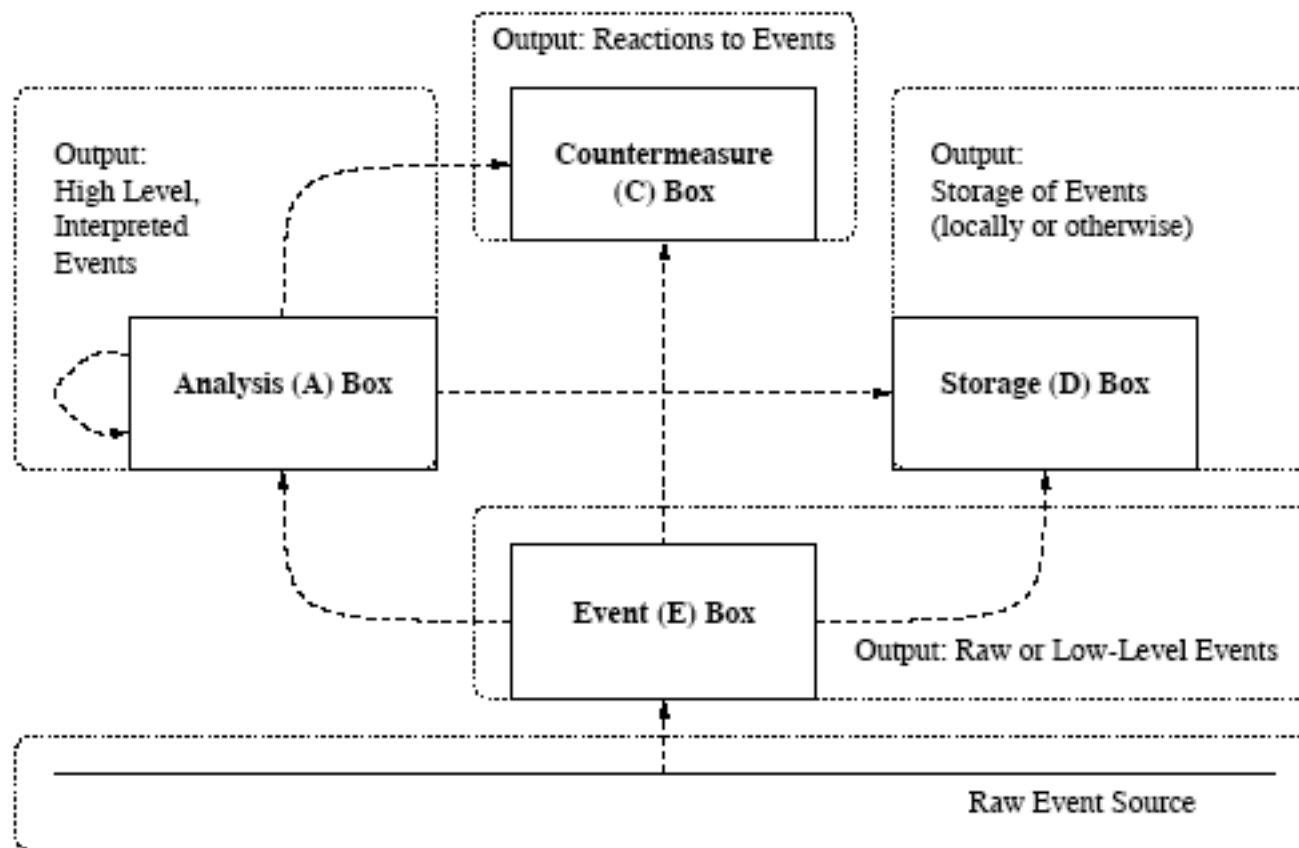


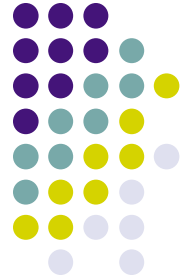
Figure 1: CIDF component relationships

Host based vs. Network based

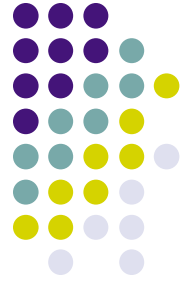


- Host based: Operating system log analyses
 - Semantically rich: Contain information about the state of the system
- Network based: Direct analysis of network traffic
 - Complete: Sees all the network events, not only those conveyed up to the higher levels of the operating system.
 - Unobtrusive: Does not degrade network or host performance

Common analysis techniques



- Attempts pattern-matching against certain known attack types.
 - For instance, substring matching.
- Passive protocol analysis.
 - Emulate the sequence of protocol events to detect attacks.



Difficulties inherent in NIDS

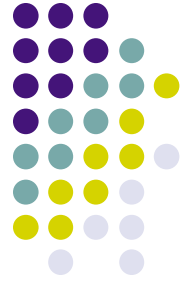
- What defines an attack is not a packet, but its induced behavior on the receiving host.
 - NIDS must determine this behavior
- NIDS runs in a different machine, even a different part of the network.
 - Proper function of the NIDS may require of each host being protected:
 - Knowledge of its place in the network topology
 - Knowledge of its TCP/UDP implementation
 - OS-based behavior variance.

Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*

Influence of Network Topology

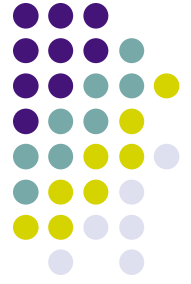


- If several internal routers exist between the network component where the NIDS resides, and where the receiver host resides:
 - TTL may result in some packets reaching the NIDS but not the receiver.
 - Some packets being dropped by filtering routers.



Influence of implementation

- UDP packets with incorrect checksum -- will be dropped or accepted? will be filtered?
- Packets with incorrect header fields.
- Fragmentation, overlap, and re-ordering issues.



Insertion attacks

- Means: Lead the NIDS into thinking a particular packet will be accepted by the receiving host, when it in fact will not.
- Goal: To prevent the NIDS from recognizing patterns (either for protocol analysis or signature recognition) by reconstructing an incorrect series of events



Example of intrusion attack

- NIDS performs signature analysis based on substring match: fragment the string into parts and add intermediate packets that are rejected by receiving host, but not by NIDS.

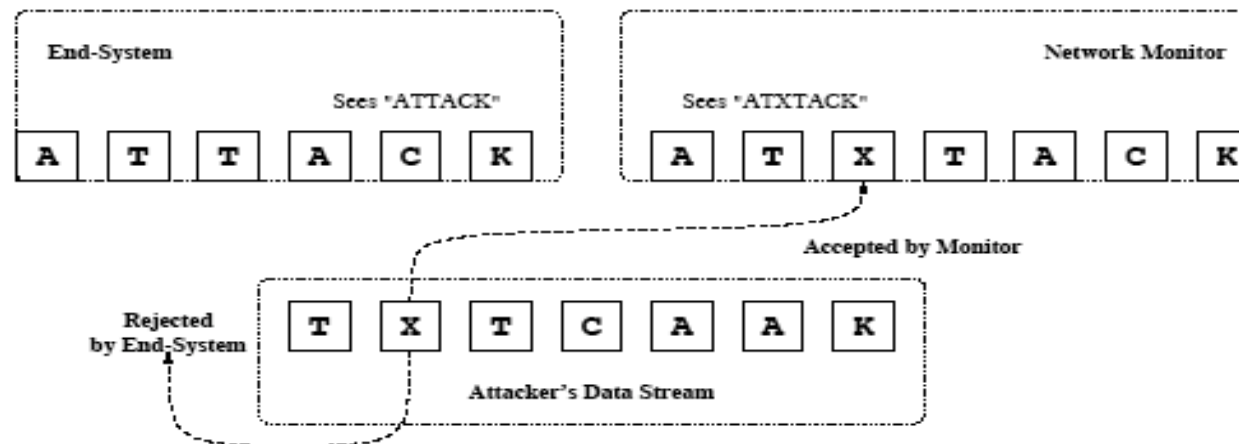
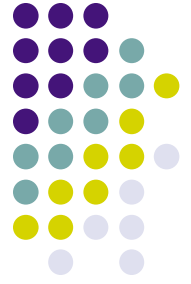


Figure 4: Insertion of the letter 'X'

Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*



Evasion attacks

- Means: Lead the NIDS into believing that a particular packet will be rejected by the host, when it will not.
- Goal: To prevent the NIDS from detecting an attack (via protocol analysis or signature analysis) by preventing the NIDS from reconstructing the correct sequence of packets processed by the receiving host.



Evasion example

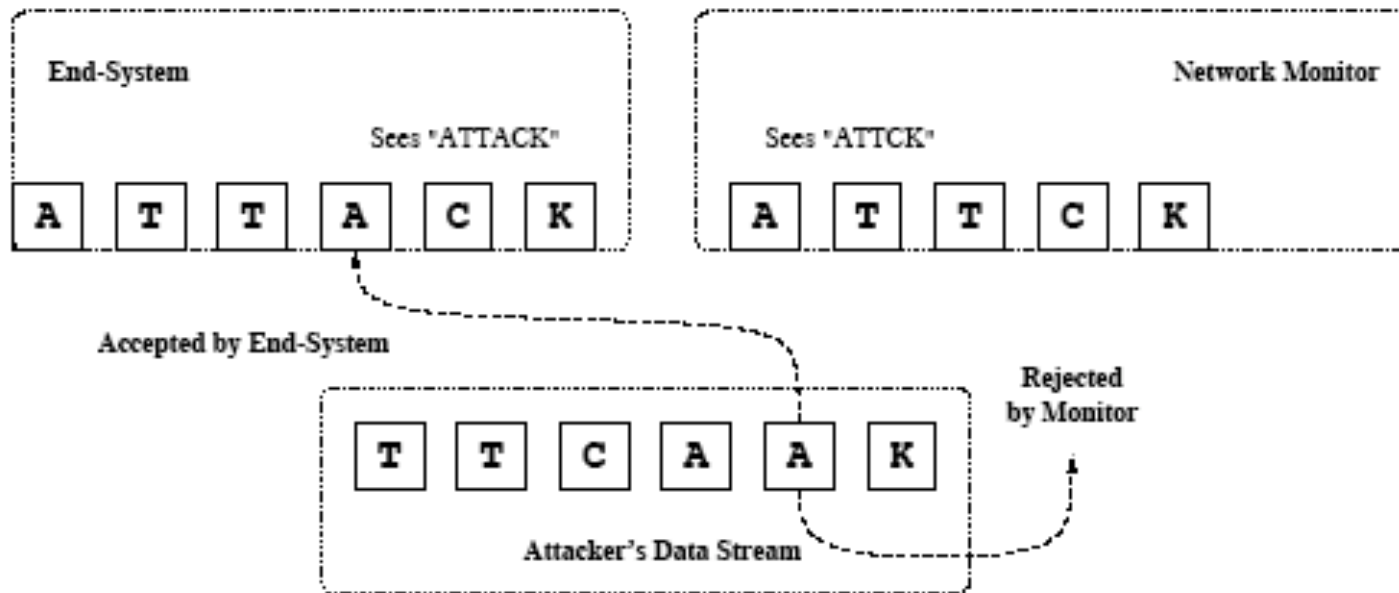


Figure 5: Evasion of the letter 'A'

Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*



Confusing the NIDS

- Some implementations of NIDS may allow evasion/ insertion attacks simply because the NIDS does not correctly implement all the steps of protocol verification.
 - An attacker specifically targets this.
- In what follows, we consider difficulties which are inherent with the design of NIDS systems, namely intrinsic ambiguity on what types of decisions the NIDS should take.

Ambiguity

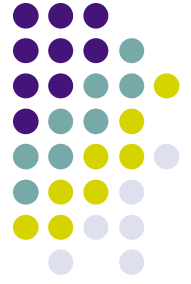


Section	Info Needed	Ambiguity
Section 4.1.1	Network Topology	IP TTL field may not be large enough for the number of hops to the destination
Section 4.1.1	Network Topology	Packet may be too large for a downstream link to handle without fragmentation
Section 4.1.2	Destination Configuration	Destination may be configured to drop source-routed packets
Section 4.3.1	Destination OS	Destination may time partially received fragments out differently depending on its OS
Section 4.3.3	Destination OS	Destination may reassemble overlapping fragments differently depending on its OS
Section 5.2.2	Destination OS	Destination host may not accept TCP packets bearing certain options
Section 5.2.2	Destination OS	Destination may implement PAWS and silently drop packets with old timestamps
Section 5.4.3	Destination OS	Destination may resolve conflicting TCP segments differently depending on its OS
Section 5.5.1	Destination OS	Destination may not check sequence numbers on RST messages

Figure 7: Ambiguities identified in this paper

Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*

Some evasion/insertion attacks



- Bad IP headers
 - Differences in NIDS' network and host network with respect to TTL and don't fragment (DF) bit.
- Bad IP options
 - Source-based packets filtered and variations in timestamp decisions
- Direct frame addressing:
 - Attacker in the same physical network as NIDS directs packet to NIDS (or to non-existing MAC address) but IP address of host.

Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*



IP packet fragmentation

- Large IP packets (larger than the size of the data-frames in the link layer) must be broken up into smaller packets.
- The IDS must be able to handle IP packet re-assembly correctly.
 - out-of-order fragments must be reordered.
 - fragments must be stored until all fragments for the packet are known.
 - DoS attack: Send partial IP packets



Packet fragmentation

- After some time, packet fragments must be discarded based on their arrival times, or the system will run out of memory.
 - If NIDS drops them faster than end system, there is opportunity for successful evasion attacks.
 - If NIDS keeps them longer than end system, there is opportunity for successful insertion attacks.
 - Coordinated attacks using many source/destination pairs can disable NIDS.



Overlapping fragments

- Two IP fragments may contain overlapping data.

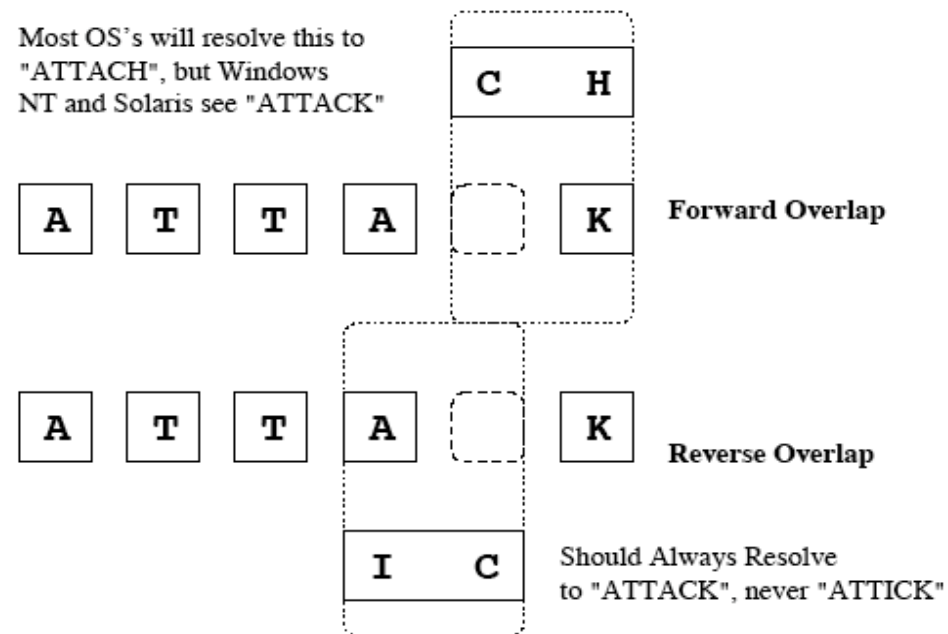


Figure 11: Forward and Reverse Overlap

Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*

Different OSes resolve this differently



Operating System	Overlap Behavior
Windows NT 4.0	Always Favors Old Data
4.4BSD	Favors New Data for Forward Overlap
Linux	Favors New Data for Forward Overlap
Solaris 2.6	Always Favors Old Data
HP-UX 9.01	Favors New Data for Forward Overlap
Irix 5.3	Favors New Data for Forward Overlap

Figure 12: IP fragment overlap behavior for various OS's

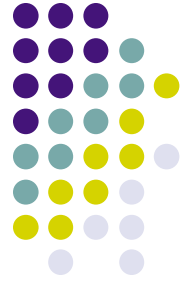
Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*



TCP layer problems

- For forensic reasons it is important to keep/analyze higher level protocol information.
- One such approach is called TCP connection monitoring
 - TCP packets can be assigned to connections, or at least requests to open connections.
 - A TCP session can be in a set of “states”
 - Established, Closed, ...
 - The NIDS and the end system should be state-synchronized for monitoring to succeed.

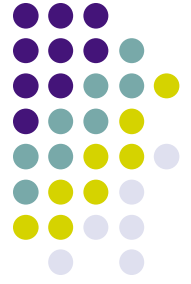
Pictures reproduced from Ptacek and Newsham.
*Insertion, Evasion and Denial of Service: Eluding
Network Intrusion Detection.*



TCB = TCP Control Block

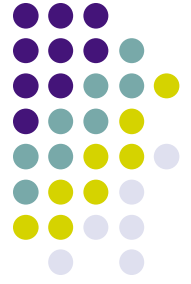
- A TCP monitoring NIDS must keep a TCB for every existing connection, with state, packet numbers, window, etc.
- TCBs must be created for new connections and should be discarded for closed connections.
 - TCB creation
 - TCB re-assembly
 - TCB teardown

TCB creation and re-assembly

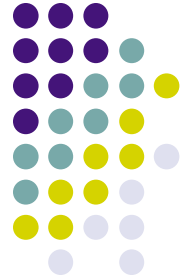


- Missed handshake sequences
 - TCP has a 3-way handshake. When to consider the connection has been established?
- Existing connections at boot time:
 - Insertion attacks
 - replay an old packet sequence number on an existing connection
 - send a packet on a closed connection
 - send a packet on a non-existing connection

TCP stream synchronization



- TCB re-assembly:
 - TCP data overlap
 - TCP time-window and acknowledgment strategies
- NIDS does not validate TCP packets in accordance with end system
 - TCP header, options, checksum
 - Wrapped sequence numbers



TCB teardown

- TCP connections are closed by sending FIN or RST packets.
- TCP connections do not “time-out”
 - DoS attack by never closing connections
- What to do with RST packets with wrong sequence numbers.
- TCP control information re-use after connection is closed.