Milestone ExptsSec: S3.c

Report of the experiments designed in Milestone b

This report includes two parts. July 22, 2011.

PART I

ExptsSec: S3.c Report

Xiaoyan Hong, Dawei Li, Jason Bowman, Darwin Witt, Bo Gu

The department of Computer Science, University of Alabama

July 20, 2011

* The details about the results listed here are presented in the paper "Evaluation of Security Vulnerabilities by Using ProtoGENI as a Launchpad", accepted by *IEEE Globecom 2011*, Houston, December 2011.

1 Project Goals:

Explore ProtoGENI system security vulnerabilities by acting as a purposeful ProtoGENI User; Make improvement suggestions based on findings.

2 Project Approaches:

2.1 Learning control system features:

a) Internal Resource and Control Channel;
b) Resource Co-Location of Slices;
c) ProtoGENI Virtualization Technology;
d) OS Images.

2.2 Design security experiments according to ProtoGENI system features to:

a) Compromise the confidentiality and availability of other experiments using ProtoGENI Resources. Attacks include sending packets across slices, ARP poisoning attack, etc.
b) Utilize ProtoGENI resources to attack global Internet users, and Flooding attack.

2.3 Experiments Tools: Netwox, Stress, Ping, Iperf etc.

3 Findings and Suggestions:

3.1 Control plane and data plane:

The control-data plane architecture leaves attackers the chance to launch ARP cache poisoning attack. We observed that the control-router won't allow its ARP cache to be modified. But we also observed that an experimental node's ARP cache can be modified. The experiment shows that the ARP attack will compromise the availability of ProtoGENI resources. We further experimented on the isolation issue between a virtual link connection and related physical link using ARP attack. We observed that the failure of the control network connection won't affect the experimental topology.

Suggestions: Use static addresses in ARP table or tools like APRon, or ARPwatch on nodes. Or more general, the control network should setup monitor tool or detection system to detect and identify these malicious packets.

3.2 Date plane and data plane:

Data Plane to Data Plane attack is the kind of attacks launched by one ProtoGENI user to other users' running experiments. We touched two issues: wireless network experiments and virtualization.

a) Wireless experiments: We have observed that packet sniffing and packet spoofing are easy to launch. We have observed that ARP cache poisoning can cause the victim node to find that its wireless nodes are disconnected and the experiment cannot be continued.

b) Virtualization: Our experiments conducted in two ways and found that bugs in virtualization can be exploited, in our case, resulting in network traffic was not isolated across slices. This could also give a way for propagating malware. This bug has been reported to the ProtoGENI team and was fixed already.

c) Another virtualization related issue about Vnodes which are co-located on the same physical machine. We performed stress test on one of the co-located Vnodes. The experiment shows that the CPU resource is dedicated to a particular Vnode, but memory resources are shared among different Vnodes residing in the same physical machine. This drawback could lead to DOS that exhausts the memory resource so to interrupt normal ProtoGENI experiments.

Suggestions: The resource providers and control framework development team could enhance user policies and add monitoring mechanisms at resource owner side.

3.3 Data Plane to Internet

Sending traffic from data plane to outside Internet could lead to danger. We analyzed the case that if unknown virus or worm is the traffic payload. We also performed experiments of ping flood to overwhelm victim nodes in Internet which

has a smaller bandwidth. We found that both types of attacks were possible. It was also interesting to notice that our ping flooding tests were detected by Utah team.

Suggestions: the DETER testbed has provided many expertise in this area. To avoid being caught by virus, one can encrypt data traffic retrieved from the ProtoGENI so that the data won't be executed automatically by the users at their local machines. To avoid ping flooding (or same type of DOS attacks), strict Firewalls can be set between resource nodes and the Internet.

Part II

GENI Project Report by W4- Net Lab (July 20, 2011)

Yang Xiao, Jingcheng Gao, and Bo Fu

Dept. of Computer Science, The University of Alabama

July 20, 2011;

1. Summary

We have conducted two works for GENI:

1. Experiments of DOS attacks to ProtoGENI
2. Analysis of Access control of GENI and protogeni

2. DoS attacks

Various DoS attacks have applied to the ProtoGENI nodes.

Classic DoS Attack: ICMP flood attack worked on the ProtoGENI node. After flooding the Victim's Node using ICMP flood for about 20minutes, the Victim's node rebooted itself. Also we conducted UDP flood attack on the ProtoGENI node; through the packets sniffed by the "victim" node we can conclude that given enough flood traffic flow, DoS is possible.

Advanced DoS Attack: we conducted source spoof attack and SYN spoof attack. Both spoof attack make the victim's SSH connection not open anymore. Even after we checked that sliverstatus.py showed the node was "started" and "ready", we could not use ssh command line to connect the victim anymore.

3. Analysis on GENI Access Control

Normally, ProtoGENI access control is performed with the credential mechanism. The credential mechanism allows identification and authentication in both directions, and allows validating privileges which a principal wishes to invoke. Credentials are

authenticated documents which describe privileges held by a principal, and they unambigiously identify the owning principal. Credentials are represented as XML documents . Credentials are used for authorize actions where certificates authenticate. Credentials specify the permissions of the owner relative to a target object. From the view of access control, credentials specify the permissions of the subject to object. For example, a “slice credential” gives a user the right to allocate and remove resources from a slice. The only differences between the ProtoGENI credential and general GENI credential is that the GENI credential allows for different privileges that might be used in other control frameworks, such as Planet Lab's SFA. In ProtoGENI, delegated credentials behave fairly similarly to the basic credentials, but instead of a signature from the authority named in the target, any delegated credential is instead signed by the owner of the parent. From this feature of delegation from along with what we have discussed, we think that ProtoGENI uses discretionary access control policy, in which an entity may enable another entity to access system resources by granting access rights from one entity to another. Access control is also guaranteed by expiration date: the current time must be no later than the expiration date of the credential.

SUGGESTIONS ON GENI ACCESS CONTROL: We think that the discretionary access control (DAC) is widely implemented in the GENI projects. Role-based access control (RBAC) defines the access control based on the roles that the users are assigned rather than the user’s identity that controls the access right in discretionary access control. The idea of DAC is clear and it is more easily to be implemented than RBAC; RBAC is flexible because of the existence of the roles.

It is possible that a GENI testbed involves a large number of researchers, users, and administrators. By using RBAC, each type of user is viewed as a role, and it is easier to control the access right of a type of users than to control the access right of an individual users. Considering the flexibility of RBAC, in this section we present a scheme of applying RBAC for ProtoGENI access control. In this scheme, we try to add credentials into the conventional RBAC scheme.

4. Future Work

1. Analysis of Authentication of GENI and Protogeni
2. Access control analysis for other clusters of GENI
3. As for spoof flood we only exploit one optional tool from netwox, there are still two other tools, such IP/ICMP spoof and TCP/IP spoof we can test for the future.
4. We didn’t fulfill the *DDoS, Reflection*, *Amplification DoS,* etc. attacks this time, but these attacks have the same feasible background: as long as the basic DoS attacks we have conducted are possible, these more complicated DoS attacks are more possibly can be successfully launched.

5. Except for using netwox, iperf, etc. as attack tools, we can possibly write our own socket communication programs corresponding to ProtoGENI nodes API, which might be same/similar to other Linux system networking programming API.