Industrial Control Systems Honeypot

May1601

Dan Borgerding
Jon Hope
Nik Kinkel
Jon Osborne
Korbin Stich

http://may1601.sd.ece.iastate.edu

Client: Alliant Energy
Advisor: Dr. Doug Jacobson

December 7, 2015

Problem Statement

The goal of the project is to create a standalone security device that can be placed in an industrial network to monitor traffic, looking for security-related irregularities, and act as a low interaction honeypot.

Deliverable

- Raspberry Pi (Raspbian)
- Hardened System
- Honeypot & Logging Framework
- Small, passive IDS
- Automated deployment process

Conceptual Sketch

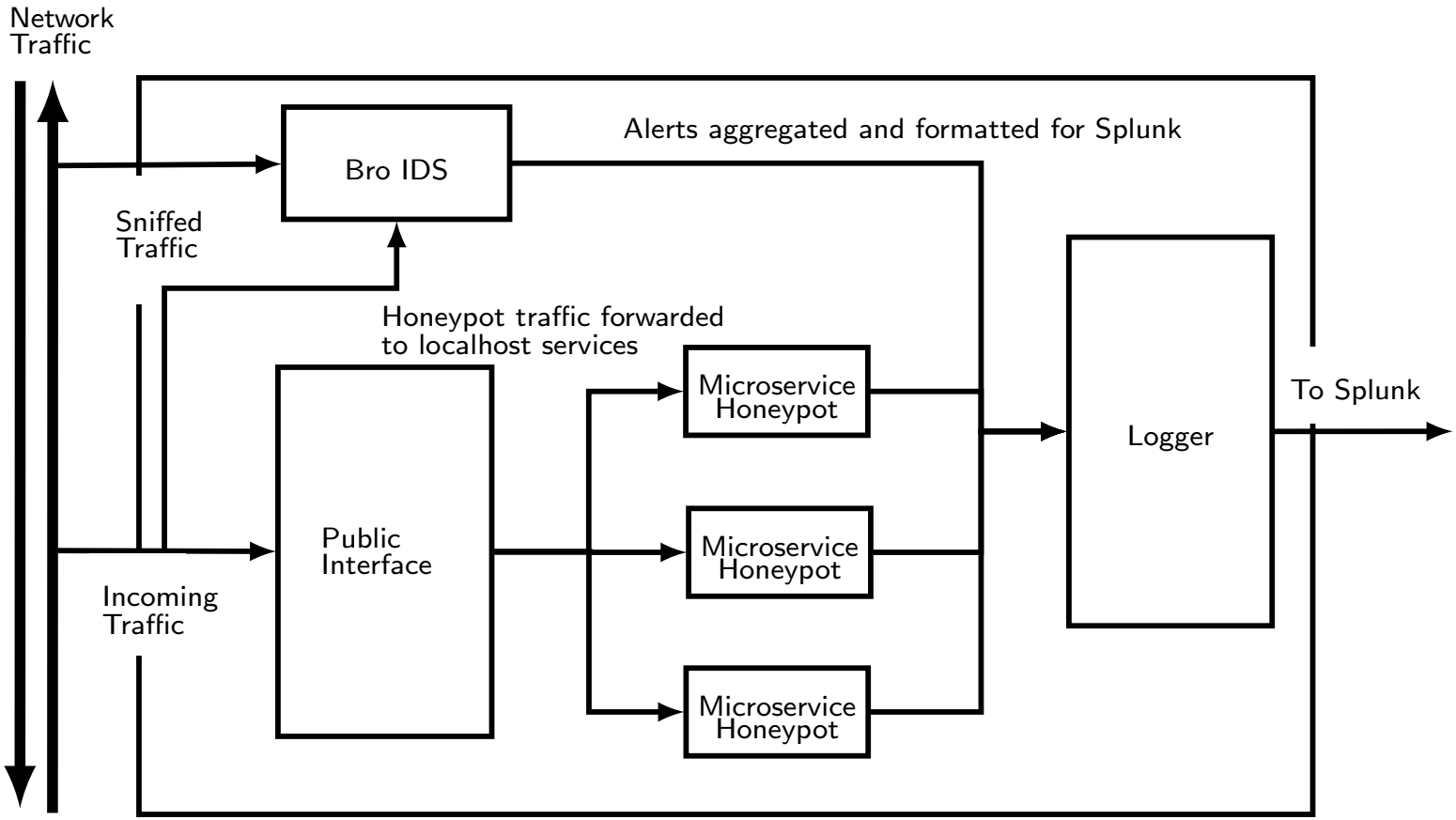


Figure: Simplified Device Internals

Functional Requirements

System Behavior

- Provide SSH, HTTP, HTTPS and necessary SCADA protocols
- A minimized passive intrusion detection system
- Log attempted intrusion attempts and alert necessary personnel
- Automatic deployment and remote management
- Easily customizable protocols

Non-functional Requirements

System Performance

- Secure system design
- Environmental considerations
- System must be low maintenance
- Simple stand alone device
- Capable of expansion beyond scope of project

Technical/Other Constraints and Considerations

- ARM architecture
- Work with Alliant's existing logging architecture
- Limited RAM provided by hardware
- Unclear SCADA protocols
- Dealing with sensitive information

Market Survey

Open Source Honeypots

ConPot	**Kippo**
Low Interaction	Medium Interaction
Siemens s7-200 PLC	Fake file system
MODBUS, HTTP, SNMP, s7comm	SSH

Potential Risks

- ESD, RFI, EMI.
- Ethernet Cable
- Physical Ingress Protection
- Limited Memory
- Security Concerns

Resource Cost Estimate

Item	Price
Raspberry PI B+	$69.99 (plus tax)
USB 2.0 Gigabit Ethernet Adapter	$16.99 (plus tax)

Total Device Cost: $89.98 (plus tax)
Total System Cost: $2,519.44 (plus tax)[1]

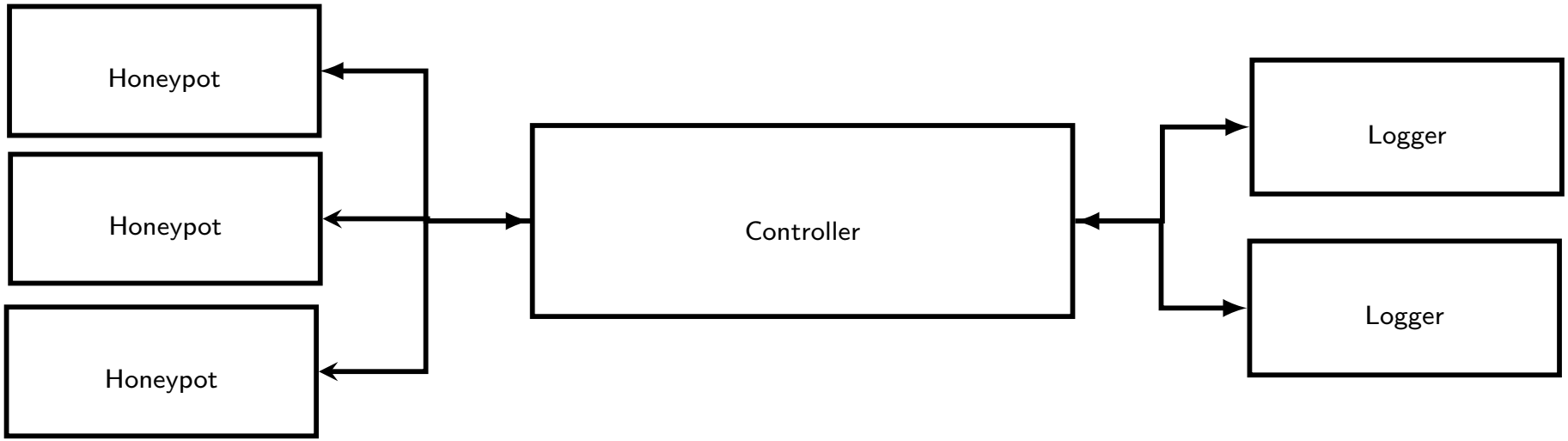
[1] Assuming 28 devices

Functional Decomposition

Function	**Component**
SSH, HTTP, etc.	Default plugin set
Monitor internal network traffic	IDS
Interaction Logs	Splunk Logger
Deployment/Management	Ansible

Detailed Design: Honeypot Framework

Figure: Plugin Framework Architecture



Modular, Extensible	**Secure by Design**
2 plugin types: Honeypot & Logger	Isolated, non-privileged processes
Communicate via lo unix socket RPC	Minimal protocol functionality

Technology Platform

Raspberry PI [2]

- Quad-Core 900 MHz Processor
- 1GB Ram
- Rasbian OS (Debian Based)

Software

- Ansible [3]
- Vagrant (Provisioned Testing) [4]
- Go Programming Language [5]

[2] http://www.amazon.com/CanaKit-Raspberry-Complete-Original-Preloaded
[3] www.ansible.com
[4] www.vagrantup.com
[5] https://golang.org

Test Plan

Go Programming Language
Integration testing can be completed by combining multiple unit tests into a larger framework with the "testing" package. What about multiple configurations or platforms though?

Vagrant allows for easy replication of test environments through virtual machines. This provides a method for plugin end-end testing for any device setup.

Vagrant allows for **Provisioning**. This means that a newly created VM can be give startup tasks that will run as an automated script.

Test Plan Continued

- Time complexity analysis
- Unit Testing, Integration Testing
- Code output verification

Example (Unit Testing)

```
func TestSplunk (t *testing.T){
m := map[string]string{"username":"bob","password":"1234"}
http:=Http{Method:"POST",Path:"index.html",Parameters:m}
ev := Event{...,Http: &http}
fmt.Println(event)
//Output: [username: bob password: 1234 \
           Method: POST Path: index.html]
}
$ go test -v
=== RUN TestSplunk
--- PASS: TestSplunk (0.00s)
```

Prototype Implementation

Component	Code	Status
Default Plugin Set	HTTP	Done
	HTTPS	Done
	SSH	Done
	Splunk Logger	Done
Automatic Deployment and Updates	Ansible playbooks	Done
Plugin Core	Framework	Work-in-progress
Physical Install	N/A	TODO
Testing	N/A	TODO

Current Project Status

Product

- Automated deployment complete
- Default honeypot plugins complete
- Near emulated prototype

In General

- Ahead of schedule
- Clear idea moving forward
- Flexible and prepared for change

Team Task Responsibilities

Dan Borgerding

- Communication Leader
- Iptables, Ansible Verification, Environmental Considerations

Nik Kinkel

- Concept Holder, Software Architect
- Ansible, Web Authorization, SSH,Vagrant

Jon Hope

- Webmaster
- Ansible

Jon Osborne

- Team Leader
- Splunk Communication, Plugin Framework

Korbin Stich

- Concept Holder
- Ansible Verification, Device Selection, Evironmental Considerations

Plan for Next Semester

Month	Schedule
January	Full prototype demo for Alliant security team
February	Incorporate client feedback, augment default plugin set
March	Hit 90% unit test coverage
April	Integration and acceptance testing, physical deployment
May	Final presentation

Table: Plan for Spring 2016

Questions