# Plan for Evaluating the Hive Mind Concept for GENI

http://hivemind.cs.ucdavis.edu/

PI: Sean Peisert
Department of Computer Science
University of California, Davis
peisert@cs.ucdavis.edu

March 11, 2011

## 1 GENI Introduction

The Global Environment for Network Innovations (GENI) is an NSF-funded, BBN-operated testbed designed to support experimental research in network science and engineering. The goals for GENI are vast: they seek to "understand networks broadly and at multiple layers of abstraction from the physical substrates through the architecture and protocols to networks of people, organizations, and societies" where "the intellectual space surrounding this challenge is highly interdisciplinary, ranging from new research in network and distributed system design to the theoretical underpinnings of network science, network policy and economics, societal values, and the dynamic interactions of the physical and social spheres with communications networks." [GEN08c] To provide a backbone, or at least a simulation of "the structure, behavior, and dynamics of our most complex systems—networks of networks," GENI seeks "to support a wide range of experimental protocols, and data dissemination techniques running over facilities such as fiber optics with next- generation optical switches, novel high-speed routers, city-wide experimental urban radio networks, high-end computational clusters, and sensor grids." [GEN08c] Finally, the "GENI suite is envisioned to be shared among a large number of individual, simultaneous experiments with extensive instrumentation that makes it easy to collect, analyze, and share real measurements." [GEN08c]

## 2 GENI Architecture

The GENI vision is one of a global network in which users—generally experimenters—can request computing, networking, and cyber-physical resources for use in experimental research. Different parts of the GENI suite are owned and/or operated by different organizations, principally including NSF. Those organizations are connected to the GENI infrastructure through a process of *federation*. GENI experiments are an interconnected set of reserved resources, or *slices*, on platforms in diverse locations [GEN08a].

Ultimately, GENI will be highly distributed. That is, though an NSF *clearinghouse* can demand, and possibly enforce globally federated policies, and collections of operations centers around the world (for international partners, countries such as Japan or the European Union) can enforce those policies after they have been negotiated. Currently, GENI exists as a set of enhancements either one of several control frameworks, including Emulab [WLS+02] (called ProtoGENI [Pro]), PlanetLab [Pla], ORCA [ORC], and ORBIT [ORB].

In order for GENI to function as a usable testbed, interconnected with the Internet, and with virtual "slices" running over or near production systems and networks, GENI needs to be both secure and reliable [GEN08b, GEN08c]. There need to be facilities to protect GENI experiments from the outside world, to protect the outside world from GENI, and to protect the systems and networks running virtual GENI hosts and slices from attack by GENI experiments themselves [Bis09], despite the increased privileges that such virtual hosts implicitly give.

# 3 GENI Security

GENI faces significant security challenges not faced by typical enterprise networks:

> "The scale of GENI, in terms of its number of resources and their geographic spread, makes it an attractive launch pad for large-scale attacks. The national and international attention garnered by the GENI project makes it an attractive target for attacks, for bragging rights if not anything else. The use of GENI for virtually any kind of networking experiment, including long-lived experiments, makes it an attractive platform for hiding and distributing illicit content." [GEN10]

Our goal is to take steps toward providing a framework for securing such a system. Given its current incarnation of four control frameworks, our current focus is on the *ProtoGENI* architecture and implementation, and, through federation from ProtoGENI, also the *DETER* testbed [BBK+06].

# 4 Hive Mind

The *Hive Mind* project is a collaboration between researchers at the University of California, Davis, Battelle, CA Labs, and Wake Forest University.

Unlike traditional network [HDL+90] or host based [KFL94] intrusion detection methods, the Hive Mind project explores a distributed method based on mobile code concepts and swarm intelligence. The goal is to provide a lightweight, distributed detection method—*digital ants*—that is adaptable to changing threats and that allows suspicious activity to be communicated across hierarchical layers and to humans-in-the-loop who can direct actions when needed. This differs even from network intrusion detection systems that perform hierarchical alert correlation [VVKK04, ZHR+07] in numerous ways, particularly in the way that the leaf nodes of the hierarchy are very lightweight, specialized sensors.

In the Hive Mind, humans and various types of software agents share the responsibilities of securing an infrastructure comprised of enclaves that belong to member organizations. One human can supervise a multi-enclave system with a few enclave-level agents, host-level agents at each monitored machine or group of similar machines, and a large swarm of simple mobile agents. Our terminology is as follows:

- Humans function as *Supervisors*. They provide guidance to and receive feedback from one or more enclaves. Action is required of them only when the lower-level agents encounter a problem that requires human involvement.
- Enclave-level agents called *Sergeants* are each responsible for the security state of an entire enclave.
- Host-level agents called *Sentinels* protect and configure a single host, node, or a collection of similarly configured hosts and nodes such as a cluster or storage network.
- Swarming agents, called Sensors, are the *digital ants* that roam from device to device within their enclave searching for problems and reporting to the appropriate Sentinel.

The Hive Mind's structure was inspired by the success of previous research by researchers part of our own team [HFFM08, HFM+09, MHF+09]—as well as others [BDT99, ESK01, PNBA06, Sel88, Smi02]—which suggests hierarchical arrangements of heterogeneous agents. Interposing logic-based rational agents between the humans and the swarm provides a basis for communication, interaction, and shared initiative. The hierarchical arrangement gives humans a single point of influence that allows multiple points of effect.

Our goal with the Hive Mind is not to replace "traditional" security mechanisms such as network and host-based intrusion detection systems, firewalls, or access control, at least not where it is feasible to also run such systems. Rather, except in very specific situations where computing power is highly limited, and so a host-based IDS would be impossible, we believe that the Hive Mind can *augment* all of these systems. For example, we believe that a traditional firewall and/or network intrusion detection system (NIDS) may be best equipped to guard the border between GENI and the regular Internet, and so running them in parallel complement to the Hive Mind would be ideal.

# 5 Current Status and Evaluation

Our earlier digital prototypes included a NetLogo [Wil99] version— initially developed by researchers at Pacific Northwest National Laboratory and Wake Forest University—that demonstrated a graphical representation of the function of the digital ants, as well as a very preliminary Java implementation—also developed by researchers at Wake Forest University—of sensors and sentinels.

Researchers in the Hive Mind team at the University of California, Davis have now developed a functional Python prototype of our digital ants work and have begun a graphical view/demo on top of the Python prototype to show an operator how the digital ants are functioning. The current Python prototype re-implements sensors, sentinels, and a rudimentary sergeant concept. We have also implemented and run this collection of digital ant programs on the ProtoGENI and DETER testbeds [HFM+11]. While we are still in the process of developing and expanding the prototype, we now have enough of a conceptual

idea of how the digital ants map to GENI that we can develop a means for evaluating their success.

We previously discussed the architecture of our Hive Mind approach [Pei10] and identified many of the requirements and challenges that exist in such an architecture. One key challenge is the concept of disparate administrative domains, encompassing numerous, potentially conflicting security policies. Another challenge is the fact that the resources allocated may or may not even be full-scale general purpose computer systems, but may be low-power devices, specialized components, and/or possibly even non-compute systems. Or, more traditional security mechanisms, such as host-based intrusion detection systems may interfere with experiments on GENI. Indeed, we believe that the GENI environment is one in which traditional intrusion detection systems are likely to be too heavyweight and therefore inappropriate.

Given that our digital ant approach aims to solve these issues, our evaluation therefore will seek to determine (a) the set of security policies that the mechanism can enforce, and (b) its resource usage. To evaluate these metrics, our plan is to create several scenarios that span an attack space of representative threats, create a suite of attack scripts to simulate these scenarios, and, and finally, evaluate the Hive Mind concept on its performance and ability to detect the attacks, and therefore violations of the security policies.

We will compare the abilities and performance of the digital ants concept with more the more traditional IDS concept as well. We initially plan to (a) compare the range of detection mechanisms that mobile, digital ants can implement with a "stationary" and more centrally-controlled modification to the digital ants concept; and then (b) compare the performance overhead of the two implementations.

We will eventually also evaluate digital ants policy implementation and resource usage in comparison to more heavyweight IDS systems. Comparative systems include network intrusion detection systems (NIDS) such as Bro [Pax99] or Snort [Roe99], as well as host-based intrusion detection systems (HIDS) such as OSSEC [HCB08]. Again, we do this not because it is our goal to replace traditional IDSs with digital ants, but to determine the best place to use traditional IDSs (e.g., the GENI border gateway) and where to use digital ants.

## 5.1 Ability to Implement Security Policies

In particular, to demonstrate the ability of digital ants and IDSs to implement security policies, we will choose scenarios that span a space of attacks against all key elements of GENI, which may include but not be limited to many of the following examples:

1. Attacks by GENI experiments against the GENI control plane
   (a) Denial of network service via packet flooding
   (b) Denial of service against commands between control plane and experiments
   (c) Altering commands between control plane and experiments
   (d) Network compromise, including against routing tables, etc...
2. Attacks by GENI experiments against the physical hosts they are running on
   (a) Denial of service (e.g., CPU cycles, disk space)
   (b) Root compromise of the host operating system
3. Attacks by GENI experiments against cyber-physical devices on the network

   (a) Denial of service
   (b) Causing cyber-physical devices to exceed operational limits (e.g., such as with Project Aurora)
4. Attacks by GENI experiments against other experiments on the same physical host
   (a) Denial of service (e.g., CPU cycles, disk space)
   (b) Side-channel attack against other virtual hosts (e.g., capturing encryption keys)
   (c) Perturbation or capture of results of other experiments
5. Attacks by GENI experiments against other GENI experiments within the same administrative domain
   (a) Denial of service
   (b) Side-channel attacks
   (c) Perturbation or capture of results of other experiments
6. Attacks by GENI experiments against the Internet
   (a) Denial of service
   (b) Malware
7. Attacks by the GENI control plane against the Internet
   (a) Denial of service
   (b) Malware
8. Attacks from the Internet against the GENI control plane
   (a) Denial of service
   (b) Malware
   (c) Network compromise, including against routing tables, etc...

## 5.2 Resource Usage

There are many axes for possible performance evaluation. Among the ones that we will explore may include—but are not limited to—the following:

1. What is the resource overhead (e.g., processor, disk space requirements) on hosts?
2. What is the resource overhead (e.g., bandwidth, latency, packet loss) on networks?
3. What was the resource use before, during and after the period of the attack that we can attribute to the detection system (not the attacks)?
4. What effect does the security system have on the testbed's control plane?
5. What effect does the security system have on researchers' experiments?
6. How does the security system affect consistent behavior of experiments?
7. How does the digital ant concept scale for different size networks with different numbers of physical and virtual hosts, and therefore different numbers of ants, sergeants, and sentinels needed?

## 5.3 Design Decisions

In addition to the evaluations and comparisons that we will make with stationary ants and traditional IDSs, we will also evaluate a number of our own design decisions for digital ants.

That is, our solution is is bio-inspired. It is not our intent to simulate actual biological entities. Our current solution contains many diversions from real biology. For example, real ants leave pheromone at many times, not just when they find food. Real ants bring resource back to the nest when they find food and don't wander off in a random direction. Real ants sense pheromone it when in the vicinity of pheromone drops, and head toward the trail, and not just when they cross the pheromone trail. Real ants are "created" only in the nest, and not at random points in the "ant environment"; our solution has no concept of a nest.

Similarly our digital ants are influenced greatly by "crowding" whereas real ants are not. While there is some statistical truth that higher ant densities increase ant loss due to predation, death rate is based on lifespan, not a pseudorandomly-generated chance of death. Additionally, abject famine and disease may increase death rate, but primarily population is controlled by adjusting birth rate based on available resources. Colonies that get top large split, they do not have mass deaths.

Of key importance, there are many real ant species, each with behavioral variations. Just as an algorithm may have enhancements made to a digital ant, we see this in real ants as well. For example, many other swarm-based models have enhancements to the basic ant foraging process. Likewise certain species of real ants have a sense of direction and will not follow the trail if it seem to be going the wrong way to the nest. Real ants must also compete against ants of their own and other species. Our ants currently come in many, many types (i.e., sensor functions), whereas real ants do not, or at least not in the same way or number of variants within one species. If we consider the different ant types different species, then the fact that the ants communicate to the other ant species so that they can benefit (instead of attacking each other), is also a digression from real ants.

Finally, the absorption and evaporation of pheromone for real ants are affected by substrate, temperature, weather, time, etc... This causes real ants to behave differently on different substrates and environmental conditions. Also, pheromone is generally long lived, hours, months, even years, with subtle behavioral aspects.

As specified, our current approach is can be thought of as an online search algorithm. It is related to neural networks [Tur48] and other swarm intelligence models [BDT99], such as *Ant Colony Optimization (ACO)* [DS04] and *Particle Swarm Optimization (PSO)* [ESK01] but contains important differences. For example, our approach is more of a distributed, parallel real-time search algorithm and secondarily a resource allocation algorithm than many existing swarm algorithms, which are closer to being optimizations of some sort: shortest path, clustering, minimum error, etc. Also, our approach is also related to artificial immune systems [FPP86, SHF97] in the way that resources are directed, however, digital ants do not currently contain immune system's primary function of identifying self from non-self.

Thus, examples of some of the variations of our current approach, as well as on swarm intelligence, neural networks, artificial immune systems that we may explore include, but are not limited to:

1. Generally, how "ant-like" and close to the concept of a real ant paradigm does it make sense for our solution to be? When does "antness" over complicate or interfere with the goals of the project, including accuracy, ease of use/configuration, performance, and environmental conditions?
2. How might influence from other insects such as bees or termites enhance or detract from our approach?

3. Does the concept of having "nests" make sense?
4. Is the four-layer approach – sensors, sentinels, sergeants, and human supervisors – appropriate, or are more or fewer layers needed?
5. What is the relative effectiveness of digital ants running in the control plane, in the hypervisor plane, and within the virtual hosts?
6. Should ants be able to communicate directly with other ants, or only via pheromones?
7. Do ants achieve a steady state, developing a sense of selve over space and time?
8. Can both forms of communication (rather than just one or the other) be beneficial?
9. Should a *gather* and *scatter* approach to instant communication with ants be possible, analogous to the concepts of the same name used for message passing with high-performance computing?
10. Should ants "wander" or be "teleported" to and from suspected attacks?
11. Should ants be stationary (like traditional IDSs) or mobile?
12. What effect would self-non-self discrimination have?
13. Is there sufficient number of monitoring elements, or ratio of monitoring elements to hosts, to make the digital ant model successful? If not, how can we modify the system to mitigate this?
14. Attacks that spread within a slice, e.g., if an experiment went bad or was compromised by malware from the Internet, may cross different enclaves. This will have visibility across multiple enclaves. Multiple sergeants may observe the activity. In such a case, how wil sergeants interact/communicate?

We will add additional metrics to study and experiment with during the evaluation process and as our prototype evolves.

# 6 Final Thoughts

Finally, we note that in addition to providing a security layer, another goal of the digital ant system is to provide data that is potentially useful to networking experiments. We do not seek to measure the efficacy of the system's ability to do this, but mention it because GENI projects who can make use of this data may provide both guidance and feedback on its usefulness.

More information about the Hive Mind project can be obtained at our project web site: `http://hivemind.cs.ucdavis.edu/`

# References

[BBK+06] Terry Benzel, Robert Braden, Dongho Kim, Clifford Neuman, Anthony Joseph, Keith Sklower, Ron Ostrenga, and Stephen Schwab. Experience with DETER: A Testbed for Security Research. In *Proceedings of the 2nd International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, March 2006.

[BDT99] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems (Santa Fe Institute Studies in the Sciences of Complexity Proceedings)*. Oxford University Press, USA, 1999.

[Bis09] Matt Bishop *et al.* GENI and Security Workshop Final Report, February 2009.

[DS04] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. The MIT Press, 2004.

[ESK01] Russell C. Eberhart, Yuhui Shi, and James Kennedy. *Swarm Intelligence (The Morgan Kaufmann Series in Evolutionary Computation)*. Morgan Kaufmann, 2001.

[FPP86] J. Doyne Farmer, Norman H. Packard, and Alan S. Perelson. The Immune System, Adaptation, and Machine Learning. *Physica D: Nonlinear Phenomena*, 22(1-3):187–204, 1986.

[GEN08a] GENI Project Office. GENI Solicitation 2, December 15, 2008.

[GEN08b] GENI Project Office. GENI Spiral 1 Overview, September 29, 2008.

[GEN08c] GENI Project Office. GENI System Overview, September 29, 2008.

[GEN10] GENI Project Office. GENI Spiral 2 Security Plan, March 15, 2010.

[HCB08] Andrew Hay, Daniel Cid, and Rory Bray. *OSSEC HIDS Host-Based Intrusion Detection Guide*. Syngress, Burlington, MA, 2008.

[HDL+90] L. Todd Heberlein, Gihan V. Dias, Karl N. Levitt, Biswanath Mukherjee, Jeff Wood, and David Wolber. A Network Security Monitor. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pages 296–304, Oakland, CA, May 1990.

[HFFM08] Jereme N. Haack, Glenn A. Fink, Errin W. Fulp, and Wendy M. Maiden. Cooperative Infrastructure Defense. In *Proceedings of the Workshop on Visualization for Computer Security (VizSec)*, 2008.

[HFM+09] Jereme N. Haack, Glenn A. Fink, Wendy M. Maiden, A. David McKinnon, and Errin W. Fulp. Mixed-Initiative Cyber Security: Putting Humans in the Right Loop. In *Proceedings of the First International Workshop on Mixed-Initiative Multiagent Systems (MIMS)*, Budapest, Hungary, May 11 2009.

[HFM+11] Jereme N. Haack, Glenn A. Fink, Wendy M. Maiden, A. David McKinnon, Steven J. Templeton, and Errin W. Fulp. Ant-Based Cyber Security. In *Proceedings of the 8th International Conference on Information Technology: New Generations (ITNG)*, April 11–13 2011.

[KFL94] Calvin Ko, George Fink, and Karl Levitt. Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring. In *Proceedings of the 10th Annual Computer Security Applications Conference (ACSAC)*, pages 154–163, Dec. 5–9, 1994.

[MHF+09] Wendy M. Maiden, Jereme N. Haack, Glenn A. Fink, A. David McKinnon, and Errin W. Fulp. Trust Management in Swarm-Based Autonomic Computing Systems. In *Proceedings of the IEEE Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pages 46–53, 2009.

[ORB] ORBIT. `http://www.orbit-lab.org/`.

[ORC] ORCA. `https://geni-orca.renci.org/`.

[Pax99] Vern Paxson. Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23):2435–2463, 1999.

[Pei10] Sean Peisert *et al.* "Hive Mind" Design Specification for GENI Security Framework, September 2010.

[Pla] PlanetLab. `http://www.planet-lab.org/`.

[PNBA06] H. Van Dyke Parunak, Paul Nielsen, Sven Brueckner, and Rafael Alonso. Hybrid Multi-Agent Systems: Integrating Swarming and BDI Agents. In *Proceedings of the 4th International Workshop on Engineering Self-Organising Systems (ESOA)*, Hakodate, Japan, May 9 2006. Springer.

[Pro] ProtoGENI. `http://www.protogeni.net/`.

[Roe99] Martin Roesch. Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th Large Installation System Administration Conference (LISA)*, Seattle, WA, November 7–12 1999.

[Sel88] Oliver G. Selfridge. Pandemonium: A Paradigm for Learning. In *Neurocomputing: Foundations of research*, pages 115–122. MIT Press, 1988.

[SHF97] Anil Somayaji, Steven Hofmeyr, and Stephanie Forrest. Principles of a Computer Immune System. In *Proceedings of the 1997 New Security Paradigms Workshop (NSPW)*, pages 75–82, Great Langdale, Cumbria, UK, 1997.

[Smi02] Frank Smieja. The Pandemonium System of Reflective Agents. *IEEE Transactions on Neural Networks*, 7(1):97–106, 2002.

[Tur48] Alan M. Turing. Intelligent Machinery. Technical report, National Physical Laboratory Report, 1948.

[VVKK04] Fredrik Valeur, Giovanni Vigna, Christopher Kruegel, and Richard A. Kemmerer. A Comprehensive Approach to Intrusion Detection Alert Correlation. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 1(3):146–169, July–September 2004.

[Wil99] Uri Wilensky. NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999.

[WLS+02] Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, and Abhijeet Joglekar. An Integrated Experimental Environment for Distributed Systems and Networks. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, pages 255–270, Boston, MA, December 2002.

[ZHR+07] Jingmin Zhou, Mark Heckman, Brennan Reynolds, Adam Carlson, and Matt Bishop. Modelling Network Intrusion Detection Alerts for Correlation. *ACM Transactions on Information and System Security (TISSEC)*, 10(1), February 2007.