Distributed PowerShell Load Generator (D-PLG)[1]

Paul Jordan, Donald Van Patten, Gilbert Peterson, and Andrew Sellers

Air Force Institute of Technology
Center for Cyberspace Research
Wright-Patterson Air Force Base, Ohio, U.S.A

SIMULTECH 2016
31 Jul - 2 Aug 2016

[1]The views expressed herein are solely those of the authors and do not reflect the official policy or position of the U.S. Air Force, the Department of Defense, or the U.S. Government.

Overview

Abstract

- Failure in cloud infrastructure is common occurrence
- Problem is often masked by the use of excessively redundant systems
- Machine learning techniques have been studied to predict failure [3]
- Unfortunately, this work has gone unused [2]

Solution?

- Framework introduced to solve problem called Adaptive Failure Prediction (AFP) Framework
- Load a service $\Rightarrow$ Inject faults $\Rightarrow$ Record failure
- How do we load a Microsoft Windows active directory domain?

Distributed PowersShell Load Generator (D-PLG)

- PowerShell script
- Remote execution
- Full-stack two-way dynamic authentication traffic
- Full-stack simulated web browsing
- Dynamic file sharing
- . . . and so much more!

Existing Tools

- Many software tools exist for generating network traffic
  Three major categories [1, 4]:
  - Application-Level
  - Flow-Level
  - Packet-Level
- Unfortunately, no existing tools generate full-stack dynamic traffic which we need to generate real authentication traffic to sufficiently load a domain controller

D-PLG

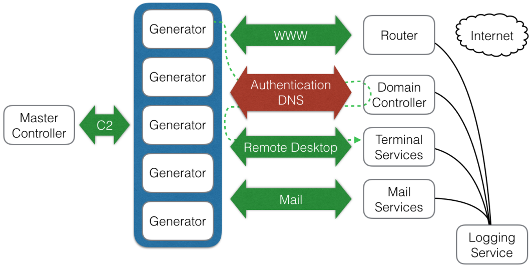

Figure: How each type of traffic that is generated is routed. Log events are offloaded to logging service for further analysis.

Web Browsing

- PowerShell cmdlet ‘Invoke-WebRequest’
- Returns full DOM object
- Allows us to simulate browsing

Remote Desktop Protocol

- Custom PowerShell cmdlet ‘Connect-Mstsc’ [2]
- Hidden window so we don’t interrupt users
- Makes connection, sleeps for a few seconds and then disconnects
- Plan to implement machine interaction

[2] https://gallery.technet.microsoft.com/scriptcenter/Connect-Mstsc-Open-RDP-2064b10b

Server Message Block (SMB) File Sharing

- PowerShell cmdlet ‘New-PSDrive’
- Connects and authenticates to remote file share
- Uploads 100 bytes of random ASCII data
- Deletes created file and disconnects

Future Modules

- PowerShell cmdlet ‘Send-MailMessage’
- PowerShell cmdlet ‘Out-Printer’

Methodology

Experiment

- Two hypotheses:
  1. We can sufficiently load the domain controller using our script
  2. We can use D-PLG without the end-user noticing
- Two tests: each five, five minute rounds of execution
- First test maximized traffic generated
- Second test balanced traffic generation with client utilization
- Capture all network traffic and performance/utilization metrics

Virtual Environment

Table: Hypervisor 1 configuration (sandbox/target).

Qty.	Role	Operating System	CPU / Mem.
1	DC	Win. Server 2008 R2	2 / 2 GB
1	Web	Win. Server 2008 R2	2 / 2 GB
5	Client	Win. 7	1 / 512 MB

Table: Hypervisor 2 configuration (controller).

Qty.	Role	Operating System	CPU / Mem.
1	RDP	Win. Server 2008 R2	1 / 4 GB
1	Log	Ubuntu 14.04 LTS	1 / 1 GB

Results

Can we sufficiently load the domain controller?

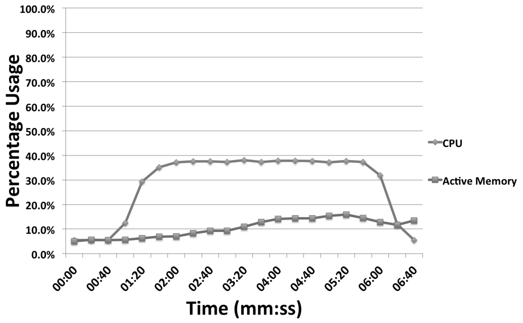

Figure: Domain controller CPU and memory utilization during the first test.

- Able to sufficiently load domain controller based on Microsoft's community recommendations (40% CPU utilization during peak use).

Packets Recieved by Domain Controller

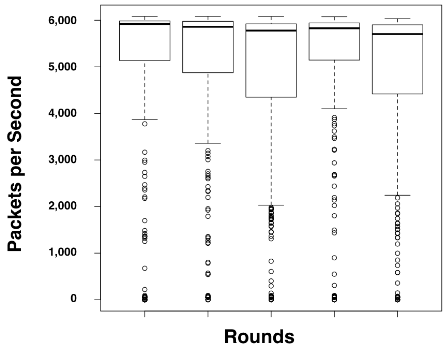

Figure: How many packets per second were sent or received by the domain controller across all five rounds of the first test. In each test, we captured approximately 1.8 million packets.

Can we use D-PLG without the end-user noticing?

- Not quite as successful as first
- Client machines were undersized compared to standard desktop computers in enterprise environment
- Result was that while they produced a sufficient amount of traffic, they would have been a little slow to use
- Solution: Use more powerful client machines, or use during idle down times

Results

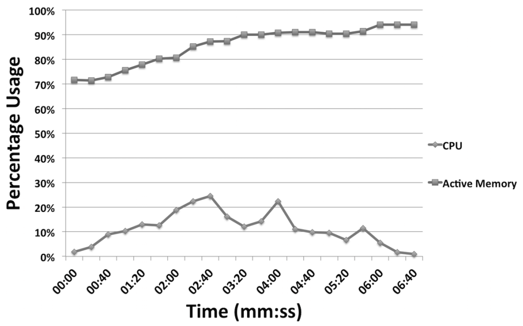

Figure: Client CPU and memory utilization during the second test.

Future Work

- Build-out tool
- Add new functionality like e-mail and printing support
- Give users more control over type of load generated
- Increase stochasticity to better simulate user behavior

Conclusion

- Based on the results of our first test, D-PLG will meet our needs to implement AFP against a domain controller
- We believe our results demonstrate D-PLG's applicability to other problems that require dynamic traffic between unbounded network components

Summary

Questions?

Acknowledgments

This research was sponsored by the U.S. National Security Agency, National Information Assurance Education and Training Program (Alice Shafer and Glenn Ellisonn, Program Managers).

A. Botta, A. Dainotti, and A. Pescapé.
A tool for the generation of realistic network workload for emerging networking scenarios.
Computer Networks, 56(15):3531–3547, 2012.

I. Irrera, M. Vieira, and J. Duraes.
Adaptive failure prediction for computer systems: A framework and a case study.
In *Proceedings of the 2015 IEEE 16th International Symposium on High Assurance Systems Engineering (HASE 2015)*, pages 142–149, 2015.

F. Salfner, M. Lenk, and M. Malek.
A survey of online failure prediction methods.
ACM Computing Surveys (CSUR), 42(3), 2010.

P. Zach, M. Pokorny, and A. Motycka.
Design of software network traffic generator.
Recent Advances in Circuits, Systems, Telecommunications and Control, pages 244–251, 2013.