NITTF Tech Talk – Trends in Insider Risk Quantification, Part 3

Dan Costa

Derrick Spooner

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM21-0743

From Our Last Tech Talk

Test your technical controls in a test environment, and not in production

To do this, you'll need:

- Virtual Environments, e.g., https://github.com/cmu-sei/crucible
- User Simulation, e.g., https://github.com/cmu-sei/GHOSTS

← Today's Focus

Use Cases to Consider

Operational

- Example: measure alert volume, validate alert accuracy

Coverage

- Example: measure percentage of indicators covered by monitoring system(s)

Latency

- Example: measure average time between activity and alert, alert throughput

Compliance

- Example: validate percentage of recommended / required (NIST SP 800-53, NITTF Minimum Standards) controls implemented

Testing and Validation

Candidate Controls

- Faulty indicators
- Faulty monitoring tools
- Legacy Systems

Tune existing policies

- Overly active alerts

Data Sources

- Exploratory Analysis

Security Testing

- Assume the identity of an insider
- Training opportunities

•Software Tools

- Commercial
- Open-Source

New Deployment Environments

- Yellow/Red/Green Networks
- Federated SIEM

Functional Requirements for Simulation

•Technical observables:	Enterprise and personal email
	Multi-browser web usage
	File operations (create/read/update/delete)
	Using Cloud storage software
	Removable Media
	Virtual Private Networks
	Printers/Scanners
	Remote Administration
•Behavioral observables:	Organizational structure
	Privileged and Non-Privileged users
	Personnel Events (performance reviews, complaints, reprimands, travel)
	Realistic work schedule

https://insights.sei.cmu.edu/blog/functional-requirements-for-insider-threat-tool-testing

Reference Approach

Configurable

- Adjust security controls, vendors, user behavior

Decoupled

- Data generation should not rely on specific tools or outputs

Controlled/Repeatable

- Follow a sound scientific method

Accessible

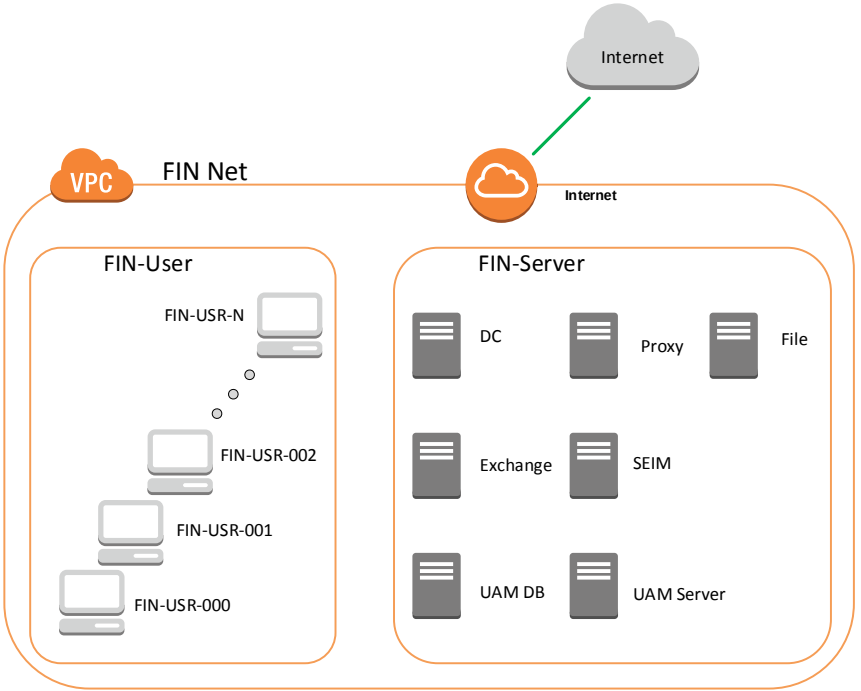
- Stakeholders, vendors, or third-parties

Comprehensive

- Technical and behavioral actions and indicators

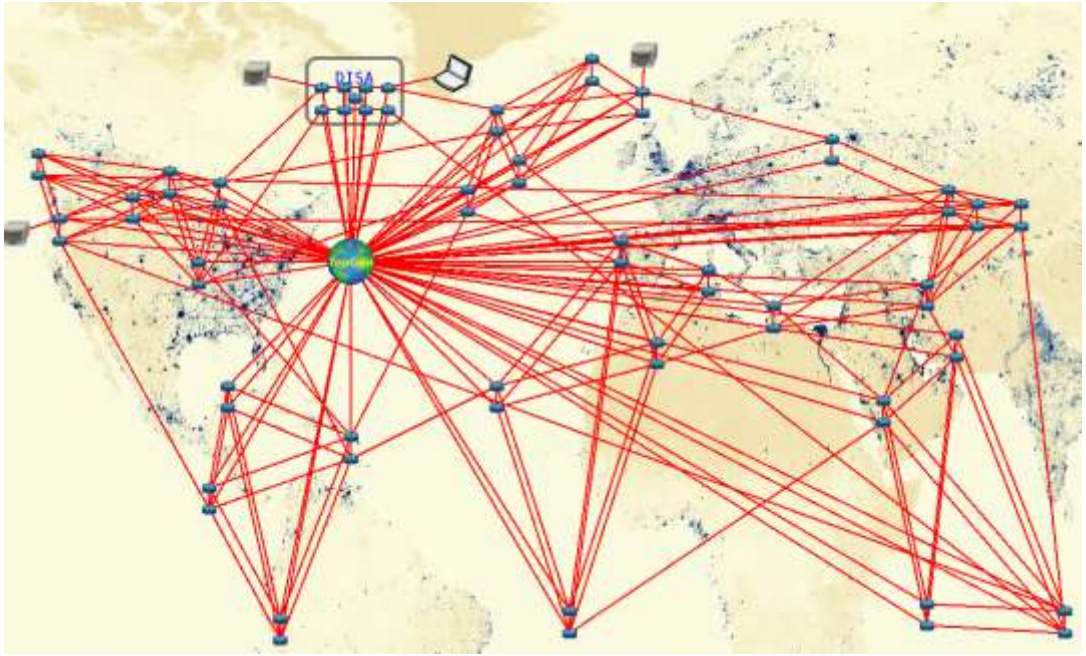
Reference Architecture

- Virtualized (VM) on-prem environment
  - 2x: 256GB, Dual 8 Core Xenon
  - 2TB storage
    - Linked VM Clones
- Amazon Web Services Proof of Concept
  - GovCloud – ITAR Compliance
  - 10 Service Systems (t3.nano – xlarge)
  - 5-10 Users (t3.small)
  - 4-8 hours/day
  - <$500/month



Open-Source/Freeware Infrastructure Tools

- Greybox/TopGen (Topology Generator) https://github.com/cmu-sei/topgen / https://github.com/cmu-sei/greybox
  - Isolated (Air-Gapped) Internet-in-a-box
- iRedMail https://www.iredmail.org/
  - Webmail simulation
- Owncloud Server https://owncloud.com/
  - Cloud storage simulation



Open-Source User Simulation Tools

- GHOSTS ((G)eneral HOSTS) Framework https://github.com/cmu-sei/GHOSTS
  - Non-player character (NPC) automation
  - Baseline traffic
- GHOSTS-Animator https://github.com/cmu-sei/GHOSTS-ANIMATOR
  - User detail generator
  - Addresses, Rank, Work Location, Potential Risk Indicators

https://insights.sei.cmu.edu/blog/generating-realistic-non-player-characters-for-training-cyberteams/

Infrastructure as Code (IaC)

```
module "dc" {
  source     = "./modules/dc"
  dc_instance_type  = var.dc_instance_type
  subnets    = var.subnets
  security_groups = var.security_groups
}
```

```
resource "aws_instance" "this" {
  ami                    = data.aws_ami.this.id
  instance_type          = var.dc_instance_type
  iam_instance_profile   = "FIN SSM Role"
  private_ip             = "10.0.40.20"
  subnet_id              = var.subnets
  vpc_security_group_ids = var.security_groups

  tags = {
    Terraform   = "true"
    Name        = "FIN-DC"
    Environment = "FIN Test"
  }
}
```

Open-Source Tools:

- Terraform https://www.terraform.io/
  - Infrastructure as Code
  - Multi-platform (Windows/Linux/OSX)
  - Multi-cloud (AWS/VMware/Azure)
- Packer https://www.packer.io/
  - Virtual machine image builder
- Ansible https://www.ansible.com/
  - System provisioning and configuration

What Next?

- Faster and more realistic baseline data generation
  - How ‘real’ is ‘real enough’ for certain test objectives?
- Variety and modularity of testing environment
- Automated test and evaluation

Q&A / Open Discussion

Presenter Contact Information

Dan Costa, CISSP, PSEM

Technical Manager, CERT National Insider Threat Center

dlcosta@sei.cmu.edu

Derrick Spooner, CISSP

Cybersecurity Engineer, CERT National Insider Threat Center

dspooner@sei.cmu.edu