Network Packet Inspection and Intrusion Detection

Marlon Winder
Laurence Wilson
Tolu Onibiyo
Idris Ozoya
Hassan Ayinde

Introduction

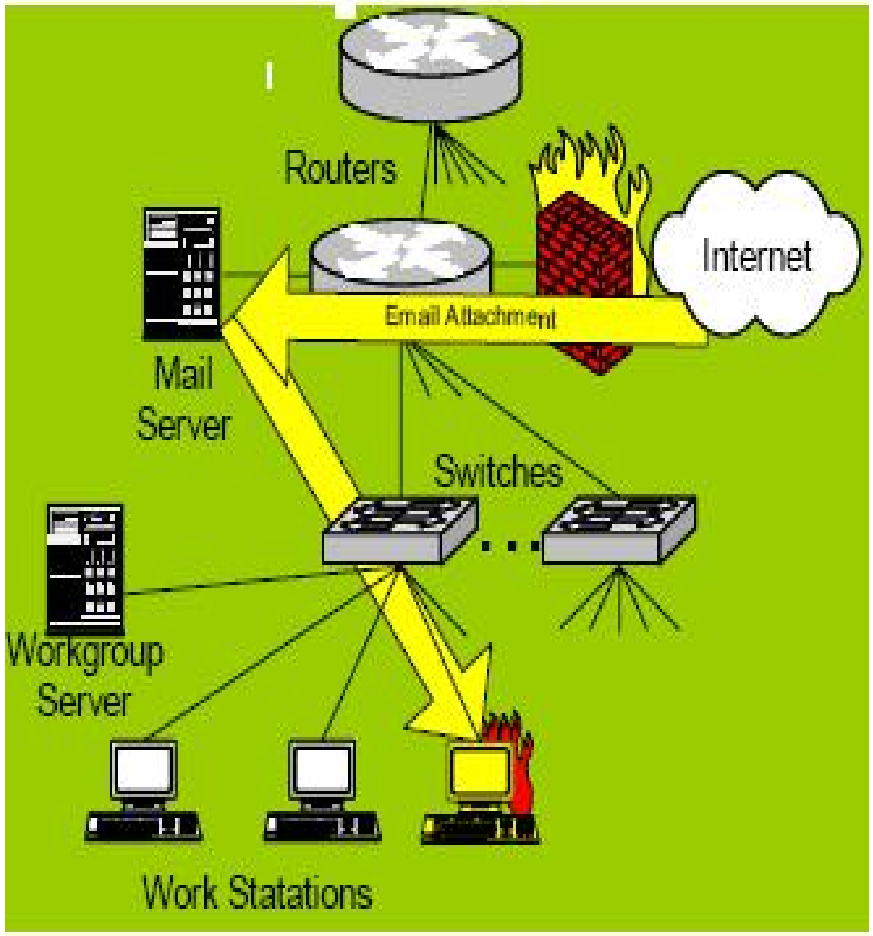
- Increase in network traffic volume and transmission speeds has given rise to the need for extremely fast packet processing

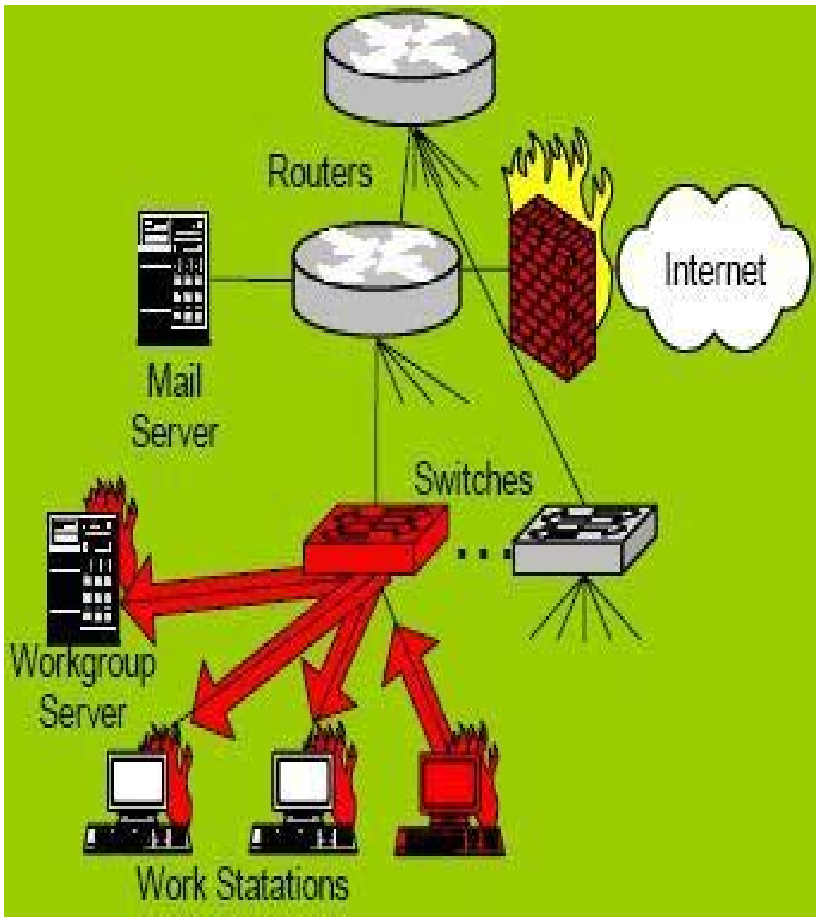
- Rapidly increasing network transmission speeds have marked the computationally heavy task of network packet inspection as an obvious bottleneck in the processing and forwarding of information across the network

Response Time to Network Packet

Peak Network Transmission speeds	Max. # of 64 Byte Packets / Second	Time To Respond (nanoseconds)		
		Total	Time Per Snort Rule, Given 1700 Rules	# of 5 ns Memory Accesses
10 Mb/s	19,531	51,200	30 ns	**6**
100Mb/s	195,313	5,120	3 ns	**< 1**
1Gb/s	1,953,125	512	.3 ns	0
10Gb/s	19,531,250	51	.03 ns	0

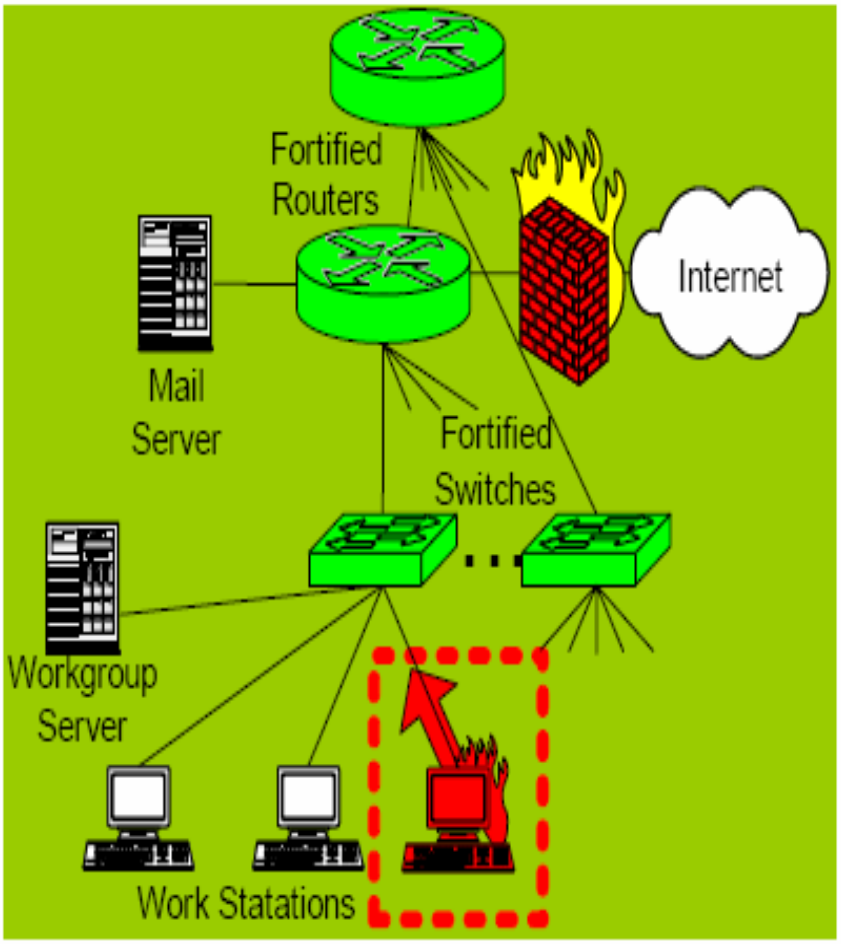
Network Infrastructure





Network incorporating routers and switches with packet inspection capabilities

- Significant research has been done on intrusion detection methodologies most of which are signature based
- Snort searches its rule-set to find any rules that match the packet under inspection
- Insufficient processor capacity makes Pattern matching algorithms such as Snort's inadequate



Possible Solutions

- Pure software based approach
  - Enhanced Parallel Processing
  - Snort Algorithm
- Hardware Based Approach
  - PIC16F877
  - Motorola 68000
  - Custom FPGA
- Hybrid Approach
  - Software
  - Hardware

Our Solution

- The proposed solution dedicates a processor to packet inspection and pattern matching
- The processor receives packets from a network backbone and classifies the packets into various protocols
- For Pattern matching, the header and payload of each classified packet is checked for any matches

Tasks and Project Management

- Write code and run simulations for classifier by November 13th 2007.
- Write code and run simulations for pattern matcher by December 6th 2007
- Write code and run simulations for UART (Universal Asynchronous Receiver and Transmitter) driver by January 1st 2008.
- Generate libraries for communication port to be incorporated into the User Interface module by January 31st 2008.
- Integrate and test hardware and software components by February 28th 2008.
- Evaluate and compare performance with existing technologies by March 5th 2008.
- Finalize user guide and specification sheet by March 10th 2008.
- Complete entire project by March 15th 2008.
- Demonstrate final product on EE Day.

Verification Plan and Deliverables

- The final deliverables include:
- Software program
  - User interface
  - Communication port link
- FPGA board (Processor)
  - Packet classifier
  - Pattern Matcher
  - UART driver
- EE Day Demonstration
  - **Audience Participation**
  - **Interactive Verification**

Costs and Resources

- FPGA board must cost less than $1000
- Content Process design must be completed and ready for testing by 3/1/2008

Conclusion

- Presents a solution to the problem.
- Save a lot in the long run in keeping networks safe.
- Classifies every protocol field of an incoming packet.
- Each classified protocol field has the ability to be analyzed.
- Adaptability and flexibility to additional resources from the review panel to the development of this product.