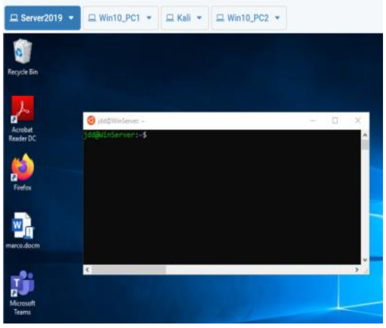
Active Countermeasures

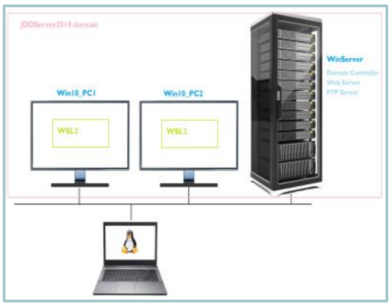
By: JingYing Dong, David Gore, Dom Tassadorn

Advisor: Sandeep Vankadari **Client:** Paul Bryant

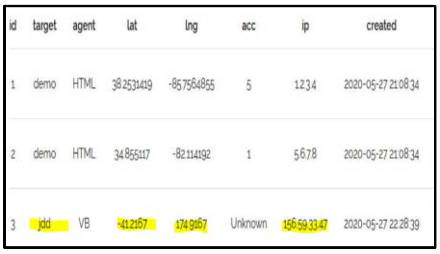


```
jdd@WinServer:~$
```

Windows Subsystem For Linux allows Active Countermeasures to run on Windows machine



The test environment topology



id	target	agent	lat	lng	acc	ip	created
1	demo	HTML	38.2531419	-85.7564855	5	1.2.3.4	2020-05-27 21:08:34
2	demo	HTML	34.855117	-82.114192	1	5.6.7.8	2020-05-27 21:08:34
3	jdd	VB	-41.2167	174.9167	Unknown	156.59.33.47	2020-05-27 22:28:39

An example of HoneyBadger finding geolocation of an attacker.

INTRODUCTION

The active countermeasures project was about taking standard defensive practices and taking them one step further. Additionally, we were tasked with doing so in a Windows environment, one of our main challenges. Our client was interested in the value of porting many Linux based tools that achieve active defence to a Windows environment to evaluate their worth in this situation.

The project team was also tasked with moving this environment into the 'could' for further evaluation. Doing so we can show the value of active countermeasures as an additional layer of security, and a means of identifying and attributing attackers.

DEVELOPMENT

The development process of our project followed an agile approach using a visual Kanban style. This was selected for our project as we are taking an iterative approach towards building our environments. This meant that we did a total of four builds, each improving and adding an aspect from the last. Having this Kanban style approach allowed the team to visualize what was going on at any point during the build. This was key in communication and keeping the team on task for the testing phases of the project.

The project team used a free software tool known as Trello to host the Kanban style approach. This was an effective tool and is recommended by the project team. The testing ranged from simply testing tools functionality, to implementing the testing in a live network. Then we moved on to adding Active Directory and a domain. Then we moved this whole environment to the 'could' for our final build.

The outcome of the project was testing manuals and user guides that the team developed for the environments. However, the primary deliverable was a 'findings report' which was a careful analysis of the tools testing and how they worked on the environment.

CONCLUSION

The project was a success in the eyes of the project team. While we encountered issues along the way, we followed the necessary processes to achieve a good outcome. We delivered the project findings report on time and completed the project with a good degree of competency. As a result of this, the project team benefited from this project, learning, and understanding more about the current cyber security climate and why it is the way it is.