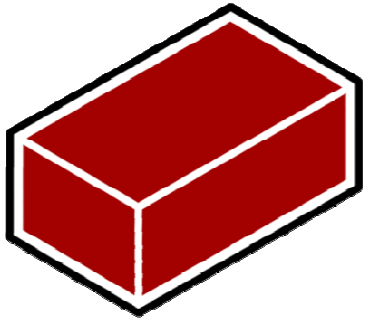


IDS/IPS Evasion Techniques

Alan Neville, BS.c, MS.c

anev@redbrick.dcu.ie

http://www.twitter.com/abnev

Speaker

- Completed BS.c in Computer Applications and MS.c in Security and Forensic Computing at DCU (2009, 2010)
- Previously worked in *Espion* and *DCU* in Honeypot related projects (2004, 2005)
  - Lead to several publications on Symantec's SecurityFocus portal – http://www.securityfocus.com
  - Development of an IPS system – *Pharos*
- Currently a Malware Research Analyst / Reverse Engineer at Symantec (2010)
  - Joined the fight against cyber crime! \o/

Agenda

- Intrusion Detection System (IDS)
- Intrusion Prevention System (IPS)
- Information Collection
- Problems with IDS
- Evasion Techniques
- Defeat Evasion Techniques
- Questions

Intrusion Detection System (IDS)

- So what are Intrusion Detection Systems?
  - A security technology that attempts to ***identify*** and ***report*** intrusions against computer systems and networks
  - Passively monitors network transactions for *suspicious patterns* of activity
    - Statistical anomaly detection
    - Signature based detection
  - Important component of a security system which compliments other security technologies
    - e.g. ACLs, firewalls, packet filters etc.

Intrusion Prevention Systems (IPS)

- An IPS is an *extension* to an IDS whereby a counter-measure component is introduced
- Performs actions to cease on-going attacks
  - Drop TCP connections (send RST packet)
  - Modify existing ACLs and other filter lists to restrict access
- Used to prevent further attacks occurring after an initial attack has been detected by the IDS component
- Also acts as a deterrent to future attacks

Information Collection (1)

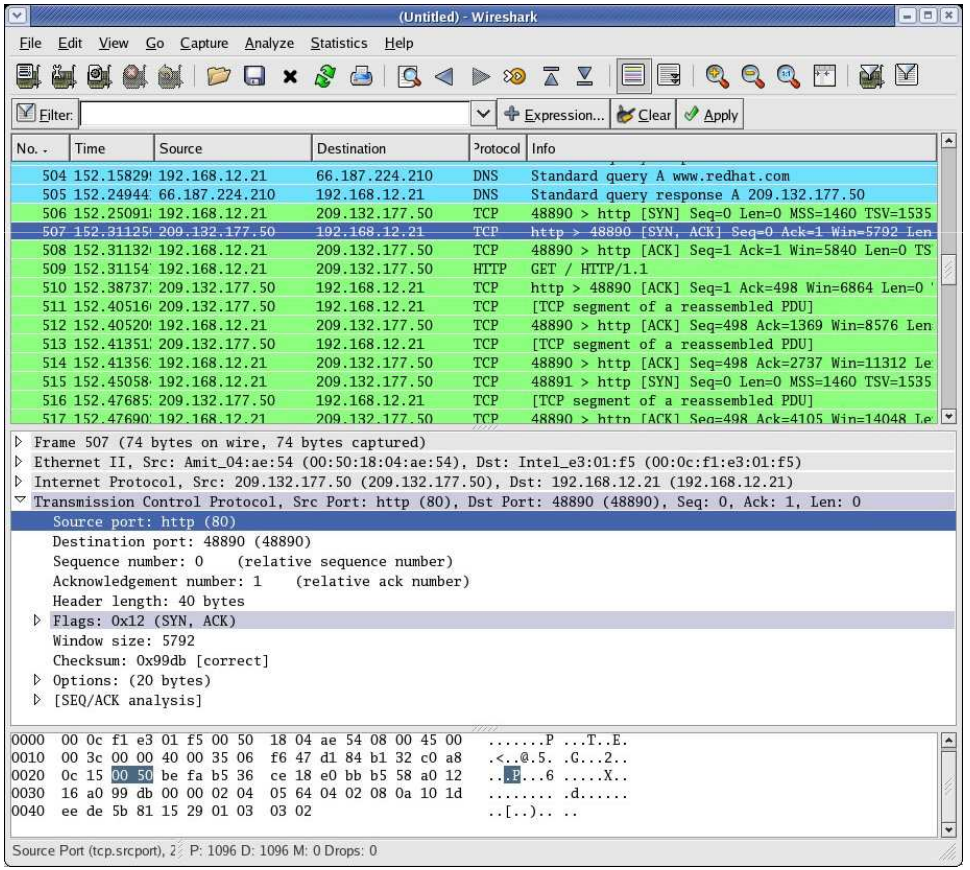
- IDS (multi-host) interpret raw traffic in order or build up a picture of what is happening on a network
  - Good at discerning attacks that involve low-level manipulation of the network
  - Parses packets, analysing the protocols used on the network
  - Extracts relevant information from collected packets for pattern matching

Information Collection (2)

- As IDS systems are *passive*, they only collect **copies** of packets involved in transactions of data over networks
  - E.g. Libpcap
    - *Wireshark, tcpdump, windump* etc
  - original packet may or may not have reached destination

Information Collection (3)

- Unobstructive
  - doesn't manipulate the data sent to/from end-systems
  - doesn't cause disruption to network
  - nor affects network performance
  - Difficult to evade from an attackers perspective



Problems with IDS (1)

- IDS systems are logical targets for attack
  - Seek to disable by force reporting of false information
    - False positives & false negatives
    - E.g. flood system with alerts to conceal real attack
- Packets which are considered invalid by the IDS may be accepted by end-systems
  - Packets deemed invalid by IDS will not generate alerts and therefore cannot invoke countermeasure activities

Problems with IDS (2)

- Inconsistencies may exist between IDS and end-systems
  - Each OS processes packets in it's own way
  - Most implementations try to keep as close to RFCs as possible
    - In reality, this doesn't happen and corners are cut
  - IDS systems may exhibit inconsistencies with how packets are handled which opens new avenues of attack

Introduction to Evasion

- In reality, evasion attacks are not as easy to exploit as theory leads to believe
  - Requires knowledge of the network in which to attack
    - Network topography
    - OS version information for end-systems and IDS platforms
- Not only do we need to generate packets considered invalid by the IDS, we may also need to evade pattern recognition mechanisms
  - IDS employs complex algorithms which vary drastically across different IDS platforms
  - Commercial IDS generally design super-secret proprietary pattern recognition algorithms which are difficult to reverse
  - This makes our job harder!

Evasion Techniques

- A. Circumventing Pattern Recognition Engines (PREs)
- B. Polymorphic Shell Code
- C. Session Splicing
- D. Fragmentation Attacks
  - Overlap
  - Time-outs
  - Time-to-Live (TTL)
- E. Denial of Service (DoS)

A. Pattern Matching Engines (1)

- Signature-based IDSs attempt extract payload information and attempt to match strings against *signatures* in order to generate alerts
  - These pattern matching engines can be complex
  - Commercial IDS difficult to analyse for obvious reasons

```
 alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC
/etc/passwd";flags: A+; content:"/etc/passwd"; nocase;
classtype:attempted-recon; sid:1122; rev:1;)
```

- Snort IDS Signature example
  - Trivial example which looks for "/etc/passwd" string
  - *nix file which contains user login information

A. Pattern Recognition Engines (2)

- Possible to evade detecting on this signature
  - /etc//\//passwd
  - /etc/rc.d/../../.\passwd
  - Countless other examples…
  - Most IDSs have powerful enough pattern matching engines which will detect this trivial type of evasion technique
- Need to employ more complex techniques which are harder to detect
  - String obfuscation and manipulating techniques
  - Hex (simple), Unicode and UTF-8 (advanced) are just some examples…

```
GET %65%74%63/%70%61%73%73%77%64 or  GET %65%74%63/%70a%73%73%77d
```

A. Pattern Recognition Engines (3)

- Unicode and UTF-8 provide unique identifiers for every character in every language
  - Managed by the Unicode Consortium
    - http://www.unicode.org/
  - Adopted by many contempory technologies
    - Java, LDAP, XML etc
  - UTF-8: represents the standard seven bit ASCII
  - IIS web server can interpert UTF-8
  - Apache also supports UTF-8 but is not enabled by default

A. Pattern Recognition Engines (4)

- How does UTF-8 help us evade IDS?
  - Can represent single characters in multiple ways which may evade signature-based detections
    - U+005C, 0x5C, C191C, E0819C – all represent "/"
    - U+0041, 0x41, U+100, U+0102, U+0104, U+01CD, U+01DE etc all represent the character 'A'
      - 30 unique representations of the letter 'A'
      - 34 unique representations of the letter 'E'
      - 83,060,640 unique representations of the string "AEIOU"
  - You get the idea…

B. Polymorphic Shell Code (1)

- A method of obfuscating buffer overflows (BoF) attacks against any service
  - Changes an exploit signature each time it is executed
- What is a buffer overflow?
  - An anomaly whereby a programming writing to a buffer overruns the buffer boundary, writing data to adjacent memory
  - Can be triggered by inputs to execute arbitrary code or alter flow of program

variable name	A								B	
value	'e'	'x'	'c'	'e'	's'	's'	'i'	'v'	25856	
hex	65	78	63	65	73	73	69	76	65	00

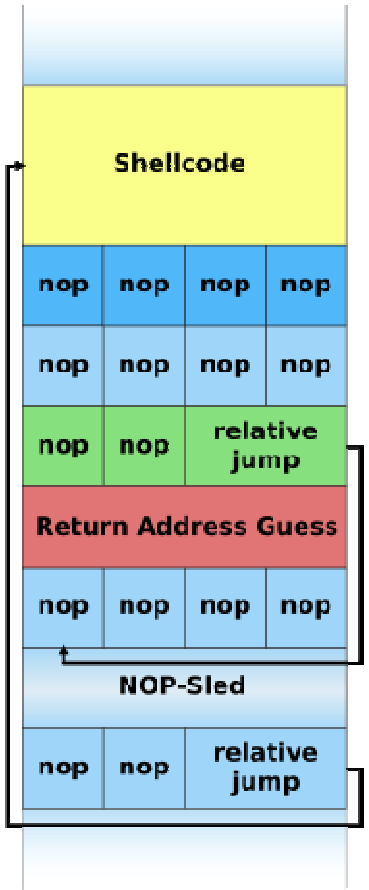
B. Polymorphic Shell Code (2)

- Based on anti-virus evasion techniques
  - Malware authors release viruses which constantly re-write themselves on the fly as part of their propagation mechanisms
- Polymorphic shell code used in IDS evasion is limited to buffer overflows

```
 alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT ssh CRC32
overflow /bin/sh"; flags:A+; content:"/bin/sh"; reference:bugtraq,2347;
reference:cve,CVE-2001-0144; classtype:shellcode-detect; sid:1324; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT ssh CRC32
overflow NOOP"; flags:A+; content:"|90 90 90 90 90 90 90 90 90 90 90 90
90 90 90
```

- Much more effective against signature-based than statistical anomaly-based detection engines

B. Polymorphic Shell Code (3)



- Example Snort detection signature looks for "/bin/sh" and no-op characters (0x90)
- Polymorphic code uses techniques to evade detection
  - There are up to 55 replacements for no-op instructions
  - Combined in a pseudo-random manner
  - Simple encryption algorithms (XOR) to create shellcode which does not resemble IDS signatures
  - Polymorphic pre-processors are used to help detect these type of "simple" techniques
    - E.g. Snort's *spp_fnord* module

C. Session Splicing (1)

- Involves splitting strings across several packets
  - String can be matched upon – "GET /etc/passwd"
  - However, a single character means nothing to an IDS
- By delivering data a few bytes at a time, string matching can be evaded
  - Can be spread out over a long period of time - Effective!
    - Apache – 6 minutes IIS - >15 minutes
  - Restricted to TCP – can't be used across all protocols!

C. Session Splicing (2)

```
 alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC whisker
space splice attack"; content:"|20|"; flags:A+; dsize:1;
reference:arachnids,296; classtype:attempted-recon; reference
```

- IDS systems need to monitor connections and context in order to detect this type of attack
  - Reassembles packet streams to identify such attacks
  - (More in-depth discussion on this later!)
- Difficult for IDSs to keep track of packet streams
  - Source, destination, source port, destination port
    - Collectively = TCP Control Block (TCB)
  - IDS has policies when to initiate session recording
    - E.g. 3-way handshake (3WH)
  - Can trick IDS to stop recording using RST, FIN tricks

D. IP Fragmentation (1)

- Reassembly issues exist within the IP Layer
- IP Fragmentation – RFC791
  - A field in the IP header which allows systems to break individual packets into smaller ones
  - The *offset* field acts as a marker
    - indicates where a fragment belongs in the context of the original packet
  - Last fragment is marked in IP header – flag indicates if additional fragments are to follow

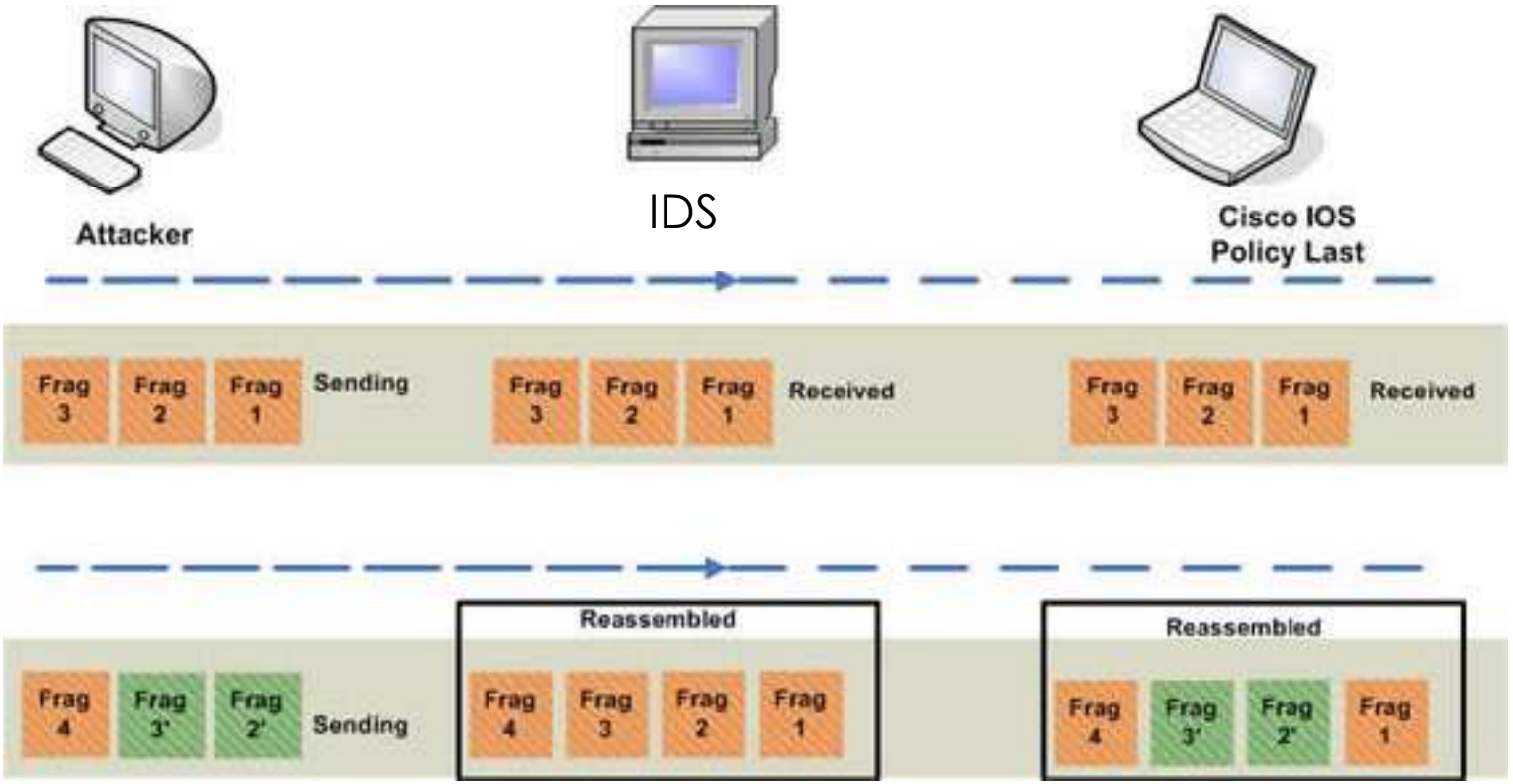
D. IP Fragmentation (2)

- IDS systems don't reconstruct packets until all fragments have arrived
  - Requires storage (caching) of all fragments until last one arrives
  - Susceptible to resource exhaustion
    - Can flood a network with partial fragment datagrams which is never completed (no final fragment)
    - Data is cached until memory is fully utilised

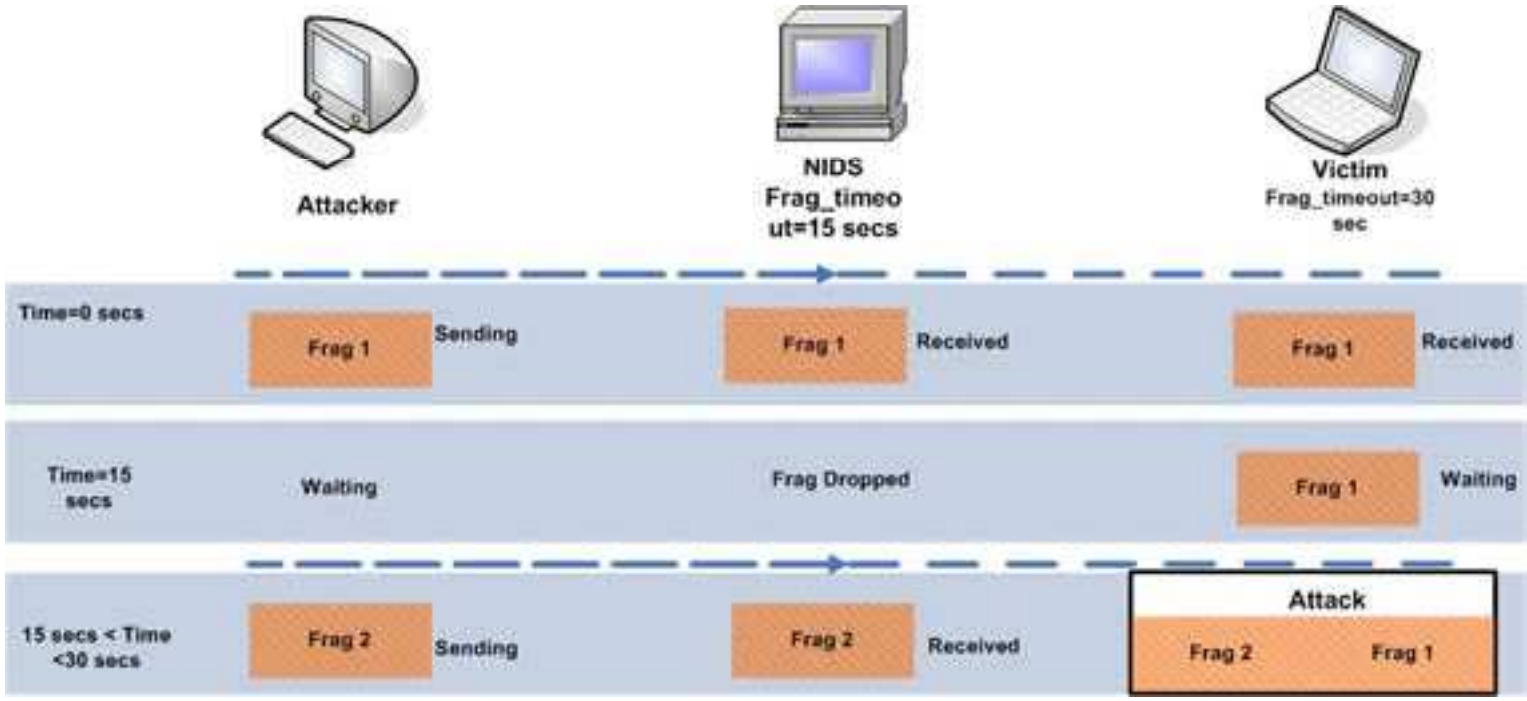
D. IP Fragmentation (3)

- Fragment Overlapping
  - Sending fragments of varying sizes out of order and in overlapping positions
  - IDS system may favor freshest fragment overwriting older data - *fragment conflict*
  - Reassembled packet at end-system can be different than what IDS sees thus effectively evading IDS
- Difficult to exploit but not impossible
  - Attacker must know the inconsistencies of fragment handling on IDS or end-system (e.g. can inspect code of network drivers)

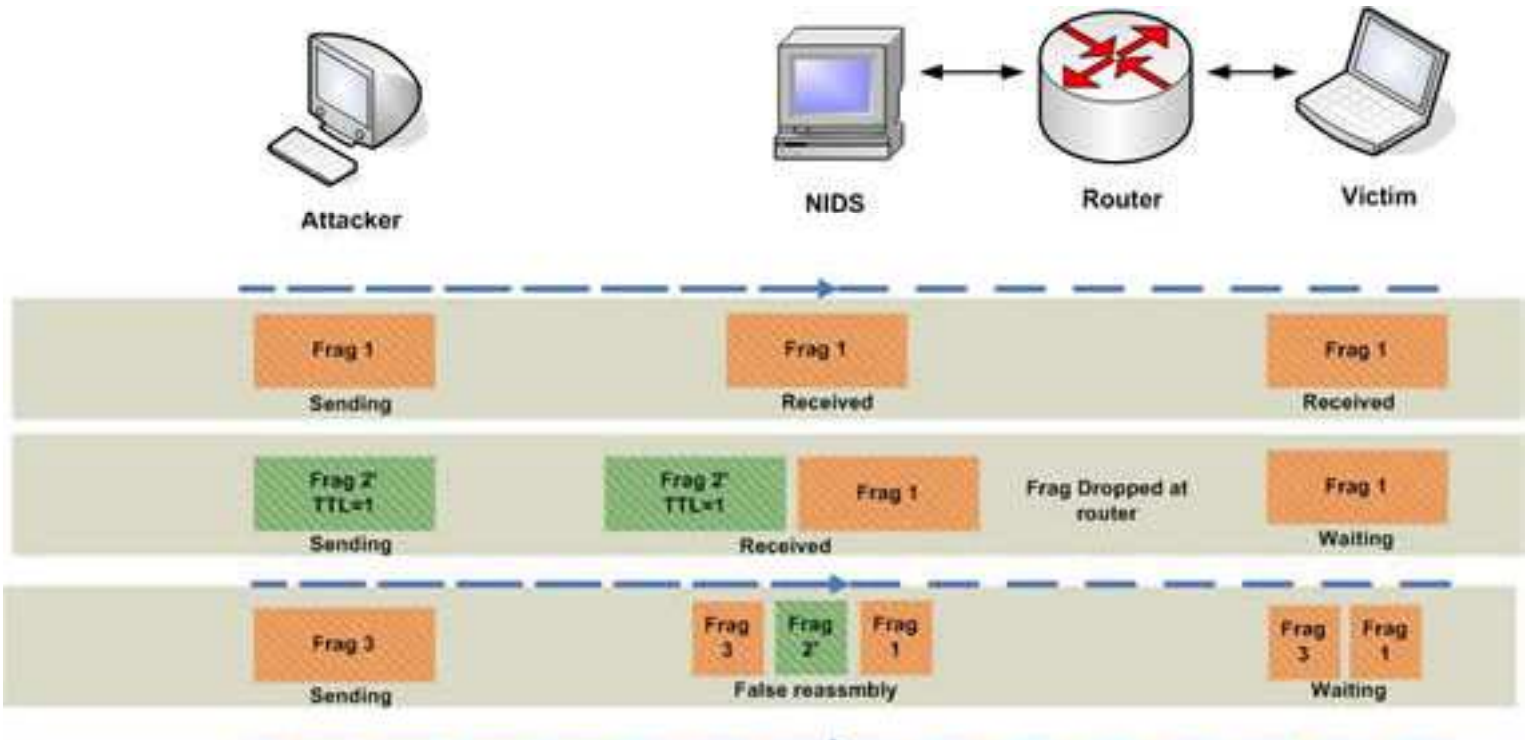
D. IP Fragmentation - Overlap



D. IP Fragmentation – Timeout



D. IP Fragmentation - TTL



E. Denial of Service (1)

- Possible to avoid detection by disabling or overwhelming the IDS
  - Severe as IDS systems are typically "fail-open" by default
  - Less sophisticated method of attack
  - Allow uninhibited access to network
- Possible to generate packets to flood IDS alert senor
  - Consume processing power
  - Hides real attack from personnel
  - Fill up disk space causing alerts not to be logged
    - Filesystem, databases, ticketing systems etc
- These types of packets are easily spoofed –
  - false negative conditions

E. Denial of Service (2)

- IPS can be fooled into invoking counter-measures against other systems due to FPs
  - Possible to spoof packets with false source address which trigger a modifications on firewalls blocking traffic
- Possible to crash IDS using bad packets or series of bad control messages
  - CORE-2003-0307: Snort Remote Integer Overflow

Defeat Evasion Techniques (1)

- IPv6 addresses fragmentation overlap in RFC 5722
- Traffic Normalisation
  - Takes obfuscated input and translates it into what the end-system will eventually see
  - Encoding in formats such as Unicode and UTF-8
  - Difficult to apply to polymorphic shellcode attacks
    - Trade off between time taken to process and effectively monitor live sessions

Defeat Evasion Techniques (2)

- Modification of packet header information
  - Modifies TTL values to large values – ensures packet arrives at destination
  - Helps defend against desynchronisation between IDS and end-systems
- Difficult to cover all evasion techniques!

Final Thoughts

- Variety of ways to evade IDS
- IDS technologies are still immature
- IDS/IPS technologies are not a full security solution
  - Don't be fooled into a false sense of security
  - Used to compliment existing security solutions
- Helps to gain knowledge of attacks against system to better harden network against future attacks

Questions?