pmcollins2
pcollins

1.)

Trace 1:

1.) amazon.com
baidu.com
bing.com

2.) "adventures in Stochastic Processes"
"来自星星的你"
"madison map"

Trace 2:

1.) Username: "shiningmoon"
Password: "public"

2.) FTP connections use two ports to send files a data and a control port. So two different port numbers need to be opened on the server and client. Active FTP open both ports by having the client connect to port 21 on the server (normally) from an arbitrary port for control. The server then responds by connecting to the arbitrary port on the client plus one for data. Usually any service trying to connect to a port from the outside network to a client will be blocked by a firewall unless the connection has already been established from the client to that service. Passive FTP connections solve this problem with Active FTP connections being firewalled by having the server respond with an open port number for the client to connect to for data. This way all client connections are established from the client connecting to open ports on the server and not the other way around.

3.) None of the FTP connections are Active

4.) All of the FTP connections are Passive

5.) "dragon.zip"
"ARP.java"
"L2Switch.java"
"phase1.html"

Trace 3:

1.) 192.168.0.100

2.) 74.125.225.46

3.) 192.168.0.1
10.131.180.1
96.34.20.20
96.34.17.95
96.34.17.112
96.34.17.77
96.34.2.4
96.34.0.7
96.34.0.9
96.34.3.9
96.34.152.30
209.85.254.120
209.85.250.28

4.) Internet Assigned Numbers Authority (IANA) (ie. private network addresses)
Charter Communications

Google Inc.

Trace 4:

1.) Username: "cs155@dummymail.com"
Password: "whitehat"

2.) 5 messages

3.) From: "joe <cs155@dummymail.com>"
To: "cs155@dummymail.com"
Subject: "foobar"
Date: "Fri, 23 Apr 2010 08:20:52 -0700"

From: "joe <cs155@dummymail.com>"
To: "cs155@dummymail.com"
Subject: "can you see this subject?"
Date: "Fri, 23 Apr 2010 08:23:25 -0700"

From: "hariny <harinym@stanford.edu>"
To: "cs155@dummymail.com"
Subject: "test message"
Date: "Fri, 23 Apr 2010 10:25:00 -0700"

From: "hariny <harinym@stanford.edu>"
To: "cs155@dummymail.com"
Subject: "geology rocks!"
Date: "Fri, 23 Apr 2010 08:22:28 -0700"

From: "hariny <harinym@stanford.edu>"
To: "cs155@dummymail.com"
Subject: "wassup"
Date: "Fri, 23 Apr 2010 08:21:50 -0700"

2.)

1.) With offline packet logging a large SYN DoS attack could fill the storage capacity of the IDS machine's disk and crash it.

The IP header struct is not IPv6 compatible and could either bypass the IDS system or possibly exploit it to crash the logger.

There is not verification of the IP or TCP headers which could exploited. Such as line 76 and line 80 the IP and TCP header lengths aren't checked to be valid which could be used to spoof the check on line 78 or the payload extraction on line 80 to bypass logging. An arbitrary offset in memory could be logged as the payload.

The way the report buffer is created on line 24 and written to by the payload with a memcpy on line 45 appears to be open to an integer overflow attack which with a buffer overflow could lead to arbitrary code execution crashing the IDS, or even maliciously logging packets for an adversary.

2.) Making the offline packet logging online would fix the issue of filling the machine's disk, or deleting old logs could possibly fix it.

Use a dual-stack header or implementation to handle both IPv4 and IPv6 packets.

Verify all inputs that are used to control the program including packet headers for IP and TCP.

3.)

See main.py

4.)

1.) Assuming we know the small business' UDP port for DNS we could forge a DNS response by guessing the valid Query ID from the victim's DNS server NS or A record request. The forged response could then poison the DNS cache by storing an arbitrary IP address for the domain or even the second level domain name server.

A successful DNS cache poisoning attack would enable the attacker to point any specific website visited by the victim to an arbitrary IP address if the attacker knew the website visited, or second level domain visited. This could be used download some malware drive by would install itself on the victim's machine, or the server at the arbitrary IP address just simply track and sniff the request through a man in the middle attack (MITM) before returning data from the legitimate site.

2.) An attacker could infer if the specific site was visited by employees by doing an A query for that site and differentiating the return time from a cache hit versus a cache miss. A cache hit (and thus a shorter return time) would imply the site had been visited in the last 15 minutes, whereas a cache miss would imply it hadn't been visited. The TTL limits the accuracy of the attack as multiple requests by multiple employees could be done within the 15 minutes and only one visit would be recorded. A shorted TTL would fix this, but make the use of the cache a moot point.