Interested in learning more about security?

SANS Institute InfoSec Reading Room

Simulating Cyber Operations: A Cyber Security Training Framework

While there are many security competitions and training platforms used to simulate the electromagnetic communication spectrum known as Cyber, they vary widely in effectiveness, assessment capabilities and flexibility. In addition, most are closed and proprietary in nature. What is needed is a publicly adopted Cyber Operations simulation standard to support training, assessment and technique development of operators within the electromagnetic communications spectrum. This paper proposes an innovative way to model Cybe...

Simulating Cyber Operations: A Cyber Security Training Framework

GIAC (GSEC) Gold Certification

Author: Bryan K. Fite, bfite@meshco.com
Advisor: Barbara L. Filkins

Accepted: February 11, 2014

Abstract

While there are many security competitions and training platforms used to simulate the electromagnetic communication spectrum known as *Cyber*, they vary widely in effectiveness, assessment capabilities and flexibility. In addition, most are closed and proprietary in nature. What is needed is a publicly adopted *Cyber Operations* simulation standard to support training, assessment and technique development of operators within the electromagnetic communications spectrum. This paper proposes an innovative way to model *Cyber Operations* by representing the core simulation elements as *Objects* and describing their interactions via a *Scenario Definition Language (SDL)*, which dictates the rules governing *Object* interactions. It further describes an approach used to create purpose-built simulations, defines fundamental object types, presents a lexicon and shows how gaming can be used to support effective cyber operations training and assessment.

Introduction

The current shortage (Finkle & Randewich, 2012) of trained and experienced *Cyber Operations Specialist* [1]coupled with the increasing threat (Sophos, 2013) posed by targeted attacks (Verizon, 2013) suggest more effective training methods must be considered. Specifically, there is a need for a publicly adopted *Cyber Operations* simulation standard to support rapid training, assessment and tool development. Such an approach can address the personnel shortage and ensure organizations can adequately respond to evolving threats.

Simulations can be very effective and powerful tools. While there are many *Cyber* security competitions and training platforms (see Appendix A: Simulations), they vary widely in effectiveness, assessment capabilities and flexibility. In addition, most are closed and proprietary in nature. Effective assessment requires standards. Standards must be declared, shared and adopted to provide practical utility. Simulations can also reinforce and enhance real-world operational capabilities. The practical benefits of adopting such standards include:

- More relevant and effective training programs
- Standardized skills assessment and accreditation criteria
- Simulation portability and scalability assurance
- The creation of campaign planning and modeling tools

This paper proposes an innovative way to model *Cyber Operations* via simulations, abstracting the fundamental elements as *Objects* and describing their interaction via a *Simulation Definition Language* (*SDL*). In addition, it describes the approach, declares fundamental object types, presents a Lexicon (see Appendix B: Lexicon) and provides examples of practical application. These elements create the functional foundation for a standard approach to executing *Cyber Operations* simulations.

[1] A trained and experienced subject matter expert operating in the electromagnetic communications spectrum.

Bryan K. Fite;bfite@meshco.com

1. Models and Simulations

Models are the physical, conceptual or mathematical representation of a system, process or organization with defined rules governing their interactions. Humans have used models to describe the world around them for centuries. Art and science are rich with examples. Ancient cave paintings depicting organized hunting parties, Michael Angelo's anatomical treatise 'Vitruvian Man' (Stanford University, 2002) and modern efforts to map the Human Genome (National Human Genome Research Institute, 2003) are just three examples.

Models provide the rules that govern simulations. A simulation is simply an artificial construct designed to emulate a real world process or system over time. Simulations have been used as effective training and assessment tools throughout history, such as wooden practice swords, mechanical horses (Nilsson, 2010) and flight simulators (Flight Simulator History, 2008).

The utility and appeal of using simulations are especially obvious within a military context. Much better to get bruised with a wooden sword during practice than lose a limb in real combat through lack of skill. Learning to control a horse while wielding a weapon requires techniques best acquired without spooking your mount. Given the skill and cost required to pilot a modern military aircraft, it is much better to train safely in a cost-effective flight simulator than risk the loss of life and multi-million dollar equipment to an untested operator. Simply put, the price of failure is high and the rate of failure will be high until the required skills are attained.

The hallmark of a good candidate for simulation is an activity that is complex, dangerous and/or expensive. The capital and operational costs associated with specialized equipment prohibits their use by unqualified operators. Learning complex skills requires repetition. Training large numbers of people simultaneously and consistently is not a trivial effort.

Controlling the level of risk reduces the consequences of making mistakes, which is often part of the learning process. Mistakes provide opportunities to "rewind" the simulation step-by-step and correct failed simulation responses thus reinforcing the learning experience. The ability to step through all the possible simulation responses

provides opportunities to find the optimum response path. Simulations that modify variables in ways that rarely exist in the real world or are theoretical provide unique opportunities to model high impact *Scenarios* [2]and support "what-if" exercises.

2. Approach

In order to create effective and portable simulations designed to teach and assess *Cyber Operations* capabilities, we must have a way to describe the simulated environments. This approach uses functional building blocks to describe simulated environments. These environments can be implemented and deployed across many diverse physical and virtual simulation platforms. This allows for rapid creation of simulations on one platform, which can easily be ported and ran on alternative platforms.

More specifically, we must define the relevant variables required to model the simulated environment and the rules that govern how elements interact with each other in that particular simulated environment or *Universe*[3]. Depending on the *Objectives* [4]and *Constraints* [5]of the simulation, the rules of a specific *Universe* may or may not parallel those of the real world.

The functional building blocks of a simulation can be expressed as *Objects*. *Objects* have specific characteristics known as *Attributes*. Their interactions with each other are governed by the rules of their respective *Universe* and can be described using the Scenario *Definition Language* (SDL)

Objects and Attributes

In order to facilitate multiple simulation integration, simulation portability and like-for-like-assessment across various simulation platforms, a common way to describe all relevant elements within a particular simulated environment is required.

---

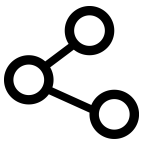
[2] A logical construct providing the context and all other required simulation elements in a human friendly format.
[3] Refers to a simulated environment that is made up of a combination of primitive objects and governed by the rules that dictate how those primitive objects interaction with each other. The rules governing a particulate simulation may or may not align with the rules governing the "real world"; physics, chemistry, biology and the like.
[4] A primitive object that defines the relative goals of a simulation.
[5] A primitive object designed to shape a simulation by limiting the actor's range of motion and sphere of influence.

Within a given simulation, *Objects* are a collection of predetermined *Attributes* used to describe a specific simulation element. Here an *Object Type* determines required and optional *Attributes*. *Attribute* characteristics dictate how *Objects* interact with each other inside a simulated environment.

There are nine core *Object Types* defined in this paper, known as *Primitives*, which can be used to describe any simulation within the electromagnetic communications spectrum also known as the *Cyber Domain*[6]. All *Primitives* must be supported by a particular simulation environment to be considered compliant. This ensures integration, portability and standard assessment capabilities across various simulated environments.

The nine *Primitives* are:

Node: Any Open System Interconnection (OSI) Layers 1 to 7 (International Organization for Standardization, 1996) connected element



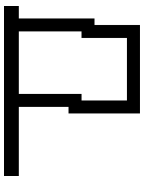
Network: The communication path or paths between nodes, typically OSI Layers 1 to 3 (International Organization for Standardization, 1996)

Software: An operating system, utility, application or service

Artifact: A file (text, audio, graphic or video) or credentials (account, username, password or key material)

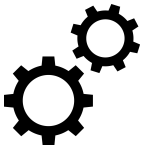
Constraint: Shapes a simulation by limiting the actor's range of motion and sphere of influence

[6] The total potential sphere of influence afforded an actor or group of actors in the electromagnetic communications spectrum.



Objective: The relative goals of a simulation

Actor: A human participant in an active simulation



Process: The workflow associated with a pre-defined simulation element interaction

Message: Communicates information, data or instructions between simulation elements

Objects can be expressed as a hieratical collection of required and optional *Attributes* in a standardized notation as shown in the following example:

Object Type: *Node*

Required Attributes: *Name (N), Host / Gateway Flag (H/G), Operating System (OS), Interface Address(es) (IA), Routing Table (RT), ARP Table (AT), Listening Ports (LP)*

Optional Attributes: *Accounts, Applications, Artifact and Services*

Adding a unique identifier and populating the *Object Attributes* yields a fully realized *Primitive*:

> The **Object ID (OID)** is a unique number, in this case 11001. The **Object Type (OT)** is a *Node* (N) named hackme running in **Host** mode. hackme is running Windows XP SP2 **Operating System (OS)** with the TCP/**IP Address** (IA) of 192.168.0.10 and a **Routing Table (RT)** which includes a default gateway of 192.168.0.1/24. There are two ARP entries in the **ARP Table (AT)** one assigned to hackme and the other to the default gateway. There are two **Listening Ports (LP)** which represent http (port 80) and https (port 443) services. An optional **Account** attribute is included – the username Administrator and its password.

We can further transform this example into a more structured and compressed notation as shown below:

> **OID** *(11001):* **OT** *(N):***N***(hackme):***H/G***(H):OS***(WINXPSP2):***IA***(192.168.0.10):* **RT***(0.0.0.0, 192.168.0.1/24):***AT** *(192.168.0.1:88-1f-a1-2c-00-7e,192.168.0.10:88-1f-a1-2c-ff):***LP** *(TCP:80,443): AC(Administrator:password)*

Please see Appendix C: Object Type Definitions for a complete list of all *Object Types*

It should be noted that some *Primitives* can be used to describe different aspects of the same element. Additionally, some *Primitives* could be used to populate variables in another *Artifact*. Example: *Software* could represent an executable file, which could also be declared as an *Artifact* and referenced in a *Message*.

Simulation Definition Language

By combining various objects together and providing structure, syntax and context, we can create an abstraction method for describing simulations. This abstraction method uses a standard *Simulation Definition Language* (*SDL*) to describe simulation elements and the rules that govern them. Using a standard *SDL* ensures consistent assessment capabilities across platforms and makes simulations available to more diverse populations, regardless of budget, access to technology or experience-level.

All simulations begin with a narrative or story in the form of a *Scenario*. A *Scenario* is a logical construct, which provides the context and other required simulation elements in a human friendly format, as in the example *Scenario* below:

> *"Your mission is to identify your adversary's security posture by enumerating the attack surface represented by their external network address 10.0.10.0/24. You must submit your findings by 17:00 ET today (1 hour from now). "*

The above *Scenario* provides the needed information to start creating an effective simulation by decomposing the *Scenario* and then populating the required *Objects* for the simulation:

Message Object: The entire *Scenario* statement above represents the actual message.

Node Objects: Attacker Platform and Defender platform(s), which are used by the Actors to communicate. These represent each Actor's Platform. (Note: an *Actor Platform* is also an Asset).

Network(s): The communication path(s) between *Actor Platforms* and the objective network's referenced attack surface 10.100.0.0/24.

Actors: In this scenario we can infer there is one Attacker (you) and at least one Defender (target).

Constraint: There is a 1 hour time limit.

Objective: Identify defender's attack surface, which would be the defender's assets with exposed IP addresses and TPC/IP/UDP based services.

Simulations can be as simple or complex as needed. However, for a simulation to have any practical utility as a training or assessment tool, a simulation must conform with several basic structural principles:

- Contains at least one *Message*
- Contains at least one *Node*

- Contains at least one *Network*
- Contains at least two *Actor*s
- Contains at least one *Declared Objective*
- Contains at least one *Constraint*

3. Using Games to Train and Assess

There exists a plethora of *Cyber* competitions and training platforms created for various purposes. Perhaps the most interesting are those implemented in the form of games. Games have rules, competitors and promised rewards. These three items parallel the elements that are fundamental to any *Cyber Operation* simulation: *Constraints* (rules), *Actors* (competitors) and *Objectives* (promised rewards).

Now that we have a way to describe simulated environments, we can bring the various elements together to create training and assessment simulations that take the form of games. This is sometimes referred to as *Edutainment*, a term used to describe the fusion of an educational experience and an entertainment experience. People can learn more effectively in immersive environments that are fun or otherwise enjoyable.

A well-known example of *Edutainment* is Packetwars™. Packetwars™ is a cyber-sport which simulates offensive and defensive computing scenarios. " PACKETWARS™ *is a Sport like nothing you have ever experienced! Games known as "BATTLES" pit individual players and teams against each other in a race to achieve defined objectives.*

The rules of engagement are simple:

- Illegal activity of any kind is prohibited

- Protect yourself at all times

- Battles are designed to be of a low, medium or high difficulty level based on the battle objectives and battle duration

- Primary, Secondary and Tertiary objectives are defined and assigned points based on difficulty

- Battles have time limits and other defined constraints

Bryan K. Fite;bfite@meshco.com

- Constraints are sometimes known to the combatants and other times are not

- Anything that is not expressly prohibited is allowed

- Points are awarded for FLARE" (Meshco Incorporated, 2009 p.how-to-play)

Players can take the assessment, that is play the game, by interacting with an implementation of a *SDL*-defined simulation in the form of a game. Remember *SDL*-compliant games are platform independent as the *SDL* creates an abstraction layer between the object primitives and the actual simulation platform. This means the games are transportable across platforms, whether dedicated, virtualized or *Cloud*[7]-based.

A simulation published on the Packetwars™ platform could be implemented on an alternative platform, assuming both platforms were *SDL*-compliant. This allows for consistent assessment of the *Actors*, called combatants in Packetwars™, across platforms, organizations, implementations and other variations, logical, virtual or physical.

Objectives and Constraints

Using the Packetwars™ example above, creating a Battle *Scenario* requires defining three *Objectives* (primary, secondary and tertiary) and setting a difficulty level. The difficulty level is dictated by the number and type of *Constraints.* A reference to duration suggests a primary *Constraint* is time. Time, in fact, is practically a universal *Constraint* and must always be considered. A short time allocated to a Battle, would be relatively more difficult than if a longer period of time were allocated to a Battle.

Objectives and Constraints are closely related but very different. Objectives are the relative goals associated with a simulation. *Constraints* are designed to shape a simulation by limiting an *Actor's* range of motion and sphere of influence. Consider how time is related to the *Objective* in the above example. An *Actor* must complete three *Objectives* in a set period of time. Time is therefore part of the *Objective* (achieve *Objectives* before time expires) and is also a *Constraint*.

Objectives come in three classes; attacker specific, defender specific and assessment specific. Attacker and defender *Objectives* are simply the goals assigned directly to the *Actors* within the simulation and communicated via a *Message*.

[7] A common term for a shared tenant computing environment.

Assessment *Objectives* can be aligned with attacker and defender *Objectives* but usually focus on measuring one or more specific aspects of that *Actor's* interaction with the simulation. Example: How much time did the *Actor* take to achieve the simulation *Objective*? Sometimes the interaction of interest is directly related to a *Constraint*, since *Constraints* by definition are a limiting factor. For assessment purposes, we could rank *Actors* by who successfully completed all *Objectives* in the shortest amount of time.

Constraints, as the name implies, are *Object Types,* which limit the flexibility of the actors and otherwise influence the "physics" of a particular "*Universe*". Allocated time or time limits, available bandwidth, toolsets (availability and effectiveness) and number of actors (attackers versus defenders) are typical simulation *Constraints*.

Objectives and *Constraints* are considered to be declared when they are communicated to the actors. For some assessment purposes, *Objectives* and *Constraints* are not exposed to the Actors. Example: A simulation might declare an *Objective* to retrieve an *Artifact* but not declare the *Objective* by file name as identifying the file name is part of the assessment challenge.

Using Game Spaces

Game Space refers to the demarcation defining the simulation "*Universe*" and contains the *Objects* that must directly interact as part of the simulation. The *Game Space* is accessible by external *Objects* such as *Actors* and *Actor Platforms* and can communicate with other simulated environments via private local area networks , wide area networks, public networks like the Internet, wireless networks, mesh networks and overlay networks like Virtual Private Networks (VPN). Think of the *Game Space* as the field of play in a sport or a Chess board in Chess. It is where most of the visible action takes place and the results of plays and moves manifest themselves.

Games can be built to leverage existing deployed resources, achieve simulation objectives and facilitate assessment requirements. Remotely deployed federated *Game Spaces* could use resources available in Deter Labs or EDURange (EDURange Project, 2013) environments and on dedicated equipment either in the *Cloud* or local network. This extensibility provides scalability, diversity and allows for extending the *Game Space* even with a small physical footprint. Additionally, it allows proprietary and critical

infrastructure systems to be shared without exposing intellectual property or production systems to risk.

Because of the nature of *Cyber Operations* simulations, if the *Game Space* is connected via the Internet or other non-isolated environments, care should be taken to contain all simulation activity within the *Game Space*.

Telemetry, Visualization and Analytics

The elements within the *Game Space* can provide valuable information in the form of *Telemetry*. *Telemetry* is data provided by simulation elements that can be used to provide visibility into significant simulation events. This information can be used within the simulation for managing the simulation. *Telemetry* data is collected from within the simulation and from other supporting platforms. *Telemetry* data can come from many sources, including; logs, alerts, traffic flows and performance monitoring tools.

Visualization [8]and *Analytics* [9]are very powerful tools and can add a new dimension to simulations, making them more effective. However, it's important to consider the different perspectives. There are four perspectives to consider:

- Attacker Perspective – The view the offensive actor has or should have.
- Defender Perspective – The view the defensive actor has or should have.
- Simulation Facilitator Perspective – Has visibility to all available *Telemetry* data.
- Scoreboard Perspective – The view available to all Actors, Simulation Facilitators and external observers.

To create the appropriate view and align with the assessment *Objectives Telemetry data* should be filtered and only shared with the appropriate entities at the appropriate time. This is called *Perspective Filtering*.

As a general rule, it is better to collect too much data then not enough. Once the data is collected, collated and available, numerous transformations are possible

[8] The transformation of simulation telemetry into a meaningful graphic representation.
[9] A term for the collection and review of data for meaningful and patterns.

(*Analytics*) and can take the form of automated, manual or ad hoc reporting. Reporting data can be used to support assessment, attestation and scoring.

Dedicated, Virtual and Cloud Platforms

SDL implementations can be deployed on dedicated hardware and software platforms, in private virtualized environments, in shared tenant *Cloud* environments or integrated with other simulated environments. Simulation implementations based on a standard *SDL* approach abstract the *Primitives* from the actual platform used to host the simulation. This allows for easy migration from one type of platform to another and easy integration between simulation environments implemented on different platforms.

Figure 1 depicts a Dedicated Physical Simulation Environment. All of the simulation *Objective Assets* are physical computers running a single dedicated operating system connected to a shared OSI Layer 1/2/3 (International Organization for Standardization, 1996) network sometimes referred to as the *Game Space*. Other simulation elements are represented, including *Actor Platform*, *Networks*, *Telemetry Platform* and *Simulation Management Platform*.

Bryan K. Fite;bfite@meshco.com

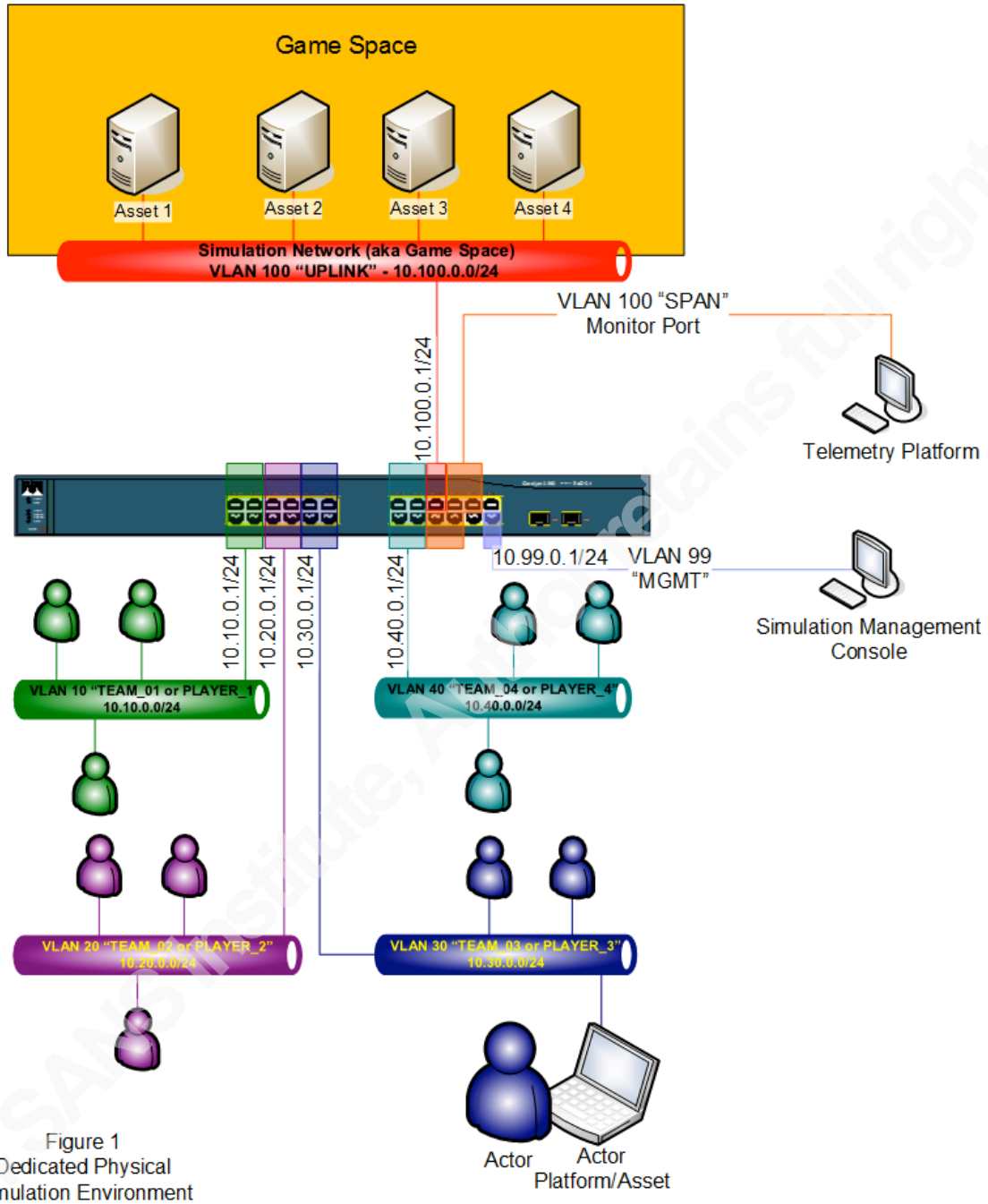


Figure 1
Dedicated Physical
Simulation Environment

Figure 2 shows how this changes when a Private Virtualized Simulation Environment is used to host the *Game Space*. All *Objective Assets* are part of the same OSI Layer 2/3 network and share the same physical hardware running virtualized

Bryan K. Fite;bfite@meshco.com

operating systems on a shared hypervisor.

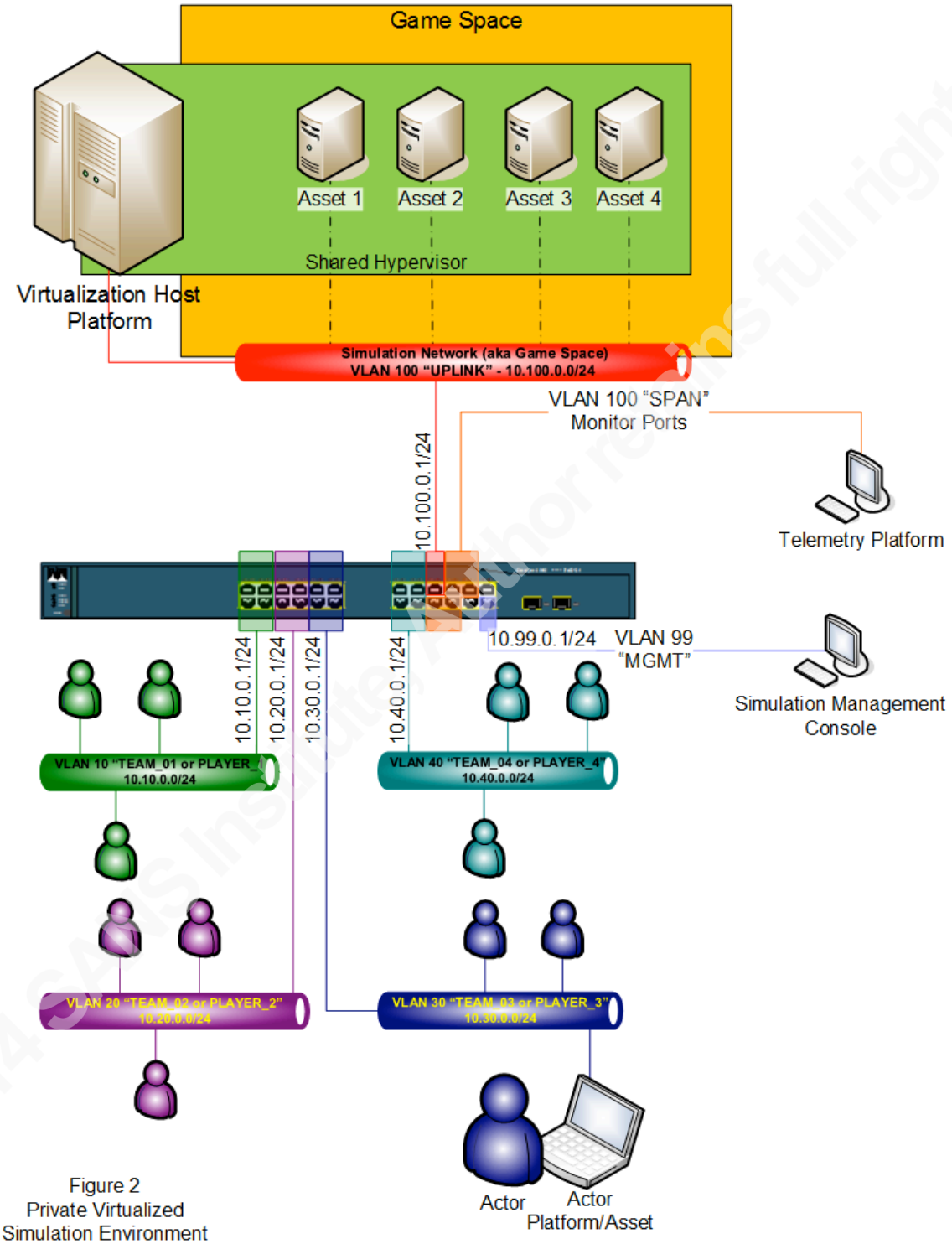


Figure 2
Private Virtualized
Simulation Environment

Bryan K. Fite;bfite@meshco.com

There are many advantages to this configuration over its physical counterpart as it requires fewer physical components, makes the platform more portable and masks many of the underlying complexities associated with incompatible and diverse hardware.

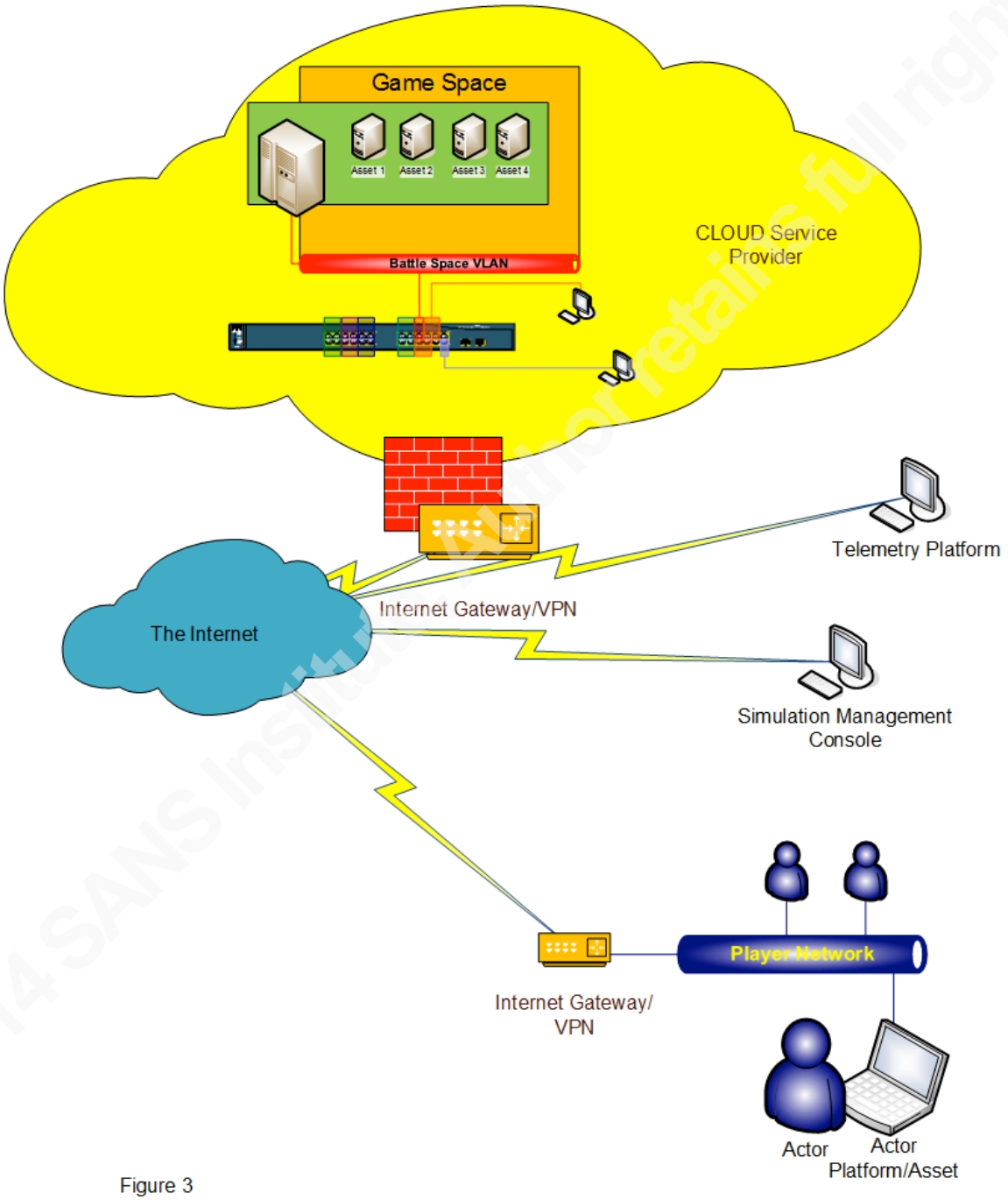


Figure 3
Cloud Based
Simulation Environment

Bryan K. Fite;bfite@meshco.com

Figure 3 shows Cloud Based Simulation Environment which is a virtualized platform residing inside of a *Cloud Service Provider* (*CSP*) environment, also known as a shared tenant environment. In this configuration, all *Objective Assets*, OSI Layer 2/3 networking, *Telemetry* feeds and simulation management interfaces reside within the *CSP's* environment. However, additional simulation management and *Telemetry* interfaces can be located outside of the *CSP environment*. The simulation participants can connect to the *Game Space* via the Internet, VPN or *CSP's* domain.

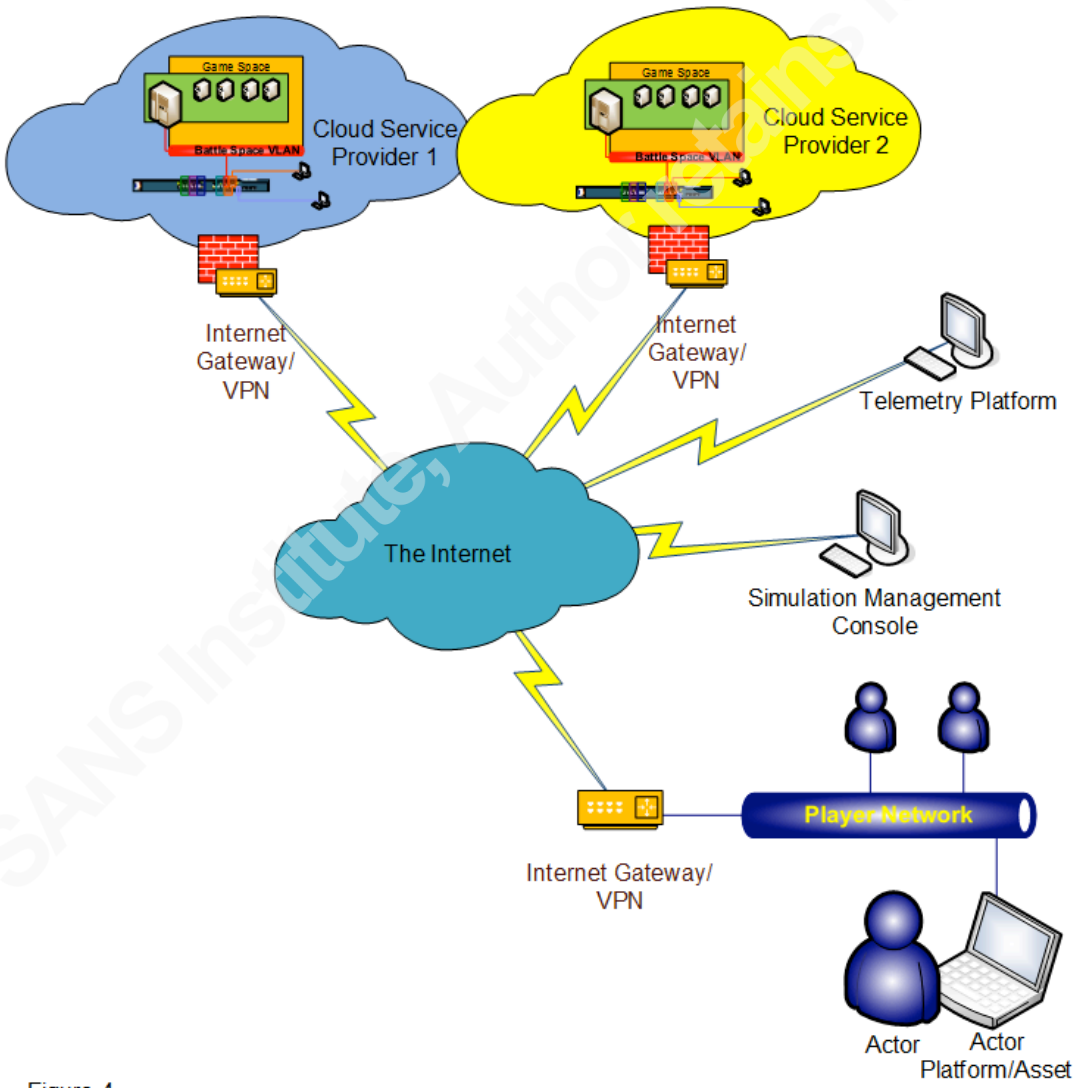


Figure 4
Federated Cloud Based Simulation Environment

Figure 4 Shows a Federated Cloud Based Simulation Environment and depicts two *Cloud*-based simulation environments that are integrated across the Internet via OSI

Layer 3 protocols. This allows the simulation to span physical and virtual domains but appear as a single interconnected environment. The *Telemetry Platform* and *Simulation Management Console* can exist inside or outside the *CSP environments*.

There are many factors that go into choosing the most appropriate simulation platform. However, cost, performance, access and agility (ease of implementation and change) are the primary considerations. For example, a Dedicated Physical Simulation Environment implementation might provide the performance and agility desired but does not easily accommodate remote users. Therefore, it might be more advantageous to use a *CSP* platform instead, which would allow simulation participants to access the *Game Space* from anywhere on the Internet.

For a platform to be considered *SDL* compliant it must be able to accommodate ALL required *Primitives* and support the relevant *Object Attributes*. If the platform cannot accommodate the relevant *Object Attributes* it is not a candidate for that specific simulation scenario.

Assessment and Scoring

The flexibility of this approach supports a myriad of different configurations, which in turn allow for assessment of many characteristics. It is important to define the specific assessment objective and design the simulation accordingly.

Baselining the *Game Space* is an important part of simulation preparation, especially if the simulation requires forensic attestation to validate the assessment objective . Baselining involves but is not limited to; identifying the known *Attack Surface* and *Attack Vectors*, creating checksums, maintaining change logs and creating versioned *Node* images.

Simulations can use *Fault Injection* and *Vulnerability Injection* to modify the known Attack Surface. This can aid in assessment scoring by assigning points to *Declared Objectives*. Achieving *Declared Objectives,* allow the *Actor* to score the associated points.

Bryan K. Fite;bfite@meshco.com

Automated or manual scoring systems can be used. However, to preserve the integrity of the scoring system the maximum possible score and how scoring elements will be verified must be determined prior to the start of a simulation.

4. Practical Application Use Cases

The flexible and modular nature of the *SDL* approach makes it easy to create purpose-built, scalable simulations for immersive training, standardized assessment and operational modeling.

Use Case 1

"EDURange is a National Science Foundation funded project with the aim of building cloud-based interactive security exercises." (EDURange Project, 2013 p.edurange). As part of the EDURange project, there was a need to create simulation environments in a structured and automated way.

The project team adopted the *SDL* approach, creating *Scenarios* and defining the *Objects* in terms of the scenario *Primitives.* They further transformed the *SDL* scenarios into configuration files written in Yet Another Markup Language (YAML) and which can easily be parsed. The YAML configuration files were then used to generate dynamic simulation environments within a *Cloud Service Provider* environment.

Each simulation can have the same *Objectives* and *Constraints* but with variable *Object* attributes. The implementation can generate large numbers of "unique" simulations but with a consistent training and assessment capability, allowing scalability and re-use for continuing assessment.

Use Case 2

Information Operations (*IO*), also known as *Cyber Operations*, encompasses the entire spectrum of Offensive and Defensive computing capabilities. "To succeed, it is necessary for US forces to gain and maintain information superiority." (US Chairman of the Joint Chiefs of Staff, 2006 p.22) The US Joint Forces doctrine (US Chairman of the Joint Chiefs of Staff, 2006) identifies 11 capabilities that need to be developed and perfected in order for this to happen. The *SDL* approach is perfectly aligned with that

mission and provides a platform for rapidly developing these 11 capabilities: Destroy, Disrupt, Degrade, Deny, Deceive, Exploit, Influence, Protect, Detect, Restore and Respond.

The development of skills, techniques, tactics and strategies associated with effective *Cyber Operations* requires the ability to test said capabilities. Using simulations to model complex scenarios without putting real assets at risk is a valuable practice. Once tested and proven in the simulation, the capability can be confidently promoted to field operations.

Simulations can also produce predictive adversarial analysis, which can be used to develop effective counter measures. The speed and flexibility afforded *SDL* simulations can aid in training and assessing large numbers of warfighters. The talent pool can be ranked using assessment scoring, which can identify superior candidates as well as areas of improvement. Using the *SDL* approach, organizations can share assessment rankings without exposing sensitive internal operational details. This has special significance within the context of the Government, Academia and Corporate ecosystems charged with protecting the world's critical infrastructure.

5. Summary

A reasonable approach to the current shortage of trained and experienced *Cyber Operations Specialists* is to use simulations to foster training, assessment and tool development though a standardized approach.

Simulations, security competitions and training platforms should adopt an *SDL* approach to improve simulation effectiveness, drive assessment standardization and support transportability. Specific benefits of the *SDL* approach include but are not limited to: making simulations more engaging and effective training tools, standardizing the assessment criteria across platforms, automating simulation creation and developing advanced adversarial modeling capabilities to keep pace with evolving threats.

The practice of using simulations to support training and assessment for *Cyber* security purposes warrants consideration and public discourse, given the number of practical use cases that can be articulated. This paper has attempted to establish the

Bryan K. Fite;bfite@meshco.com

fundamentals for a standards-based approach using core *Objects* and a standard *SDL* to support practical use cases and provide a vision of what is possible. As the *SDL* approach matures and is adopted in the field, new applications and innovations will materialize, supporting the needs of stakeholders practicing in the *Cyber Domain.* Further, this paper is intended as an 'a call to action' for the creation of an open standard to describe simulations for the benefit of the wider community.

Bryan K. Fite;bfite@meshco.com

6. References

EDURange Project, (2013). EDURange: A Cybersecurity Competition Platform to Enhance Undergraduate Security Analysis Skills. Retrieved October 20, 2013 from http://blogs.evergreen.edu/edurange/

Finkle, Jim & Randewich, Noel. Rueters, (2012). Online article. Retrieved October 19, 2013 from http://www.reuters.com/article/2012/06/12/us-media-tech-summit-symantec-idUSBRE85B1E220120612

Flight Simulator History, (2008). Flight Simulator Timeline. Retrieved October 20, 2013 from http://fshistory.simflight.com/fsh/timeline.htm

International Organization for Standardization, (1996). ISO/IEC 7498-1. Retrieved November 25, 2013 from http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip

Meshco Incorporated, (2009). Website and video. Retrieved October 19, 2013 from http://PacketWars.com/how-to-play

National Human Genome Research Institute, (2003). All About The Human Genome Project (HGP). Retrieved October 20, 2013 from http://www.genome.gov/10001772

Nilsson, Henrik University of Nottingham, (2010). ITU-FRP2010. Retrieved October 20, 2013 from http://www.cs.nott.ac.uk/~nhn/ITU-FRP2010/LectureNotes/lecture07-4up.pdf

Bryan K. Fite;bfite@meshco.com

Sophos, (2013). Security Threat Report 2013. Retrieved on October 19, 2013 from http://www.sophos.com/en-us/medialibrary/PDFs/other/sophossecuritythreatreport2013.pdf

Stanford University, (2002). Leonardo: Science, Technology, and Art. Retrieved October 20, 2013 from http://leonardodavinci.stanford.edu/submissions/clabaugh/history/leonardo.html

US Chairman of the Joint Chiefs of Staff, (2006). Joint Publiction 3-13: Information Operations. Retreived October 20, 2013 from http://www.carlisle.army.mil/DIME/documents/jp3_13.pdf

Verizon, (2013). 2013 Data Breach Investigations Report. Retrieved October 19, 2013 from http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2013_en_xg.pdf

Bryan K. Fite;bfite@meshco.com

7. Appendix A: Simulations

This section contains a list of well-known simulations. They vary widely in structure and purpose. This does not constitute a complete list of all simulations nor an endorsement of any specific simulation.

Simulation Name	URL	Description
Cyber Olympics	http://www.cyberlympics.org/	Global Assessment Competition
CYBER Patriot	http://www.uscyberpatriot.org	United States Air Force Association's National High School Cyber Defense Competition
Defcon CTF	http://www.defcon.org/html/links/dc-ctf.html	Annual Hacker Convention Capture The Flag Competition
Deter	http://deter-project.org/	Cyber Research Project
EDURange	http://blogs.evergreen.edu/edurange/	National Science Foundation Funded Competitive Security Exercises
National Collegiate Cyber Defense Competition	http://www.nationalccdc.org/	Collegiate Cyber Competition
Netwars	http://www.sans.org/netwars	SANS Training Platform
Packetwars™	http://packetwars.com	Cyber Sport: Information Operations Training and Assessment Platform
Trace Fire	http://csr.lanl.gov/tf/	Forensic Incident Response Exercise

Bryan K. Fite;bfite@meshco.com

8. Appendix B: Lexicon

This section contains a combination of terms, acronyms and definitions. Many are indicated throughout this document as capitalized and/or in italics. Additional terms, not used within the document, are included because of their general use within the domain so this Appendix can serve as a general reference.

Term	Definition
Actor	A participant in an active campaign, scenario or simulation. They play one of several roles; attacker, defender, stakeholder or observer.
Actor Platform	The computing device, typically a node, used by an actor in an active campaign to achieve an objective. Is also considered and asset.
Adversary	An actor whose objectives and motivations put them in competition or conflict with another actor.
Analytics	A term for the collection and review of data for meaningful and patterns.
ARP (acronym)	Address Resolution Protocol (see ARP)
ARP	A method used to resolve network layer addresses to link layer addresses.
ARP Table	Memory resident index of network layer addresses mapped against link layer addresses.
Artifact	Is a type of primitive object in the form of a file (text, audio, graphic or video) or credentials (account, username, password or key material).
Asset	Is a resource of tactical or strategic value, which can take the form of a human, object or capability
Attack Path	Refers to the specific route selected by an adversary out of all the known avenues of compromise.
Attack Surface	Refers to the total known avenues of compromise.
Attack Vector	Refers to the specific route selected by an adversary out of all the known avenues of compromise.
Attribution	The ability to identify a specific adversary or actor based on key indicators or forensic evidence.
Baselining	The act of defining a specific threshold associated with the normal operating parameters of a system.

Battle Node	An actor's platform within a Packetwars™ simulation.
Battle Space	The logical environmental demarcation associated with a simulation.
Beach Head	Common term used to describe a foothold acquired by an actor used to establish a pivot point for exploitation and lateral movement.
Blue Team	Common term for describing a group of defenders. Used primarily in reference to assessment exercises or staged simulations in production environments.
Campaign	A group of adversary actions designed to achieve a specific objective, which can span multiple battles or simulations.
Cloud	A common term for a shared tenant computing environment.
CSP (acronym)	Cloud Service Provider (see CSP)
CSP	An entity, normally commercial, which provides shared tenant computing services.
Combatant	One who is engaged in a struggle between one or more adversaries. Often used in Packetwars™ to describe players.
Constraint	A primitive object designed to shape a simulation by limiting the actor's range of motion and sphere of influence.
Counter Intelligence	The ability to use the enemies/advisory's knowledge of a condition to direct the adversary by manipulating their interpretation of the facts and reality.
Crypto	Refers to the discipline of crypto, cryptographic algorithms and otherwise the ability to transform plain text to cipher text.
Cyber	A common prefix used to denote the use of electromagnetic communications systems amongst actors.
Cyber Domain	The total potential sphere of influence afforded an actor or group of actors in the electromagnetic communications spectrum.
Cyber Operations	Actions conducted amongst actors in the electromagnetic communications spectrum.
Cyber Operations Specialist	A trained and experienced subject matter expert operating in the electromagnetic communications spectrum.

Bryan K. Fite;bfite@meshco.com

Deceive	A tactic and strategy used by one actor against another actor with the intent of to mislead by manipulating their perception of reality.
Declared Constraint	A primitive object designed to shape a simulation by limiting the actor's range of motion and sphere of influence, which is declared and known to simulation participants.
Declared Objective	A primitive object that defines the relative goals of a simulation, which is declared and known to simulation participants.
Defensive Fuzzing	The practice of systematically responding to unsolicited communications from external nodes directed by potential adversaries with the intent of gaining some advantage of said potential adversary.
Degrade	A tactic and strategy used by one actor against another actor with the intent of limiting the effectiveness or to otherwise retard an adversary's capability.
Deny	A tactic and strategy used by one actor against another actor with the intent of prevent an adversary's access to a capability.
Destroy	A tactic and strategy used by one actor against another actor with the intent of damaging an asset to the point that the adversary cannot exercise a capability.
Detect	A tactic and strategy used by one actor against another actor with the intent of discovering intrusions or indicators of compromise.
Disrupt	A tactic and strategy used by one actor against another actor with the intent of interrupting an adversary's flow of information.
Domain	An area of expertise or logical demarcation used to group assets.
Edutainment	A term used to describe the fusion of an educational experience and an entertainment experience.
Element	A single discrete component of a simulation.
Emulation	The ability, via software, to impersonate hardware via abstractions layers thereby allowing operating systems, applications and other software to execute on non-native hardware platforms.
Exfiltration	The act of extracting an asset from a security domain operated by an adversary.

Bryan K. Fite;bfite@meshco.com

Exploit	A tactic and strategy used by one actor against another actor with the intent of using an adversary's capability against themselves.
Fault Injection	The act of inserting known bugs or configuration flaws into a simulation.
Forensics	The science of determining the details of an event using digital or physical evidence.
Fuzzing	The practice and technique of populating computer interfaces with random, oversized and otherwise malformed input with the intent of exposing vulnerabilities.
Game	An entertaining competition which has rules, competitors and a promised reward.
Game Space	The logical environmental demarcation associated with a competition.
Honeypot	A generic term used to describe a dummy/fake system, service or other simulation element for the purpose of enticing or deceiving an adversary into revealing themselves, intent or as a form of counter-intelligence.
Incident response	Refers to the practice of responding to an event in a pre-defined way, typically associated with security events.
Includes	The introduction of messages or artifacts into a simulation.
Influence	A tactic and strategy used by one actor against another actor with the intent of causing other actors to behave in a favorable way.
Injects	The introduction of messages or artifacts into a simulation.
IO (acronym)	Information Operations (see IO)
IO	Also known as Cyber Operations encompasses the entire spectrum of Offensive and Defensive computing capabilities.
Lateral Movement	A term used to describe the tactic of leveraging a compromised asset to compromise another asset with the intent of advancing a campaign.
Medium	The physical platform used to communicate messages between actors and nodes.
Message	A primitive object that communicates information, data or instructions between simulation elements.

Bryan K. Fite;bfite@meshco.com

Model	The physical, conceptual or mathematical representation of systems, processes and organizational elements and define the rules of their interactions.
Motivation	Tactical and strategic objective(s) that drive and influence adversary behavior.
Network	The communication path or paths between nodes, typically layer 1-3 of the OSI model (http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip)
Node	Any layer 1-7 of the OSI model (http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip) connected element.
Object	A collection of predetermined attributes used to describe a simulation element.
Object Attributes	The characteristics that dictate how elements interact with each other inside a simulated environment.
Object ID	A unique identifier assigned to a simulation primitive.
Object Type	Simulation element classification designation with predetermined attributes that dictate which characteristics are mandatory and optional.
Objective	A primitive object that defines the relative goals of a simulation.
Packetwars™	A cyber-sport which simulates offensive and defensive computing scenarios for training, assessment and entertainment.
Perspective Filtering	The practice of creating the appropriate telemetry data view based on actor role and assessment objectives.
Primitives	The 9 fundamental object types used to define simulations in the Simulation Definition Language
Process	A primitive object describing the workflow associated with a pre-defined simulation element interaction.
Protect	A tactic and strategy used by one actor against another actor with the intent of guard an asset against compromise.
Red Team	Common term for describing a group of attackers. Used primarily in reference to assessment exercises or staged simulations in production environments.

Bryan K. Fite;bfite@meshco.com

Respond	A tactic and strategy used by one actor against another actor with the intent of reacting to an adversary in the optimum way.
Restore	A tactic and strategy used by one actor against another actor with the intent of bringing assets back to a known good functional state.
Route	The logical path information takes between nodes and networks.
Routing Table	An index of the logic paths information takes between specific nodes and networks.
Scenario	A logical construct providing the context and all other required simulation elements in a human friendly format.
Scenario Definition Language	A way to model Cyber Operations Simulation elements by abstracting the primitives as objects and describing their interaction.
Security Domain	A demarcation that defines a logical or physical within the control or custodianship of an actor.
Shared Tenant Environment	Sometimes referred to as "The Cloud". This computing environment leverages a common infrastructure to provide services to multiple entities, typically as part of a commercial operation.
Simulation	An artificial construct designed to emulate a real world process or system over time.
Simulation Management Console	The node used to control elements within an training or assessment environment.
Simulation Network	The communication environment hosting simulation assets sometimes referred to as the Game Space.
Software	Operating system, application or service.
Sphere Of Influence	The breadth and depth of an actor's capabilities in relation to another actor or actors.
Target	An identified objective usually considered an asset.
Telemetry	Data provided by simulation elements that can be used to provide visibility into significant simulation events.
Telemetry Platform	The infrastructure for the collection, transformation and consumption of simulation data feeds.
Trust Boundary	The physical or virtual interface between two unlike security domains

Bryan K. Fite;bfite@meshco.com

Universe	Refers to simulated environment that is made up of a combination of primitive objects and governed by the rules that dedicate how those primitive objects interaction with each other. The rules governing a particular simulation may or may not align with the rules governing the "real world"; physics, chemistry, biology and the like.
Visualization	The transformation of simulation telemetry into a meaningful graphic representation.
Vulnerability	A bug, misconfiguration or other condition that exposes an asset to a specific threat.
Vulnerability Injection	Including a known bug or configuration flaw into a simulation, typically a bug with a publically available exploit.
YAML (acronym)	Yet Another Markup Language (see YAML)
YAML	Is a human readable serialization standard for data and can be used for all programming languages.

Bryan K. Fite;bfite@meshco.com

9. Appendix C: Object Type Definitions

Object Type: *Node*
Required Attributes: *Name, Host / Gateway Flag, OS, Interface Address(es) and listening ports*
Optional Attributes: *Accounts, Applications, Artifact* and Services

Syntax:
Object ID *(unique number):* **Object Type** *(Node):***Name** *(Name):***H/G***(H or G):***OS** *(OS ID):***Interface address***(IP address,1 IP address2, IP address3…):***Listening Ports** *(TCP Port1, Port2, Port3…):User Account(User1, User2, User3…)*
Example:

Object Type: *Network*
Required Attributes: *Name, Layer 1 (Protocol, Address and Security Domain [Public, Private or DMZ]), Layer 2 (Protocol, Address and Security Domain [Public, Private or DMZ]), Layer 3 (Port, Protocol, Address and Security Domain [Public, Private or DMZ]).*
Optional Attributes: Routing, Capacity, ACL's and Local/Remote flag

Syntax:
Object ID *(unique number):* **Object Type** *(Network):***Name** *(Name):***L1***(P,A,PU/PR/D):* **L2***(P,A,PU/PR/D):* **L3***(P,A,PU/PR/D):***OS** *(OS ID):***Routing:***(static route):***L/R** *(L or R)*

Object Type: *Software*
Required Attributes: *Name, Vendor, Version*
Optional Attributes: Dependencies, requirements, Files, Files Sizes, File Hashes, Configuration and Comments

Syntax:
Object ID *(unique number):* **Object Type** *(Software):* **Name** *(Name):* **Vendor** *(vendor name):* **Version** *(version#):* **Files** *(file1, file2, file3…),* **File Sizes** *(FS1, FS2, FS3…),* **File Hashes** *(FH1, FH2, FH3…),* **Comments** *(comments-description of software function)*

Object Type: *Artifact*
Required Attributes: *Name, Media Type (service element, written/physical, email, sms/txt, file), Artifact Type (binary, service element, identity element, informational message)*
Optional Attributes: Comment

Syntax:
Object ID *(unique number):* **Object Type** *(Artifact):* **Name** *(Name):* **Media Type** *(SE, W/P, E,S/T or F):* **AT***(B,SE,IE or IM):* **Comment***(context)*

Bryan K. Fite;bfite@meshco.com

Object Type: *Constraint*
Required Attributes: *Name, Constraint Type (Environmental or capability), Constraint Description*
Optional Attributes: Comments

Syntax:
Object ID *(unique number):* **Object Type** *(Constraint):* **Name** *(Name):* **CT** *(E or C):* **Constraint** *(description):* **Comments** *(context)*

Object Type: *Objective*
Required Attributes: Name, Objective Class (Attacker, Defender or Assessment), *Objective Type (Intel, Compromise, Escalate Privilege, Exfiltration, Destroy, Disrupt, Degrade, Deny, Deceive, Exploit, Influence, Protect, Detect, Restore and Respond), Objective and Attestation Method (key, flag, file, shared secret, hash file, moderator observation or, demonstrable capability)*
Optional Attributes: Motivation Qualifiers and Comments

Syntax:
Object ID *(unique number):* **Object Class** *(Objective Class):* **Object Type** *(I,C,EP,E,D1,D2,D3,D4 or D5):***Name** *(Name):***Objective***(objective):* **Attestation Method (***method):* **Comments***(context)*

Object Type: *Actor*
Required Attributes: *Name, Alignment (Attacker, Defender, Both or Neutral), Actor Class (Adversary type), Role*
Optional Attributes: Capabilities (Skill Index), Handicap and Comments

Syntax:
Object ID *(unique number):* **Object Type** *(Actor):* **Name** *(Name):* **Alignment** *(alignment type):* **Actor Class** *(Adversary/Worker):* **Capabilities** *(Skill1, Rating1, Skill2, Rating2, Skill3, Rating3…)*

Object Type: *Process*
Required Attributes: *Name, Function Description, Function Flowchart*
Optional Attributes: Implementation (manual or automated) and Comments

Syntax:
Object ID *(unique number):* **Object Type** *(Process):* **Name** *(Name):* **Function Description** *(human friendly description of process):* **Implementation** *(automated script):* **Comments** *(context)*

Object Type: *Message*
Required Attributes: *Name, Media Type (live, written/physical, email, sms/txt, file, audio or video), Broadcast/Directed flag, encrypted/obfuscated/plaintext switch, Message type (scenario setup or inject), Message Content*
Optional Attributes: TBD

Bryan K. Fite;bfite@meshco.com

Syntax:
Object ID *(unique number):* **Object Type** *(Message):*__Media Type__ *(L,W/P,E,S/T,F,A or V):*__B/D__*(B or D):* __E/O/P__*(E, O or P):* **Message Type***(S or I):*__Message Content__ *(message*

Bryan K. Fite;bfite@meshco.com

Last Updated: January 1st, 2018

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Security East 2018	New Orleans, LAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SEC599: Defeat Advanced Adversaries	San Francisco, CAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, NL	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Dubai 2018	Dubai, AE	Jan 27, 2018 - Feb 01, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NVUS	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Miami 2018	Miami, FLUS	Jan 29, 2018 - Feb 03, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MDUS	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS Scottsdale 2018	Scottsdale, AZUS	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS London February 2018	London, GB	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS SEC455: SIEM Design Beta One 2018	Arlington, VAUS	Feb 12, 2018 - Feb 13, 2018	Live Event
SANS Secure India 2018	Bangalore, IN	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CAUS	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Dallas 2018	Dallas, TXUS	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Secure Japan 2018	Tokyo, JP	Feb 19, 2018 - Mar 03, 2018	Live Event
SANS Brussels February 2018	Brussels, BE	Feb 19, 2018 - Feb 24, 2018	Live Event
Cloud Security Summit & Training 2018	San Diego, CAUS	Feb 19, 2018 - Feb 26, 2018	Live Event
SANS New York City Winter 2018	New York, NYUS	Feb 26, 2018 - Mar 03, 2018	Live Event
CyberThreat Summit 2018	London, GB	Feb 27, 2018 - Feb 28, 2018	Live Event
SANS London March 2018	London, GB	Mar 05, 2018 - Mar 10, 2018	Live Event
SANS Secure Singapore 2018	Singapore, SG	Mar 12, 2018 - Mar 24, 2018	Live Event
SANS Paris March 2018	Paris, FR	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS San Francisco Spring 2018	San Francisco, CAUS	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Secure Osaka 2018	Osaka, JP	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	McLean, VAUS	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Munich March 2018	Munich, DE	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, AU	Mar 19, 2018 - Mar 24, 2018	Live Event
ICS Security Summit & Training 2018	Orlando, FLUS	Mar 19, 2018 - Mar 26, 2018	Live Event
SANS Pen Test Austin 2018	Austin, TXUS	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Boston Spring 2018	Boston, MAUS	Mar 25, 2018 - Mar 30, 2018	Live Event
SANS SEC460: Enterprise Threat Beta	OnlineCAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced