



FIREWALL-ANTIVIRUS

Response

Kraysis team initiated by doing analysis of the application and its underlying architecture. This included understanding its communication processes, its interaction with operating systems and also exploring deep into the communication protocols and their inner working.

After the exploration and domain understanding (through communication with the development team), the testing team started performing functional testing of the application. While this phase was performed to analyze the stability of application, it was also essential to gauge the response of this application under different combinations of system interactions. In order to assure the correct response, a sub-network was created to simulate the interaction of application not only within one network but also across different networks.

Some of the protocols tested over this application are,

Background

Our client, a world renowned antivirus development company required a thorough testing of their antivirus and firewall combination application, both at functional and usability level. Application was required to be tested on multiple platforms along with its communication with different networks as well as operating systems. Antivirus application needed to be tested for its live updates from all the operating systems.

Challenge

- An antivirus and firewall application needs to be tested on several protocols
- Ensure the smooth running of the application on any operating system
- To ensure that communication between multiple platforms is working under all scenarios.
- Meet the defined timelines which were short due to *time-to-market* nature of the product.
- Reproduce the similar level of bugs on the testing as well as development side due to variegated nature of environments



Table 1 Protocols Tested

PROTOCOL NAME	PORT RANGE
DHCP	68
SSH	22
SMTP	587
POP3	110
SMTSPS	465
HTTPS	443
NNTP	119
Telnet	23
Whois	43
NetBios	137-139
IRC	194
FTP	21
DNS	53
IMAP	143
Remote Desktop	3389
MSRPC	135
SMB	445

Table 2 Operating Systems Tested

Operating System	Version
Windows	Vista
	XP
	2000
	NT
	98
Linux	Red Hat
	Ubuntu

Analysis

Protocols in table 1 and operating systems in table 2 are just some of the ones which were tested and reported to our client as per the requirement. Testing these applications by making sub-networks proved to be an efficient technique which ensured that multiple protocols can be tested which would not have been possible otherwise. Data generated through the testing revealed the fact that some protocols which might have a lesser usability by the users were not properly performing their required functions. Also there were less number of problems when application was running its processes within a network, however if it had to operate between networks there were significant issues identified and reported. Many problems were found while application was tested to verify protocol communications between same operating systems. Number of problems significantly increased when the communication was tested between two different operating systems.

Since there was time-to-market pressure and this project required significant understanding of protocols, a team was also dedicated to perform

R&D. The effort of this team was not charged against the time spent however the knowledge acquired was shared with the client.

Conclusion

The project was conducted successfully through the combination of greater domain knowledge understanding, thorough R&D and by creating different test beds for application testing. This approach helped ensure that the client was able to launch its product in time with greater reliability and confidence.

United Kingdom

Kraxis Ltd Corporate Headquarters,
139 - Groveway, Dagenham, Essex,
RM8 3XL, United Kingdom.
Voice: +44 203 287 1186

Pakistan

Suite # 415, Al-Qadir Heights,
1 Babar Block, New Garden Town, 54700,
Pakistan.
Voice: +92 42 32404824-25